Faculty of Aerospace Engineering

An Optimal Control Approach for Estimating Aircraft Command Margins

With applications to Loss-Of-Control prevention

N. Govindarajan

October 25, 2012

Delft University of Technology

Challenge the future

An Optimal Control Approach for Estimating Aircraft Command Margins

With applications to Loss-Of-Control prevention

Master of Science Thesis

For obtaining the degree of Master of Science in Aerospace Engineering
at Delft University of Technology

N. Govindarajan

October 25, 2012

Faculty of Aerospace Engineering · Delft University of Technology

Delft University of Technology

Delft University Of Technology
Department Of
Control and Simulation

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled **"An Optimal Control Approach for Estimating Aircraft Command Margins"** by **N. Govindarajan** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: October 25, 2012

Readers:

prof. dr. ir. J. A. Mulder

dr. ir. Q. P. Chu

dr. ir. E. van Kampen

dr. ir. C.C. de Visser

ir. H. Bijl

Abstract

This dissertation presents an optimal control framework to determine a collection of open-loop command signals, that mathematically guarantees operation of a dynamical system within prescribed state constraints. The framework is applied to estimate real-time command margins for aircraft control systems so that, safe operation within the flight envelope can be assured under appropriate control action. The margins are perceived as useful information to a pilot, especially during off-nominal conditions, as it can aid the pilot in avoiding flight envelope excursions, generally considered as causal factors to Loss-Of-Control incidents in aviation.

Summary

Loss-Of-Control (LOC) is a major contributor to accidents and fatalities in all vehicle classes of aviation. Because of their general association with flight outside the normal operating envelope, LOC-induced accidents can potentially be avoided with Flight Envelope Protection (FEP). Current FEP systems however, are not sophisticated enough to adapt to off-nominal conditions arising from failures, damages, or other causes. As a result, lack-of situational awareness under these conditions often leads to improper control-actions. Towards this need, this thesis presents a novel framework for determining real-time "safety margins" for reference command signals of an aircraft control system. These margins mathematically guarantee operation of a dynamical system within state constraints. Provided that off-nominal dynamics are detected and identified almost immediately, the computed margins are perceived as useful information to a pilot operating in off-nominal conditions.

The proposed framework involves optimizing a cost functional over a space of admissible command signals. The cost functional describes whether a state trajectory of the system can violate the envelope within a given time-interval. This property of the cost functional is exploited in an iterative procedure to find margins for the command signals. The extrema of the cost functional is found through the Dynamic Programming (DP) principle. This consists of finding the associated value function, defined as the viscosity solution of a time-dependent Hamilton-Jacobi-Bellman Partial Differential Equation (HJB PDE). A novel scheme is presented to solve the non-linear PDE using regression techniques. The scheme employs value-iteration to estimate the coefficients of the candidate solution in a sequence of least-squares problems. The considered method is applied together with multivariate simplex splines to find approximations of the value function.

The entire concept is illustrated on a simplified aircraft system with state constraints on the pitch attitude. Basic heuristics are used to iteratively adjust the margins of the pitch command reference in a real-time setting. Experiments are conducted to study the behavior of the margins in response to abrupt changes in the dynamics. In compliance with expectations, simulations confirm that envelope excursions may occur under prolonged neglect of the margins. The envelope excursion is commonly preceded by a rapid shrinkage of the margins, indicating that the aircraft is approaching the edge of the envelope. Given the time-scale in which events occur, the proposed framework for envelope protection appears to be more suitable

for outer-loop control variables of aircraft control systems. The dynamics of these variables evolve at a slower pace, giving enough response-time for the pilot to take countermeasures.

The optimal control formulation assumes availability of complete information on the system dynamics. Acknowledging the fact that the dynamics are uncertain, and therefore unknown during off-nominal conditions, a routine has to be developed to estimate the aircraft dynamics in-flight. Since deliberate excitation of the controls is unacceptable during a failure condition, the dynamics have to be approximated with a method that circumvents the persistence of excitation requirement. An attempt is made to estimate the dynamics indirectly with a prediction model that closely follows the input-output behavior of the plant. By adaption of certain parameters, the prediction error dynamics are known to be Lyapunov stable, without any prerequisites on the richness of the signals. However, this convergence does not imply identification of the unknown plant parameters: if certain modes of the system are not sufficiently excited, the prediction model will not "learn" the anomalies in the system. As a result, the margins will be estimated incorrectly.

Acronyms

ACES	Adaptive Control and Evolvable Systems
ADM	Aeronautical Decision Making
AvSP	NASA's Aviation Safety Program
BVP	Boundary Value Problem
CLF	Courant-Friedrichs-Lewy
DP	Dynamic Programming
ENO	Essentially Non-Oscillatory
FEP	Flight Envelope Protection
GTM	Generic Transport Model
HJB	Hamilton-Jacobi-Bellman
HJB PDE	Hamilton-Jacobi-Bellman Partial Differential Equation
KKT	Karun-Kuhn-Tucker
LF	Lax-Friedrichs
LOC	Loss-Of-Control
LTI	Linear Time-Invariant
LTV	Linear Time-Variant
NDP	Neuro-Dynamic Programming
NASA	National Aeronautics and Space Administration
NextGen	Next Generation
OP	Optimization Problem
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
QLC	Quantitative Loss-of-control Criteria
RK	Runge-Kutta
ROA	Region-Of-Attraction
RSS	Residual Sum-of-Squares
VSST	Vehicle Safety Systems Technologies
WENO	Weighted Essentially Non-Oscillatory

Nomenclature

$\boldsymbol{x}$	State of the system
$\boldsymbol{y}_{ref}$	Reference command input
$\boldsymbol{y}$	Output of the system
n	State dimension
m	Reference command/output dimension
$\boldsymbol{f}_0$	Nominal system dynamics
$\boldsymbol{f}$	Off-nominal System dynamics
$\boldsymbol{h}$	Output equation
Y_{ref}	Command space
$\boldsymbol{\mathcal{Y}_{ref}}$	Command signal space
t_0	Current/initial time
$\boldsymbol{x}_0$	Current/initial state
T	Time-horizon
$\boldsymbol{\phi}$	System state trajectory
$\left(\boldsymbol{y}_{ref_{\min}}, \boldsymbol{y}_{ref_{\max}}\right)$	Command margins
K	Envelope
l	Implicit surface function
J	Cost functional
V_1	Value function associated with cost functional J
V_2	Value function associated with terminal cost opt. control problem
H	Hamiltonian
$\boldsymbol{g}^*$	Optimal feedback control law
$\hat{V}_2$	Approximation of value function V_2
A_0, B_0	Nominal LTI system matrices
A, B	Off-nominal LTI system matrices
$\Delta A, \Delta B$	Unknown terms in off-nominal LTI system
$\hat{\boldsymbol{x}}$	Predicted state
$\tilde{\boldsymbol{x}}$	Prediction error
$\Delta\hat{A}, \Delta\hat{B}$	Adaptive parameters of the prediction model
λ	Tuning parameter for the prediction model
γ	Adaption rate

Contents

Summary vii

Acronyms ix

Nomenclature xi

Acknowledgments xvii

1 Introduction 1
- 1-1 The thesis project . . . 1
  - 1-1-1 Research motivations . . . 1
  - 1-1-2 The research objective . . . 2
  - 1-1-3 Contributions . . . 2
- 1-2 Outline of the report . . . 3

I Problem description 5

2 Aircraft Loss-Of-Control prevention 7
- 2-1 Introduction . . . 7
- 2-2 Constitution of Loss-Of-Control incidents . . . 7
  - 2-2-1 Causal and contributing factors . . . 8
  - 2-2-2 Progression of LOC . . . 8
  - 2-2-3 Quantitative metrics for LOC . . . 9
- 2-3 NASA's aviation safety project . . . 9
- 2-4 A holistic approach to LOC accident prevention . . . 10
- 2-5 Flight Envelope Protection . . . 10
  - 2-5-1 Two different design philosophies in FEP systems . . . 11
  - 2-5-2 Current research in adaptive FEP technologies . . . 11
  - 2-5-3 Flight envelope estimation of anomalous aircraft . . . 12

3 Mathematical problem formulation **13**
3-1 Introduction 13
3-2 The aircraft as a command-driven control system 13
3-3 Some preliminaries on notation 14
3-3-1 The command-signal space 14
3-3-2 The system state trajectories 15
3-3-3 The state envelope 15
3-4 The mathematical problem statement 15

II An optimal control approach to the problem **19**

4 The optimal control framework for estimating aircraft command margins **21**
4-1 Introduction 21
4-2 Formulation as an optimal control problem 21
4-2-1 The cost functional 22
4-2-2 The value function and associated Hamilton-Jacobi-Bellman PDE 22
4-2-3 Notion of the 'safe-set' 26
4-3 The general procedure for estimating command margins 26

5 Illustrations on a simplified aircraft system **27**
5-1 Introduction 27
5-2 The simplified pitch dynamics of a Generic Transport Model 27
5-2-1 The nominal system 27
5-2-2 The off-nominal system 28
5-3 Off-line implementation of the optimal control formulation 29
5-3-1 Selection of a suitable implicit surface representation 29
5-3-2 Evaluation of the Hamiltonian and optimal feedback control law 29
5-3-3 Numerical solutions of the HJB PDE using level set methods 30
5-4 Results 30
5-4-1 The safe-set for different command margins 31
5-4-2 The safe-set for different time-horizons 31
5-4-3 System state trajectories inside and outside of the safe-set 31
5-4-4 'Safe' command margins for the nominal and off-nominal system 31

III Approximation of the value function in a real-time setting **39**

6 A Neuro-Dynamic Programming approach using value-iteration **41**
6-1 Introduction 41
6-2 Discretization of the HJB PDE 42
6-2-1 Discretization of the state 42

6-2-2 Discretization in time 43
6-3 Transcription into a non-linear optimization problem 46
6-4 Value-iteration for time-dependent HJB PDEs 47
6-4-1 Theoretical convergence analysis 49
6-4-2 Empirical study of the convergence 51
6-5 The value function approximation algorithm 52

7 Approximation of the value function with simplex splines 55
7-1 Introduction 55
7-2 The spline method 56
7-2-1 Spline model selection 56
7-2-2 Selection of suitable collocation points 57
7-2-3 Numerical implementation of the least-squares solution 57
7-3 Results for the simplified pitch dynamics model 58
7-3-1 Two different methods to find the safe-set 58
7-3-2 Performance metrics for quality assessment 59
7-3-3 Analysis of the results 59
7-4 Comparison to level set algorithms 66

IV In-flight command margin estimation 69

8 A step towards real-time command margin estimation 71
8-1 Introduction 71
8-2 A simple algorithm for the pitch dynamics model 71
8-3 Simulations 75
8-3-1 Case I: Pilot follows the margins 75
8-3-2 Case II: Pilot ignores the margins 78

9 Command margin estimation with uncertain dynamics 81
9-1 Introduction 81
9-2 The prediction model 81
9-2-1 The Linear Time-Invariant case 82
9-2-2 Estimating command margins with the prediction model 84
9-3 Simulations 85
9-3-1 Effects of the prediction model parameters 85
9-3-2 Convergence of the adaptive parameters 88

10 Conclusions 91

11 Recommendations 93

A Level set methods **95**
A-1 Introduction . . . 95
A-2 Numerically approximating the value function on a finite grid . . . 95
A-3 Approximation of the PDE terms . . . 96
A-3-1 The spatial derivatives . . . 96
A-3-2 The temporal derivative . . . 97
A-3-3 The Hamiltonian term . . . 97
A-4 The overall numerical scheme . . . 98
A-4-1 Stability and convergence properties . . . 99
A-5 Drawbacks of level set algorithms . . . 99

B Multivariate simplex splines **101**
B-1 Introduction . . . 101
B-2 Simplex spline theory . . . 101
B-2-1 Preliminaries on simplices, triangulations and Barycentric coordinates . . 101
B-2-2 The B-form polynomial . . . 103
B-2-3 Continuity order and inter-simplex constraints . . . 105
B-2-4 Computation of spline derivatives . . . 105
B-3 Regression with simplex splines . . . 106

C Measurable functions **109**

Bibliography **111**

Acknowledgments

This thesis project has turned out to be an excellent collaboration between the *Control & Simulation* department of the Aerospace Engineering faculty, and the *Adaptive Control and Evolvable Systems* (ACES) group at *NASA Ames Research Center*. The thesis was commenced at Delft University, and much of the work was performed at Ames under the direct guidance of NASA scientists. The author of this report would like to thank people from both institutions.

First and foremost, he would like to express his sincere gratitude to his mentor at Ames: *Dr. K. Krishnakumar*. Without his guidance and support, the research could not have been completed successfully. Special acknowledgments go also to *Dr. V. Stepanyan* and *Dr. J. Barlow*. Their tips and advice on technical matters are sincerely appreciated, and contributed significantly to the quality of the work.

Next, he would like to give special acknowledgments to his thesis supervisor at Delft: *Dr. Q. P. Chu*. His trust and confidence in the author's capabilities, by recommending him a position as an exchange student at NASA Ames is very much appreciated. In this regard, special acknowledgments must also go to professor *Dr. J.A. Mulder* for making the opportunity of working at NASA a reality, including his initiatives of contacting Krishna during the preliminary stages of the project. Furthermore, the author would like to acknowledge the help of *Dr. C.C. de Visser* on the implementation of multivariate simplex splines in Matlab. A word of thanks goes also to *Dr. E. van Kampen* and *Dr. T. Lombaerts* for their tips and advice on writing the report.

Finally, the author of this report would like to thank all his friends, colleagues, and family for their support and encouragement during research. Especially the unconditional moral support of his parents and sister has helped a great deal in completing the thesis project with success.

To my grandparents K. Seetha Laxmi and N. Krishnamoorthy.

Chapter 1

Introduction

Safety stands paramount in aviation. To this day, air travel remains the safest mode of transportation. Regardless, accidents in aviation are often accompanied with disastrous consequences. Therefore, improvement of flight safety remains an everlasting endeavor of the aerospace industry.

The majority of accidents in aviation are nowadays attributed to Loss-Of-Control (LOC). LOC is generally associated with unintended departure of aircraft into unusual flight conditions and are complex, in the sense that no single intervention strategy exists to prevent them. As part of NASA's Aviation Safety Program (AvSP), the Vehicle Systems Safety Technologies (VSST) project focuses on developing technologies that help improve safety of future air-vehicles. This includes extensive research in systems for LOC prevention.

1-1 The thesis project

This thesis project is aimed towards development of innovative Flight Safety Assurance (FSA) technologies. The FSA technologies must have the capability to provide support and guidance to future air-vehicles under any flight condition, including damage scenarios. The research specifically focuses on FSA technologies for large transport aircraft operated by fly-by-wire systems.

1-1-1 Research motivations

LOC often occurs when aircraft are pushed out of their normal operating envelope. In off-nominal conditions, current flight control systems and flight deck software appear to be ineffective in maintaining an aircraft within prescribed limits. The flight control system is designed for a nominal aircraft model and the system operators (i.e. pilot/auto-pilot) typically only know how to operate the aircraft safely under those nominal conditions.

Unawareness on the impact of in-flight failures and hazardous flight conditions (e.g. severe turbulence, wind-shear, icing conditions) to vehicle flight safety, often result in pilots/auto-pilots giving improper commands to the control system. These commands consequently put the aircraft in a dangerous condition for which recovery to normal flight is difficult to obtain. In off-nominal conditions, adaptive controllers may help improve flying & handling qualities to a certain extent. However, shrinkage of the safe maneuvering envelope is often inevitable.

For large transport aircraft, generally, reference command signals are provided to the aircraft control system, after which an existing controller steers the aircraft towards that reference. Due to system complexity and lack of situational awareness, pilots may give "unsafe" reference commands that push the aircraft out of the envelope. This thesis project focuses on classifying a collection of "safe" command signals that can guarantee operation of an aircraft within the flight envelope.

1-1-2 The research objective

The research objective of this thesis project is formulated as follows:

> *For an uncertain command-driven control system, determine, in real-time, margins for the reference command signals, such that, the system does not violate any predefined state constraints.*

In the formulated objective, the *command-driven control system* refers to the aircraft control system, driven by reference command signals of the pilot/auto-pilot. The adjective "*uncertain*" is used to emphasize that the dynamics are only known for the nominal system. The *margins* refer to bounds on the reference command signals, necessary to keep the aircraft within *state constraints*. The goal is to estimate these margins in *real-time* such that they adapt to changing conditions in-flight.

1-1-3 Contributions

The contributions of this thesis project can be summarized as follows:

- The optimal control approach, originally proposed by (Lygeros, 2004) to compute reachable sets of control systems, has been extended to find "safety margins" for command-driven control systems. Provided that the dynamics of the system are modeled correctly, these margins found by the proposed method give a mathematical guarantee of operation within the envelope.
- In response to shortcomings of the level set method regarding their real-time implementation, a novel scheme has been developed to find approximate solutions for a specific Hamilton-Jacobi-Bellman Partial Differential Equation (HJB PDE). The proposed scheme extends the collocation method in (Alwardi, 2010) with value-iteration, such that it can be implemented in an adaptive framework.
- Multivariate simplex splines have been used as regressors in the aforementioned scheme to find approximate solutions for the HJB PDE. To the best of the author's knowledge,

simplex splines, as in (de Visser, Chu, & Mulder, 2009), have never been used to solve this particular type of PDE.

- The global concept of finding command margins for aircraft control systems, has been successfully demonstrated on a simplified pitch dynamics model with state limitations on the pitch attitude. Simulations are conducted to illustrate the dynamic behavior of the margins, in response to sudden system degradation and changes in the envelope.
- As an alternative to traditional parameter estimation techniques, the prediction model, as in (Stepanyan, Barlow, Krishnakumar, & Bijl, 2011), has been implemented to find real-time approximations of the system dynamics during off-nominal conditions. The prediction model is used along with the optimal control formulation to obtain real-time estimates of the margins for systems with uncertain plant dynamics.

1-2 Outline of the report

This MSc dissertation is organized into *four* parts.

In *Part I* of the dissertation, the background and motivations of this thesis project are discussed in further detail. Chapter 2 presents an elaborate description of LOC, and introduces Flight Envelope Protection (FEP) as a potential remedy to these incidents. Chapter 3 formulates a mathematical problem statement that best describes the goals in the research objective.

In *Part II* of the dissertation, an optimal control framework is introduced to solve the mathematical problem statement. Chapter 4 casts the problem in an optimal control framework, which is subsequently solved using Dynamic Programming (DP). Chapter 5 illustrates the optimal control framework on a simplified longitudinal aircraft model with state limitations on the pitch attitude. The margins are estimated off-line, using level set algorithms.

In *Part III* of the dissertation, a novel scheme is presented to approximate the value function of the optimal control problem with regression techniques. Chapter 6 discusses the derivation of the novel scheme, that uses function approximators and value-iteration to obtain approximate solutions of the HJB PDE. Chapter 7 applies multivariate simplex spline theory to find approximations of the value function with the scheme derived in chapter 6.

In *Part IV* of the dissertation, the entire concept of finding command margins for aircraft control systems is illustrated in a real-time environment. Chapter 8 presents a simple algorithm to iteratively determine command margins for the simplified pitch dynamics model, introduced earlier in chapter 5. Experiments are conducted to study behavior of the margins to abrupt changes in the system dynamics and flight envelope. Chapter 9 introduces the prediction model for a Linear-Time-Invariant (LTI) system. Again, simulations are conducted for the simplified pitch dynamics model, but this time with the approximations of the system dynamics obtained through the prediction model.

The last part of the dissertation is followed by respectively: the conclusions in chapter 10, and recommendations in chapter 11. Furthermore, the dissertation contains two appendices: appendix A gives a brief introduction to level set methods, whereas appendix B gives a background on multivariate simplex spline theory.

Part I

Problem description

Chapter 2

Aircraft Loss-Of-Control prevention

2-1 Introduction

Loss-Of-Control (LOC) is a major contributor to accidents and fatalities across all vehicle classes, operational categories, and phases of flight. In the commercial jet category alone, LOC was the cause for 22 accidents resulting in 1,991 fatalities for the period between 1999 and 2008 (Belcastro & Foster, 2010). Consequently, prevention of LOC accidents remains one of the prime research focuses of the aeronautics community when concerning safety of flight.

This chapter aims to give a background to LOC and its associated causal factors. Extensive research has already been dedicated to LOC and comprehensive research programs have been laid out by aviation authorities to prevent LOC-induced accidents in the near and long-term future. In these programs, *Flight Envelope Protection* (FEP) is seen as a potential approach in dealing with LOC.

The chapter is organized as follows. In section 2-2, a detailed description is given on the composition of LOC incidents. Section 2-3 brings up current aviation safety projects of the National Aeronautics and Space Administration (NASA) specifically focused on research in LOC prevention. Section 2-4 introduces the holistic approach taken by industry to reduce likelihood of future LOC accidents. Section 2-5 discusses current developments in adaptive FEP technologies.

2-2 Constitution of Loss-Of-Control incidents

There exists no exact, quantitative definition for LOC. However, in the literature, LOC is often associated with flight outside of the normal operating envelope, with non-linear influences, and with an inability of the pilot to control the aircraft(Kwatny et al., 2009).

2-2-1 Causal and contributing factors

Aircraft LOC accidents are complex as they result from numerous causes. The major causal and contributing factors can be organized into three categories: *human-induced* factors, *environmentally-induced* factors and *system-induced* factors. Table 2-1 shows examples of causal factors in each category.

Human-induced	**Environmentally-induced**	**System-induced**
Poor pilot awareness.	*Adverse weather conditions(turbulence, icing, wind shear).*	*Poor design.*
Automation confusion.	*Foreign object damage(hail, bird strike, volcanic ash).*	*Loss of power and/or control effectiveness.*
Improper procedure.	*Wake vortices.*	*Aircraft system failures.*
Spatial disorientation.		*Erroneous sensor data.*

Table 2-1: Causal factors in LOC.

Studies in (Jacobson, 2010) cite human-induced factors as the most significant causal factor for LOC. However, no single category is solely responsible for LOC accidents. Accidents usually happen when combinations of breakdowns happen in different categories simultaneously, or in a short sequence.

2-2-2 Progression of LOC

LOC events usually occur in off-nominal conditions that may creep-up gradually, but also suddenly without any warning. LOC incidents are frequently preceded by a chain of events that can be grouped into three categories. These precursors, as they are referred to in (Belcastro & Foster, 2010), are:

- *Adverse on-board conditions*: vehicle impairment(including inappropriate vehicle configuration, contaminated airfoil, and improper vehicle loading), system faults and failures, or, vehicle damage to airframe and engines.
- *External hazards and disturbances*: poor visibility, wake vortices, wind shear, turbulence or icing conditions.
- *Vehicle upsets*: abnormal attitude, abnormal airspeed, uncontrolled descent (e.g. spiral dive) or stall/departure (including falling leaf and spin).

Analysis of accident data have shown that these precursors usually follow a certain sequential order. According to (Belcastro & Foster, 2010), LOC incidents are frequently initiated by adverse on-board conditions or external hazards/disturbances. Upsets are rarely the cause, but rather an outcome of the other precursors. A generalized sequence can be attributed to many LOC incidents. This generalized sequence is shown in figure 2-1 in which a LOC is initiated by a vehicle impairment/external hazard, followed by a inappropriate crew response causing an upset condition. Figure 2-1 indicates that flight crew decision-making and human-machine interactions appear to be often ineffective during off-nominal conditions.

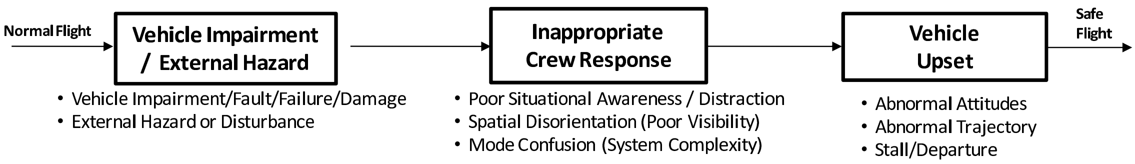


Figure 2-1: Generalized LOC sequence, courtesy of (Belcastro & Jacobson, 2010)

2-2-3 Quantitative metrics for LOC

As earlierly stated, there exists no exact quantitative definition for LOC. In an attempt to quantify LOC, (Wilborn & Foster, 2004) defined metrics and criteria that can be used to identify LOC events from flight data. These metrics are collectively known as the *Quantitative Loss-of-Control Criteria* (QLC) and consist of five envelopes related to airplane flight dynamics, aerodynamics, structural integrity and flight control use. The five envelopes, as defined in (Wilborn & Foster, 2004), are:

- The *Adverse Aerodynamics* envelope, describing limits on the angle of attack α and side-slip angle β.
- The *Unusual Attitude* envelope, describing on the bank angle ϕ and pitch attitude θ.
- The *Structural Integrity* envelope, describing limits on the airspeed V and normal load factor n_z.
- The *Dynamic Pitch Control* envelope, describing limits on the pitch axis control authority and dynamic pitch attitude ($\theta' = \theta + \dot{\theta}$).
- The *Dynamic Roll Control* envelope, describing limits on the roll axis control authority and dynamic roll attitude ($\phi' = \phi + \dot{\phi}$).

These five envelopes allow for a numerical methodology to reliably identify LOC events. The numerical methodology consists of analyzing the time history of state trajectories in the parametric spaces defined by the envelopes. A maneuver that violates three or more QLC envelopes is classified as a LOC incident. A maneuver that violates only two envelopes is considered borderline LOC. Ordinary maneuvers, even when performed aggressively, appear to hardly ever violate any of these envelopes. These conclusions were made based upon analysis of flight-test data and actual LOC accident data.

2-3 NASA's aviation safety project

In order to meet with the expected growth in future air travel, NASA in conjunction with the Federal Aviation Authorities (FAA) has initiated the Aviation Safety Program (AvSP). The AvSP aims to develop technologies in order to improve overall safety of flight for vehicles operating in the Next Generation Air Transportation System (NextGen).

The Vehicle Systems Safety Technologies (VSST) project is one of the projects which comes under the AvSP. The VSST project has the following objective:

> *Provide knowledge, concepts and methods to* ***avoid****,* ***detect****,* ***mitigate*** *and* ***recover*** *from hazardous flight conditions, and to maintain vehicle airworthiness and health.* (source: http://www.aeronautics.nasa.gov/programs_avsafe.htm)

To meet this goal, one of the problems that are being addressed by the VSST project is the occurrence of LOC accidents, induced by unintended departure of aircraft into unusual flight conditions. The VSST project aims to develop, assess, and validate technologies for avoiding, detecting and resolving conditions that can lead to LOC. This thesis project addresses concerns which come under the VSST project's framework.

2-4 A holistic approach to LOC accident prevention

Due to the complexity and multidisciplinary nature of LOC accidents, there exists no single intervention strategy to prevent them. Rather, a holistic approach must be taken which systematically break-downs the chain of events that precede a LOC incident. This holistic approach, as discussed by (Belcastro & Jacobson, 2010), requires the development of the following technologies:

- *Advanced modeling and simulation technologies* for characterizing off-nominal condition effects on vehicle dynamics and control characteristics, including vehicle failures and damage, vehicle upset conditions, wind shear and turbulence, wake vortices, icing, and key combinations of these.
- *Vehicle Health Management technologies* for continually assessing and predicting the health of the airframe, propulsion system, and avionics systems in real-time.
- *Flight Safety Assurance (FSA) technologies* for continually assessing and predicting the impact of off-nominal conditions on vehicle flight safety, and to provide resilient guidance and control capabilities under off-nominal conditions.
- *Effective crew-system interface technologies* for providing improved situational awareness and crew response under off-nominal conditions.
- *Validation and verification technologies* for the comprehensive evaluation and certification of these technologies.

2-5 Flight Envelope Protection

Currently, flight control systems and flight deck software are effective under only nominal conditions. To compensate for these shortcomings, FSA systems need to be developed that can provide more support to system operators (pilot/auto-pilot) in off-nominal conditions. Flight Envelope Protection (FEP) is seen in this regard as a useful tool to improve flight safety. The task of FEP is to monitor and maintain vehicle operation within prescribed limits under all circumstances.

2-5-1 Two different design philosophies in FEP systems

Researchers and industry handle two distinct philosophies when concerning the design of FEP systems. In the first philosophy, the responsibility of maintaining the aircraft within prescribed limits is given to the flight control system. An active role is taken by the control system as it can override pilot control actions to prevent aircraft from leaving the envelope. In the second philosophy, on-board systems compute margins for pilots control actions. Similar to a flight director, these margins are presented to the pilot, who in turn, has to interpret these signals to operate the aircraft safely. In this design philosophy, the pilot is given more freedom over the control of the aircraft as he/she has the final authority.

According to (Falkena, Borst, Chu, & Mulder, 2011), the advantage of the first design philosophy is that protection systems will maintain aircraft within the safe flight envelope, regardless of pilot control actions. However, such a system may also prevent the aircraft from being operated at its full capacity. In fact, for some cases, it might even be necessary to violate the envelope in order to save the aircraft. (Falkena et al., 2011) presented the China Airlines incident in 1985 as an example of this point. In the incident, the crew was forced to over-stress the horizontal tail surfaces (i.e. exceeding the load factor envelope) of a B-747 to recover from a roll and near- vertical dive following an automatic disconnect of the autopilot.

This thesis project focuses more on FEP systems that follow the second design philosophy. The motivation for this is that automation cannot take over all tasks of the pilot as systems, including FEP technologies, can fail. The important Aeronautical Decision Making (ADM) should always be left to a human operator and situational awareness must be improved instead, such that pilots make the correct decisions.

2-5-2 Current research in adaptive FEP technologies

Current envelope protection technologies are very crude and ineffective under off-nominal conditions. Off-nominal flight conditions resulting from failures, damage, or other causes, often lead to unsafe regions in the flight envelope. Insufficient knowledge seems to exist on the operators part for controlling the aircraft safely in these conditions. In recent times however, a lot of research has been dedicated to adaptive flight envelope estimation & protection systems.

In (Barlow, Stepanyan, & Krishnakumar, 2011) and (Krishnakumar, Stepanyan, & Barlow, 2011) limits were estimated for control actions of pilots in order to provide assistance for avoiding envelope excursions. The minimum step inputs were determined for some given time instances, required to steer an aircraft out of the envelope. The most restrictive step command is subsequently taken as a safety margin. The data-based predictive control approach, used to estimate this margins, had however its limitations. First of all, the method is applicable to only linear systems. Besides, potential state constraint violations were analyzed only at some discrete time instances, instead over a continuous interval.

In (Unnikrishnan, 2006), another method for adaptive envelope protection was introduced. The method, specifically designated for unmanned aerial vehicles, was based on minimizing an objective functional which was a sum of both time and control effort. For certain measures of aggressiveness in the controls, the objective functional determined the time required for an aircraft to hit the envelope boundary. The area norm of the optimal control profile was subsequently used to derive limits for the control inputs.

Finally, (Bayen, Mitchell, Oishi, & Tomlin, 2007) introduced an approach for envelope protection which involved finding a state domain with the property that *at-least one* is retained which can prevent the aircraft from the envelope. In the viability literature (Aubin, 1991), this state domain is referred as the viability kernel. An optimal control formulation for computing reachable sets was used to find this viability kernel. The gradient of the value function associated to the optimal control problem is used to design a regulatory feedback law to keep the system inside the envelope. This feedback control law is provisionary as it only needs to be activated when the aircraft get near to boundary of the viability kernel.

2-5-3 Flight envelope estimation of anomalous aircraft

The flight envelope is commonly described in terms of limits on airspeed, pitch and roll angles, angle of attack, and load factor. During the production phase, engineers have specifically designed the aircraft for operation within a certain envelope. In the occurrence of failures(e.g. structural damage of the aircraft), one can expect that the envelope limits will narrow. The original envelope is no longer correct and a new envelope has to be determined in-flight.

In the literature, assessment of the flight envelope was proposed to be done in several ways. The paper of (L. Tang et al., 2009) summarizes some of the recent development in adaptive flight envelope estimation. Anyhow, one proposal was to determine all achievable aircraft trim conditions. The collection of attainable equilibrium states and their local stability maps is seen as a comprehensive and consistent way of representing the aircraft maneuvering envelope. In (Goman, Khramtsovsky, & Kolesnikov, 2008), a systematic approach was presented to compute all attainable steady states for a general class of helical trajectories. In (Y. Tang, Atkins, & Sanner, 2007), the stable and controllable trim conditions were determined off-line for a Generic Transport Model (GTM) with left wing damage.

Another approach is to treat the estimation of the safe flight envelope as a reachability problem. In (van Oort, 2011), the flight envelope was characterized as the intersection between the forward and backward reachable set of the aircraft trim set. In (Kwatny et al., 2009), the flight envelope was defined as the viability kernel for some predefined (never-to-be exceeded) constraints. (Seube, 2002) computed the viable set for an aircraft taking-off in wind-shear. The system was modeled as a differential game with the wind-shear components taken as inputs of an adversary player.

Mind that any of the aforementioned approaches are computationally very intensive. Furthermore, the global model of the off-nominal system needs to be known in order to make high-fidelity predictions on the envelope limits. This makes envelope estimation in off-nominal conditions a highly physical and multi-dimensional problem as it requires a fully integrated modeling of aerodynamic, structural and propulsive aspects of the aircraft. As a result, real-time flight envelope estimation is not a trivial task and stands as a separate research topic by itself. In this thesis project, the following assumption is made regarding knowledge of the envelope.

Assumption: *The flight envelope is known for any given condition of the aircraft.*

Chapter 3

Mathematical problem formulation

3-1 Introduction

Many physical systems operate safely only when confined to certain operating conditions. In other words, the system state trajectories have to be kept within certain state constraints under all conditions. For an aircraft, the collection of acceptable state trajectories is referred to as the *maneuvering flight envelope*. Operation outside of the flight envelope is strongly discouraged as it puts the aircraft in a dangerous condition. It remains the responsibility of the system operator(i.e. pilot/automatic control system) to ensure that state constraints are never violated. The operator must exercise the proper commands such that the envelope is never exceeded.

The primary objective in this thesis project is to classify a collection of feasible command signals that mathematically guarantee operation of an aircraft within the flight envelope. This chapter discusses how the objective is approached quantitatively by formulating a clear and concise mathematical problem statement.

The chapter is organized as follows. Section 3-2 discusses the treatment of the aircraft as a command-driven control system. Section 3-3 introduces some definitions and notation which are necessary to formulate the problem statement. In section 3-4, the actual problem statement is finally presented.

3-2 The aircraft as a command-driven control system

From the viewpoint of the pilot or auto-pilot, the aircraft is seen as a *command-driven* control system. A reference command signal is provided to the aircraft after which an existing controller steers the aircraft towards that reference. Any proper command-driven control system has the tendency to track the reference signal as closely as possible. The tracking performance depends on the type of controller and plant dynamics. In off-nominal conditions degradation of the tracking performance is inevitable. Under these circumstances, command signals to the control system have to be given with extreme caution.

In this thesis, the aircraft is treated as a command-driven control system. The dynamics are described by a system of ordinary differential equations:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{y}_{ref}\right) \tag{3-1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ denotes the state, $\boldsymbol{y}_{ref} \in \mathbb{R}^m$ denotes the input, and $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$, a function mapping the state and input to the state derivative. The notation $\boldsymbol{y}_{ref}$ in eq(3-1) is used to emphasize that the input to the system is a reference command. The output of the dynamical system is given by expression:

$$\boldsymbol{y} = \boldsymbol{h}\left(\boldsymbol{x}\right) \tag{3-2}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ denotes the output, and $\boldsymbol{f} : \mathbb{R}^n \mapsto \mathbb{R}^m$, a function mapping the state to the output.

Remark: *Mind that the dynamics eq(3-1) are* not *known exactly, i.e. a model of the dynamics is available only for the nominal system, denoted by the expression:*

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_0\left(\boldsymbol{x}, \boldsymbol{y}_{ref}\right) \tag{3-3}$$

3-3 Some preliminaries on notation

To formulate the problem statement, certain notations have to be introduced. These notations describe three aspects of the problem. The aspects are respectively: the command signals that are provided to the system, the resulting state trajectories being followed, and, the state constraints that have to be satisfied.

3-3-1 The command-signal space

Let the reference commands $\boldsymbol{y}_{ref}$ take values from the set $\mathrm{Y}_{ref} \subset \mathbb{R}^m$. Within this context, Y_{ref} is referred to as the command space. A function space can be defined which describes the collection of admissible reference command signals $\boldsymbol{y}_{ref}(\cdot)$[1]. This function space is referred to as the *command-signal space* for which the formal definition is given below.

Definition 3.1: ***The command-signal space***
Let $\mathrm{Y}_{ref} \subset \mathbb{R}^m$. The command-signal space is defined as the collection of functions:

$$\boldsymbol{\mathcal{Y}_{ref}} := \left\{ \boldsymbol{y}_{ref}(\cdot) : [t_0, t_0 + T] \mapsto \mathrm{Y}_{ref} \;\middle|\; \boldsymbol{y}_{ref}(\cdot) \text{ is measurable} \right\} \tag{3-4}$$

where t_0 denotes the present time and $T > 0$ the prediction horizon.

Remark: *In definition 3.1 measurable effectively translates to the set of piece-wise continuous signals in the time-interval $[t_0, t_0 + T]$, see (van Oort, 2011). A more thorough description of measurable signals/functions can be found in appendix C.*

[1] The notation $(\cdot)$ is used to distinguish a signal from a single reference command input.

3-3-2 The system state trajectories

For some initial condition $\boldsymbol{x}_0$ and reference command signal $\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}$, the system will follow a specific state trajectory $\boldsymbol{\phi} : [t_0, t_0 + T] \mapsto \mathbb{R}^n$. The following notation is introduced to denote this trajectory:

$$\boldsymbol{\phi}\left(\tau; \boldsymbol{x}_0, \boldsymbol{y}_{ref}(\cdot)\right) \tag{3-5}$$

The running variable in eq(3-5) is $\tau \in [t_0, t_0 + T]$ and the semi-column separates the argument τ from the parameters $\boldsymbol{x}_0$ and $\boldsymbol{y}_{ref}(\cdot)$.

3-3-3 The state envelope

The trajectories of the system have to be maintained inside a state envelope $K \subset \mathbb{R}^n$. This envelope represents the set of states for which the aircraft is advised to operate in. The flight envelope is described by a collection of inequality constraints that need to be complied with simultaneously.

Suppose there are r independent inequality constraints:

$$l_i(\boldsymbol{x}) \leq 0, \qquad \text{for } i = 1, \ldots, r \tag{3-6}$$

Then the envelope K may be denoted as a sub-zero level set of a higher dimensional surface. This is known as an implicit surface representation for which a definition is given below.

Definition 3.2: ***Implicit surface representation***
Consider a closed set $K \subseteq \mathbb{R}^n$ defined by the inequality constraints eq(3-6), an implicit surface representation for K is a function $l(\boldsymbol{x}) : \mathbb{R}^n \mapsto \mathbb{R}$ such that:

$$K := \{\boldsymbol{x} \in \mathbb{R}^n \mid l(\boldsymbol{x}) \leq 0\} \tag{3-7}$$

Remark: *There are multiple choices for $l(\boldsymbol{x})$ to describe the same set. A straightforward one would be:*

$$l(\boldsymbol{x}) = \max\{l_1(\boldsymbol{x}), \ldots, l_r(\boldsymbol{x})\} \tag{3-8}$$

The implicit surface presentation provides a useful means to check whether a state trajectory has crossed envelope boundaries. A particular trajectory $\boldsymbol{\phi}$ has left the envelope K at least once for the time interval $[t_0, t_0 + T]$, if and only if:

$$\max_{\tau \in [t_0, t_0+T]} l\left(\boldsymbol{\phi}\left(\tau; \boldsymbol{x}_0, \boldsymbol{y}_{ref}(\cdot)\right)\right) > 0 \tag{3-9}$$

This fact is exploited later in chapter 4 to solve the problem statement.

3-4 The mathematical problem statement

This thesis aims to classify a "safe" command-signal space $\boldsymbol{\mathcal{Y}_{ref}}$ for the system eq(3-1). Safety is interpreted here as whether a command signal $\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}$ can steer the system out of the envelope in the time window $[t_0, t_0 + T]$.

Let $\mathrm{Y}_{ref} \subset \mathbb{R}^m$ be parametrized as:

$$\mathrm{Y}_{ref} := \left[\boldsymbol{y}_{ref_{\min}}, \boldsymbol{y}_{ref_{\max}}\right] \tag{3-10}$$

where $\boldsymbol{y}_{ref_{\min}}$ and $\boldsymbol{y}_{ref_{\max}}$ denote respectively the lower and upper limits on the reference command. The pair $\left(\boldsymbol{y}_{ref_{\min}}, \boldsymbol{y}_{ref_{\max}}\right)$ is collectively referred to as the *command margins* for the system.

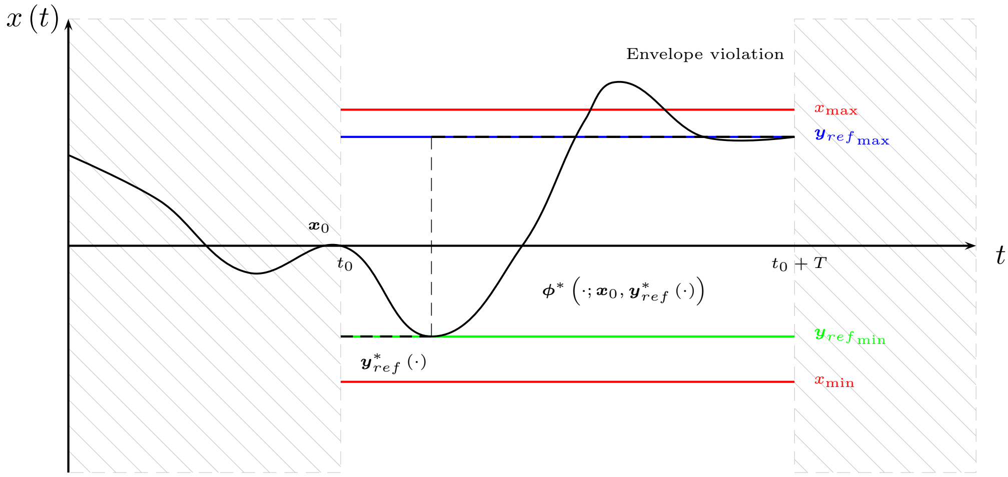
The following problem is formulated:

> *Determine the command margins* $\left(\boldsymbol{y}_{ref_{\min}}, \boldsymbol{y}_{ref_{\max}}\right)$*, such that, no matter what command signal* $\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}$ *is provided to the system, the state* $\boldsymbol{x}$ *does not leave the envelope* K *for the next* T *seconds, i.e.* $\boldsymbol{\phi}\left(\tau; \boldsymbol{x}_0, \boldsymbol{y}_{ref}(\cdot)\right) \in K$*,* $\forall \tau \in [t_0, t_0 + T]$*.*

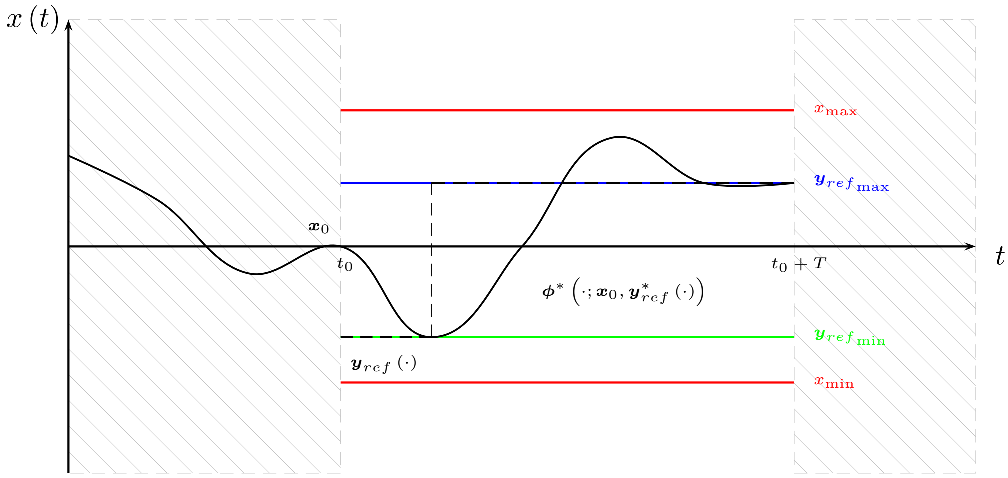
Figure 3-1 illustrates[2] the problem formulated. The figure shows a state trajectory of an aircraft system until some present time t_0. The system is currently at the state $\boldsymbol{x}_0$ for which a time window is depicted from the present time t_0 to some future time $t_0 + T$. Certain command margins are set for the time-window shown in the figure. For the margins in figure 3-1(a) it appears that there exists a $\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}$ which gives rise to an extremal trajectory which violates the envelope. On the other hand, for the margins in figure 3-1(b) there exists no such command signal. In this respect, the command margins in figure 3-1(b) are considered to be "safe" for the system.

The objective is to estimate the command margins in *real-time* for uncertain dynamics along the trajectory being followed. The time window in figure 3-1 is hence moving and the time-horizon T is a design parameter that needs to be chosen carefully such that the transients in the dynamics are captured sufficiently in the prediction.

[2]For illustration purposes, the figure assumes that the reference command is equal to the state. Mind that this does not hold true for eq(3-2) in the generic case.



(a) There exists *at-least one* reference command $\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}$ that can steer the system outside the envelope in T seconds.



(b) There exists *no* reference command $\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}$ that can steer the system outside the envelope in T seconds.

Figure 3-1: "Safe" and "unsafe" command margins for some hypothetical dynamical system

Part II

An optimal control approach to the problem

Chapter 4

The optimal control framework for estimating aircraft command margins

4-1 Introduction

The problem statement in section 3-4 requires one to analyze the properties of a whole class of system state trajectories at once. The objective is to determine margins for the reference command signals, such that a predefined state envelope is never violated for a given future time interval.

In this chapter, a methodology is presented to solve the above-stated problem in an effective manner. The problem statement is cast as an optimal control problem and applies principles from (Lygeros, 2004) and (Mitchell, Bayen, & Tomlin, 2005) to compute reachable sets of dynamical systems. The proposed method offers a systematic way of analyzing state trajectory properties of a dynamical system, in response to a large class of input signals. The optimal control approach is very generic as it assumes nothing on the structure of the system except for the dynamics $\boldsymbol{f}$ being Lipschitz continuous.

The chapter is organized as follows. Section 4-2 covers the details of the optimal control approach. The section introduces the cost functional which needs to be optimized and discusses also the rationale behind choosing it. Furthermore, the value function that optimizes the cost functional, is characterized as the unique, viscosity solution of a time-dependent Hamilton-Jacobi-Bellman Partial Differential Equation (HJB PDE). Section 4-3 summarizes the overall iterative procedure to find margins for the command-driven control system.

4-2 Formulation as an optimal control problem

The following iterative approach is taken to solve the problem statements. Given certain margins on the reference command, one verifies the existence of command signals that can steer the system out of the envelope. The margins $\left(\boldsymbol{y}_{ref_{\min}}, \boldsymbol{y}_{ref_{\max}}\right)$ are adjusted accordingly and the process is repeated until satisfactory margins are obtained.

The iterative approach requires solving the following complimentary problem:

> *Given an initial condition $\boldsymbol{x}_0$ and command margins $\left(\boldsymbol{y}_{ref_{\min}}, \boldsymbol{y}_{ref_{\max}}\right)$, does there exist the property that, no matter what admissible command signal $\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}$ is given, the system state trajectories $\boldsymbol{\phi}\left(\tau; \boldsymbol{x}_0, \boldsymbol{y}_{ref}(\cdot)\right)$ stay inside the envelope K for the time interval $[t_0, t_0 + T]$?*

This complimentary problem can be solved with optimal control ideas.

4-2-1 The cost functional

Consider the following cost functional:

$$J\left(\boldsymbol{x}_0, \boldsymbol{y}_{ref}(\cdot)\right) = \max_{\tau \in [t_0, t_0+T]} l\left(\boldsymbol{\phi}\left(\tau; \boldsymbol{x}_0, \boldsymbol{y}_{ref}(\cdot)\right)\right) \tag{4-1}$$

where $\boldsymbol{\phi}\left(\tau; \boldsymbol{x}_0, \boldsymbol{y}_{ref}(\cdot)\right)$ is a system trajectory as defined by eq(3-5). Recall from eq(3-9) that when $J \leq 0$, the state trajectory of the system has not violated the envelope in the time window $[t_0, t_0 + T]$.

Now suppose the following optimization problem is solved:

$$J^*\left(\boldsymbol{x}_0\right) = \max_{\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}} J\left(\boldsymbol{x}_0, \boldsymbol{y}_{ref}(\cdot)\right) \tag{4-2}$$

Then the following holds true:

- $J^*\left(\boldsymbol{x}_0\right) \leq 0$: the system will not violate any state constraints in the next T seconds, no matter what command signal $\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}$ is provided.
- $J^*\left(\boldsymbol{x}_0\right) > 0$: there exists some command signal(s) $\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}$ that can steer the system out of the envelope within the next T seconds.

Basically, the optimization in eq(4-2) is a search within the function space $\boldsymbol{\mathcal{Y}_{ref}}$ for the extremal command signal that would steer the system as close to the envelope boundaries as possible. The sign of $J^*\left(\boldsymbol{x}_0\right)$ indicates whether the envelope is violated or not. The next section shows how the optimal control problem can be solved.

4-2-2 The value function and associated Hamilton-Jacobi-Bellman PDE

Let $t \in [t_0, t_0 + T]$ and denote:

$$\boldsymbol{\mathcal{Y}_{ref}}_{[t, t_0+T]} := \left\{ \boldsymbol{y}_{ref}(\cdot) : [t, t_0 + T] \mapsto \mathrm{Y}_{ref} \;\middle|\; \boldsymbol{y}_{ref}(\cdot) \text{ is measurable} \right\} \tag{4-3}$$

Furthermore, let:

$$\boldsymbol{\phi}\left(\tau; \boldsymbol{x}, t, \boldsymbol{y}_{ref}(\cdot)\right) \tag{4-4}$$

denote the system trajectory for some state $\boldsymbol{x}$ at time t and reference command signal $\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}_{[t,t_0+T]}$.

The following function $V_1 : [t_0, t_0+T] \times \mathbb{R}^n \mapsto \mathbb{R}$ is introduced:

$$V_1(t, \boldsymbol{x}) := \sup_{\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}_{[t,t_0+T]}} \left\{ \max_{\tau \in [t, t_0+T]} l\left(\boldsymbol{\phi}\left(\tau; \boldsymbol{x}, t, \boldsymbol{y}_{ref}(\cdot)\right)\right) \right\} \tag{4-5}$$

Eq(4-5) is known as the value function for the cost functional eq(4-1). From the definition it follows that:

$$V_1(t_0, \boldsymbol{x}_0) = J^*(\boldsymbol{x}_0)$$

The value function V_1 can be solved using the dynamic programming (DP) principle. The principle of optimality states the following for V_1:

$$\begin{aligned} V_1(t, \boldsymbol{x}) = \sup_{\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}_{[t,t_0+T]}} \Bigg(\max \Bigg\{ & \max_{\tau \in [t, t+h]} l\left(\boldsymbol{\phi}\left(\tau; \boldsymbol{x}, t, \boldsymbol{y}_{ref}(\cdot)\right)\right), \\ & V_1\left(t+h, \boldsymbol{\phi}\left(t+h; \boldsymbol{x}, t, \boldsymbol{y}_{ref}(\cdot)\right)\right) \Bigg\} \Bigg) \end{aligned} \tag{4-6}$$

In words, the eq(4-6) states that the best possible cost at a time t and state $\boldsymbol{x}$, is either equal to the cost-to-go at time $t+h$ and state $\boldsymbol{\phi}\left(t+h; \boldsymbol{x}, t, \boldsymbol{y}_{ref}(\cdot)\right)$, or to the maximum of l along the system trajectory $\boldsymbol{\phi}$ during time period: $[t, t+h]$.

The *infinitesimal version* of eq(4-6) is a specific Hamilton-Jacobi-Bellman Partial Differential Equation (HJB PDE). The following theorem presents the HJB PDE associated with eq(4-5).

Theorem 4.1
The value function $V_1 : [t_0, t_0+T] \times \mathbb{R}^n \mapsto \mathbb{R}$ is the unique, continuous viscosity solution of the following time-dependent Hamilton-Jacobi-Bellman PDE:

$$\frac{\partial V_1(t, \boldsymbol{x})}{\partial t} + \max \left\{ 0, H\left(\boldsymbol{x}, \frac{\partial V_1(t, \boldsymbol{x})}{\partial x}^T\right) \right\} = 0 \tag{4-7a}$$

$$V_1(t_0+T, \boldsymbol{x}) = l(\boldsymbol{x}) \tag{4-7b}$$

where the Hamiltonian H is defined by:

$$H(\boldsymbol{x}, \boldsymbol{p}) = \max_{\boldsymbol{y}_{ref} \in \mathrm{Y}_{ref}} \boldsymbol{p}^T \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{y}_{ref}\right) \tag{4-8}$$

Proof. The formal proof of this theorem involves the rigorous theory of viscosity solutions, and can for example be found in (Lygeros, 2004). □

Remark: *HJB PDEs, including the variant in theorem 4.1 rarely have solutions in the classical sense. Even when $\boldsymbol{f}$ and l are infinitely smooth, the value function may exhibit shocks and rarefactions. As a result, the value function is not being differentiable everywhere. (Mitchell, 2002) gives a physical explanation of this phenomena which is basically caused by the existence of multiple optimal trajectories. Nevertheless, researchers (Crandall & Lions, 1983; Crandall, Evans, & Lions, 1984) introduced a non-classical or weak solution for HJB*

PDEs like eq(4-7), under the name of viscosity solutions. Formally the viscosity solution for eq(4-7) can be defined as follows. Let $\psi(t, \boldsymbol{x})$ denote any continuously differentiable test function. The bounded, uniformly continuous function V_1 is a viscosity solution to eq(4-7) if, whenever $\psi(t, \boldsymbol{x}) - V_1(t, \boldsymbol{x})$ is a local maximum:

$$\frac{\partial \psi(t, \boldsymbol{x})}{\partial t} + \max\left\{0, H\left(\boldsymbol{x}, \frac{\partial \psi(t, \boldsymbol{x})}{\partial x}^T\right)\right\} \leq 0,$$

and whenever $\psi(t, \boldsymbol{x}) - V_1(t, \boldsymbol{x})$ is a local minimum:

$$\frac{\partial \psi(t, \boldsymbol{x})}{\partial t} + \max\left\{0, H\left(\boldsymbol{x}, \frac{\partial \psi(t, \boldsymbol{x})}{\partial x}^T\right)\right\} \geq 0$$

Although this definition is not particularly insightful. The viscosity solution has a very important practical application: wherever the value function is differentiable, the HJB PDE is satisfied in the classical sense. Furthermore, the viscosity solution is unique. For more on HJB equations and viscosity solutions, readers are referred to (Bardi & Capuzzo-Dolcetta, 2008).

Remark: *In eq(4-8), $\boldsymbol{p}$ denotes the co-state as in the Pontryagin's Maximum Principle. See (Evans, unknown) or (Kirk, 1970) for more background.*

V_1 has connections with a value function associated to a more standard optimal control problem. Let the function $V_2 : [t_0, t_0 + T] \times \mathbb{R}^n \mapsto \mathbb{R}$ be defined by:

$$V_2(t, \boldsymbol{x}) = \sup_{\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}_{[t, t_0+T]}} l\left(\boldsymbol{\phi}\left(t_0 + T; \boldsymbol{x}, t, \boldsymbol{y}_{ref}(\cdot)\right)\right) \tag{4-9}$$

V_2 is the value function that optimizes a finite-horizon cost functional with only a terminal cost and no running costs. It is related to V_1 as follows.

Theorem 4.2
Consider the value function V_2 as given in eq(4-9). The following relation holds:

$$V_1(t, \boldsymbol{x}) = \max_{\tau \in [t, t_0+T]} V_2(\tau, \boldsymbol{x}) \tag{4-10}$$

Proof. A proof of this statement can be found in (Lygeros, 2004) □

Remark: *Theorem 4.2 basically shows that one can interchange* max *and* sup *operators in eq(4-5), i.e.*

$$\begin{aligned} V_1(t, \boldsymbol{x}) &= \sup_{\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}_{[t, t_0+T]}} \left\{ \max_{\tau \in [t, t_0+T]} l\left(\boldsymbol{\phi}\left(\tau; \boldsymbol{x}, t, \boldsymbol{y}_{ref}(\cdot)\right)\right) \right\} \\ &= \max_{\tau \in [t, t_0+T]} \left\{ \sup_{\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}_{[\tau, t_0+T]}} l\left(\boldsymbol{\phi}\left(t_0 + T; \boldsymbol{x}, \tau, \boldsymbol{y}_{ref}(\cdot)\right)\right) \right\} \\ &= \max_{\tau \in [t, t_0+T]} V_2(t, \boldsymbol{x}) \end{aligned}$$

The value function V_2 can be used to obtain V_1 indirectly. The HJB PDE associated with V_2 is of a simpler form and does not have the extra maximization term $\max\{\cdot, 0\}$ like in eq(4-7). The principle of optimality states the following for V_2:

$$V_2(t, \boldsymbol{x}) = \sup_{\boldsymbol{y}_{ref}(\cdot)\in \boldsymbol{\mathcal{Y}_{ref}}_{[t,t_0+T]}} V_2\left(t+h, \boldsymbol{\phi}\left(t+h; \boldsymbol{x}, t, \boldsymbol{y}_{ref}(\cdot)\right)\right) \tag{4-11}$$

The HJB PDE associated to eq(4-11) is given in the following theorem.

Theorem 4.3
The value function $V_2 : [t_0, t_0+T] \times \mathbb{R}^n \mapsto \mathbb{R}$ is the unique, continuous viscosity solution of the following time-dependent Hamilton-Jacobi-Bellman PDE:

$$\frac{\partial V_2(t,\boldsymbol{x})}{\partial t} + H\left(\boldsymbol{x}, \frac{\partial V_2(t,\boldsymbol{x})}{\partial \boldsymbol{x}}^T\right) = 0 \tag{4-12a}$$

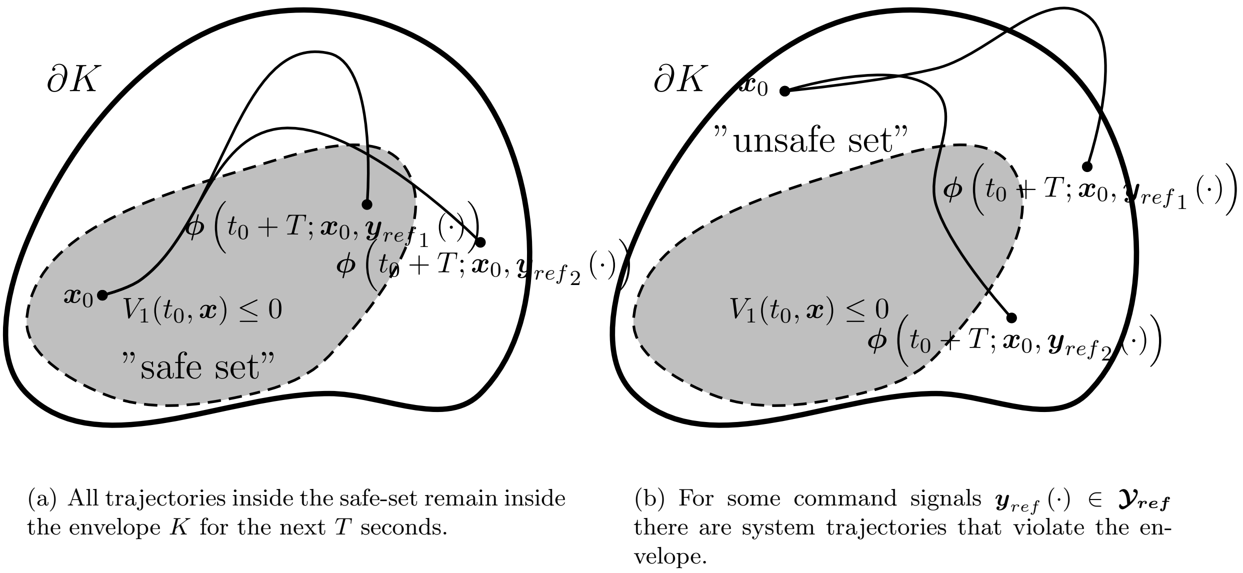
$$V_2(t_0+T, \boldsymbol{x}) = l(\boldsymbol{x}) \tag{4-12b}$$

where the Hamiltonian H is again defined by eq(4-8), i.e.

$$H(\boldsymbol{x}, \boldsymbol{p}) = \max_{\boldsymbol{y}_{ref}\in\boldsymbol{\mathcal{Y}_{ref}}} \boldsymbol{p}^T \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{y}_{ref}\right)$$

Proof. A proof can be found in (Fleming & Soner, 2006). □

Hence, computing $J^*(\boldsymbol{x}_0)$ can be done as follows. One first solves eq(4-12) to obtain V_2. Then, one applies eq(4-10) to obtain V_1. Finally, $J^*(\boldsymbol{x}_0)$ is obtained through evaluation of V_1 at $t = t_0$ and $\boldsymbol{x} = \boldsymbol{x}_0$.



(a) All trajectories inside the safe-set remain inside the envelope K for the next T seconds.

(b) For some command signals $\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}$ there are system trajectories that violate the envelope.

Figure 4-1: The safe-set for some given time-horizon T and command-signal space $\boldsymbol{\mathcal{Y}_{ref}}$.

4-2-3 Notion of the 'safe-set'

The sub-zero level set of V_1 at t_0, i.e. $\{\boldsymbol{x} \in \mathbb{R}^n \mid V_1(t_0, \boldsymbol{x}_0) \leq 0\}$ has a physical interpretation. It describes the collection of state conditions for which the system is guaranteed to stay inside the envelope for the next T seconds, irrespective of what command signal $\boldsymbol{y}_{ref}(\cdot) \in \boldsymbol{\mathcal{Y}_{ref}}$ is provided to the system. This state domain is referred to as the *safe-set*. The safe-set has the property that it can only shrink or maintain size as the prediction horizon T increases. Furthermore, by definition, the safe-set is a subset of (or equal to) the envelope K. Figure 4-1 illustrates the safe-set. The figure shows that a state, which is not part of the safe-set, will not necessarily violate the envelope in the next T seconds. Violations of the envelope occur only for some specific command signals of the set $\boldsymbol{\mathcal{Y}_{ref}}$.

4-3 The general procedure for estimating command margins

The objective is to check continuously whether the current state of the aircraft is inside the safe-set. This mathematically guarantees operation of the aircraft inside the state envelope. The contours of the safe-set is a function of the system dynamics, the current state and command margins.

The task is to keep the state of the aircraft within the safe-set at all times by continuously adjusting the command margins. The adjustment procedure is shown in figure 4-2. The most challenging step in the procedure outlined in figure 4-2 is the computation of the value function V_1. Analytic expressions for eq(4-5) (or eq(4-9) for that matter) are rarely found. Instead, numerical methods needs to be employed to solve the HJB PDE.

In the next chapter, level set methods are used to compute the value function. In chapters 6 and 7 a novel scheme is introduced to compute the value function with multivariate simplex splines.

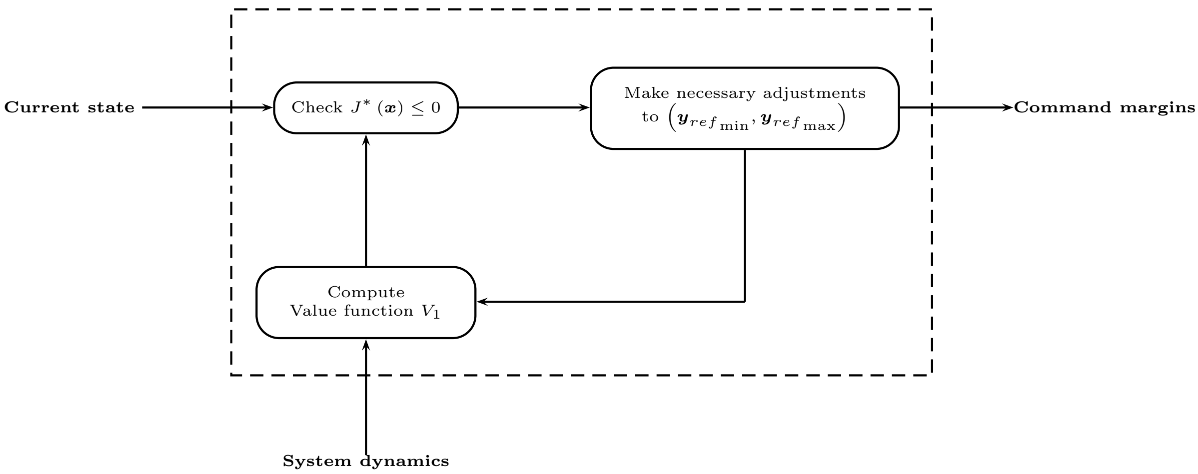


Figure 4-2: The procedure for finding safe command margins.

Chapter 5

Illustrations on a simplified aircraft system

5-1 Introduction

In this chapter, the safe command margins are estimated for the longitudinal pitch dynamics of a Generic Transport Model (GTM) (Jordan et al., 2004). The aircraft state envelope is constrained by limitations on the pitch attitude. The objective is to illustrate the working principles of the optimal control approach detailed in chapter 4 on a concrete example. The command margins are estimated in an *off-line* setting using level set methods.

The chapter is organized as follows. In section 5-2 the GTM pitch dynamics is approximated with a second-order linear system. Section 5-3 discusses the implementation of the optimal control approach on the pitch dynamics model for a state envelope with limitations on the pitch attitude. Section 5-4 presents the results obtain off-line using high-accuracy level set algorithm.

5-2 The simplified pitch dynamics of a Generic Transport Model

The longitudinal pitch dynamics of the GTM is approximated with a second-order linear system. The approximation describes the short period motion and omits the much slower phugoid motion. The state of the system is described by the pitch angle θ and pitch rate q. The input to the system is the elevator deflection δ_e.

5-2-1 The nominal system

Let the state be denoted by $\boldsymbol{x} = \left[\theta, q\right]^T$ and the input by $\boldsymbol{u} = \delta_e$. Under nominal conditions, the aircraft dynamics are:

$$\dot{\boldsymbol{x}} = \mathrm{A}_0\boldsymbol{x} + \mathrm{B}_0\boldsymbol{u}, \quad \boldsymbol{u}_{\min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max} \tag{5-1}$$

where[1]:

$$\mathrm{A}_0 = \begin{bmatrix} 0 & 1 \\ -2.6923 & -0.7322 \end{bmatrix}, \quad \mathrm{B}_0 = \begin{bmatrix} 0 \\ -3.3552 \end{bmatrix}$$

and

$$\boldsymbol{u}_{\min} = -30^\circ, \quad \boldsymbol{u}_{\max} = 30^\circ$$

The natural frequency of the nominal system ω_{n_0} is equal to 1.64 rad/s and the damping ratio ς_0 is 0.223.

A pitch command tracker is designed for the nominal system eq(5-1). The pitch command tracker is a PD controller and has the form:

$$\delta_e = k_\theta \left(k_s \theta_{ref} - \theta\right) + k_q \left(0 - q\right)$$

Let $K_1 = \begin{bmatrix} k_\theta & k_q \end{bmatrix}$ and $K_2 = k_\theta k_s$, such that:

$$\delta_e = -K_1 \boldsymbol{x} + K_2 \theta_{ref} \tag{5-2}$$

In eq(5-2), θ_{ref} is reference pitch attitude. Stability requirements demand the natural frequency ω_{n_r} to be 2.5 rad/s and the damping ratio ς_r to be 0.707. Pole-placement yields:

$$K_1 = \begin{bmatrix} -1.0604 & -0.8354 \end{bmatrix} \tag{5-3}$$

The gain K_2 is used to eliminate the steady-state error for a step reference command. For the nominal system, this gain is set to -1.8628. The saturation of the elevator introduces non-linear effects to an otherwise completely linear system. Within the saturation bounds, the dynamics are given by:

$$\dot{\boldsymbol{x}} = \left(\mathrm{A}_0 - \mathrm{B}_0 K_1\right) \boldsymbol{x} + \mathrm{B}_0 K_2 \theta_{ref}$$

5-2-2 The off-nominal system

Many different failure scenarios can be considered for the GTM. This report restricts the analysis to only one (representative) off-nominal condition in which the open-loop dynamics become *marginally* stable. Furthermore, a 50% *reduction* in elevator effectivenes is present. The dynamics of the off-nominal condition are:

$$\dot{\boldsymbol{x}} = \mathrm{A}\boldsymbol{x} + \mathrm{B}\boldsymbol{u}, \quad \boldsymbol{u}_{\min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max} \tag{5-4}$$

where:

$$\mathrm{A} = \begin{bmatrix} 0 & 1 \\ -2.3388 & -0.0252 \end{bmatrix}, \quad \mathrm{B} = \begin{bmatrix} 0 \\ -1.7676 \end{bmatrix}$$

The natural frequency for the off-nominal system ω_n is 1.53 rad/s and the damping ratio ς is reduced to 0.0083.

Note that the pitch attitude tracker eq(5-2) is non-adaptive and hence the gains K_1 and K_2 do not get modified. Apart from from a sluggish response(i.e. high overshoot, large settling time, etc), a steady-state error in the pitch command tracker can also be expected. Within the linear region of the system, the dynamics are given by:

$$\dot{\boldsymbol{x}} = \left(\mathrm{A} - \mathrm{B}K_1\right) \boldsymbol{x} + \mathrm{B}K_2 \theta_{ref}$$

[1] The matrices A_0 and B_0 are given for θ, q and δ_e expressed in radians.

5-3 Off-line implementation of the optimal control formulation

The command margins $(\theta_{ref,\min}, \theta_{ref,\max})$ are estimated for a state envelope defined in terms of limits on the pitch attitude:

$$K := \left\{ (\theta, q) \in \mathbb{R}^2 \;\middle|\; \theta_{\min} \leq \theta \leq \theta_{\max} \right\} \tag{5-5}$$

In the simulations, $\theta_{\min} = -10°$ and $\theta_{\max} = 25°$.

Following the optimal control formulation of section 4-2, certain values are first assumed for $(\theta_{ref,\min}, \theta_{ref,\max})$ and J^* is evaluated for some initial condition $\boldsymbol{x}_0$. The value function V_1 is computed indirectly by solving for eq(4-12) first and then applying eq(4-10).

5-3-1 Selection of a suitable implicit surface representation

A suitable implicit surface representation is selected for the state envelope K. The implicit surface representation is used as a boundary condition for eq(4-12). For the example in this chapter, eq(3-8) is applied to yield:

$$l(\boldsymbol{x}) = \max\{-\theta - 10°, \theta - 25°\} \tag{5-6}$$

Notice that eq(5-6) is non-smooth at $\theta = 7.5°$. Consequently, one could anticipate that the value function is non-smooth at that location as well[2]. In other words, no classical solution exists for the problem. However, according to theorem 4.3, an unique viscosity solution can always be found for eq(4-12).

5-3-2 Evaluation of the Hamiltonian and optimal feedback control law

The HJB PDE is coupled with an optimization problem. The optimization consists of evaluating the Hamiltonian eq(4-8), i.e.

$$H(\boldsymbol{x}, \boldsymbol{p}) = \max_{\boldsymbol{y}_{ref} \in \mathrm{Y}_{ref}} \boldsymbol{p}^T \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{y}_{ref}\right)$$

In the example problem: $\boldsymbol{y}_{ref}$ and Y_{ref} in eq(4-8) are respectively replaced by θ_{ref} and $[\theta_{ref,\min}, \theta_{ref,\max}]$.

Next, an analytic expression for the closed-loop optimal control law $\boldsymbol{g}^* : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ (which optimizes the Hamiltonian) is found. Let:

$$\boldsymbol{f}_0(\boldsymbol{x}, \theta_{ref}) = \begin{cases} \mathrm{A}_0\boldsymbol{x} + \mathrm{B}_0\delta_{e,\min} & \text{if } -K_1\boldsymbol{x} + K_2\theta_{ref} < \delta_{e,\min} \\ \mathrm{A}_0\boldsymbol{x} + \mathrm{B}_0\delta_{e,\max} & \text{if } -K_1\boldsymbol{x} + K_2\theta_{ref} > \delta_{e,\max} \\ (\mathrm{A}_0 - \mathrm{B}_0K_1)\boldsymbol{x} + \mathrm{B}K_2\theta_{ref} & \text{otherwise} \end{cases} \tag{5-7}$$

denote the dynamics of the nominal system. Similarly, let:

$$\boldsymbol{f}(\boldsymbol{x}, \theta_{ref}) = \begin{cases} \mathrm{A}\boldsymbol{x} + \mathrm{B}\delta_{e,\min} & \text{if } -K_1\boldsymbol{x} + K_2\theta_{ref} < \delta_{e,\min} \\ \mathrm{A}\boldsymbol{x} + \mathrm{B}\delta_{e,\max} & \text{if } -K_1\boldsymbol{x} + K_2\theta_{ref} > \delta_{e,\max} \\ (\mathrm{A} - \mathrm{B}K_1)\boldsymbol{x} + \mathrm{B}K_2\theta_{ref} & \text{otherwise} \end{cases} \tag{5-8}$$

[2]In fact, even when the boundary condition is smooth, the value function can still develop discontinuities in its partial derivatives.

denote the dynamics of the off-nominal system. The Hamiltonians are respectively:

$$H_0(\boldsymbol{x},\boldsymbol{p}) = \max_{\theta_{ref}\in\left[\theta_{ref,\min},\theta_{ref,\max}\right]} \boldsymbol{p}^T \boldsymbol{f}_0(\boldsymbol{x},\theta_{ref}) \tag{5-9}$$

and

$$H(\boldsymbol{x},\boldsymbol{p}) = \max_{\theta_{ref}\in\left[\theta_{ref,\min},\theta_{ref,\max}\right]} \boldsymbol{p}^T \boldsymbol{f}(\boldsymbol{x},\theta_{ref}) \tag{5-10}$$

Observing the fact that the optimization variable θ_{ref} in eq(5-9) (and eq(5-10)) is affine to $\boldsymbol{x}$ and $\boldsymbol{p}$, the expressions:

$$\boldsymbol{g^*}_0(\boldsymbol{x},\boldsymbol{p}) = \begin{cases} \theta_{ref,min} & \text{if } \boldsymbol{p}^T \mathrm{B}_0 K_2 \leq 0 \\ \theta_{ref,max} & \text{if } \boldsymbol{p}^T \mathrm{B}_0 K_2 > 0 \end{cases} \tag{5-11}$$

and

$$\boldsymbol{g}^*(\boldsymbol{x},\boldsymbol{p}) = \begin{cases} \theta_{ref,min} & \text{if } \boldsymbol{p}^T \mathrm{B} K_2 \leq 0 \\ \theta_{ref,max} & \text{if } \boldsymbol{p}^T \mathrm{B} K_2 > 0 \end{cases} \tag{5-12}$$

optimize respectively the Hamiltonians eq(5-9) and eq(5-10). The Hamiltonian is maximized along the gradient of the value function, i.e. $\boldsymbol{p} = (\partial V/\partial \boldsymbol{x})^T$. From eq(5-11) and eq(5-12) it follows that the optimal control $\boldsymbol{y}^*_{ref}(\cdot)$ is a *bang-bang* signal.

5-3-3 Numerical solutions of the HJB PDE using level set methods

Analytic solutions are rarely found for eq(4-12). This holds also true for the example considered in this chapter. However, well established numerical methods exist for solving the HJB PDE with finite-difference techniques. These algorithms go under the name of level set methods and are specifically designed to obtain viscosity solutions of Hamilton-Jacobi PDEs. The numerical schemes approximate the solution of the PDE on a fixed Cartesian grid and the approximation is known to converge to the true solution as the grid gets more refined. Appendix A covers the details of the algorithm regarding their application to eq(4-12). In this thesis, extensive use is made of the level set toolbox of (Mitchell, 2007) for implementing the algorithms in Matlab.

Note that Level set methods are applied in an off-line setting and suffer severely from the curse-of-dimensionality. The exponential growth of the grid-size with respect to the state dimension limits the applicability of the method to low dimensional problems. The curse-of-dimensionality is an inherent problem in DP, and, in the next two chapters, a new method is proposed to approximate the value function with principles from Neuro-Dynamic Programming (NDP). The new method is used to reduce computational costs substantially, such that, command margins can be estimated for aircraft in a real-time environment.

5-4 Results

The value function V_1 is computed for different command margins and time-horizons. The value function is approximated with a level set algorithm. The spatial derivatives are approximated with a *third-order ENO* scheme, a *LF approximation* is used for the Hamiltonian, and, the value function is integrated in time with an *explicit third-order RK method.* The value function is solved over the domain $\Omega = [-0.3491, 0.8727] \times [-1.5, 1.5]$ rad. The accuracy of the grid are $\Delta\theta = \frac{2}{180}\pi$ rad and $\Delta q = \frac{2}{180}\pi$ rad. In the following sections, the results obtained are analyzed for both the nominal and off-nominal conditions.

5-4-1 The safe-set for different command margins

In figure 5-1 the safe-set of the nominal system is shown for three different command margins. In figure 5-2 the same is done for the off-nominal system. For both figures the time-horizon is set to three seconds. The figures illustrate the dependency of the safe-set on the command margins. The safe-set tends to shrink as the command margins are more relaxed. Furthermore, the safe-sets of the nominal system are significantly larger than their off-nominal counterparts.

5-4-2 The safe-set for different time-horizons

In figure 5-3 and figure 5-4 the safe-set is shown for different time-horizons. The command margins are set to $\theta_{ref,\min} = -5°$ and $\theta_{ref,\max} = 20°$. Figure 5-3 shows that the safe-set for the nominal system does not shrink after $T = 1$ second. This indicates that the transients in the dynamics are already captured sufficiently in a one second time window. For the off-nominal system, this is however no longer true: figure 5-4 shows that the safe-set shrinks even after $T = 3$ seconds. A larger time-horizon is required to capture the complete dynamics. This can be expected since for the off-nominal system the time constant is much larger.

5-4-3 System state trajectories inside and outside of the safe-set

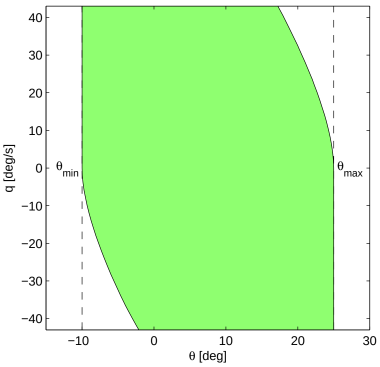
In Figure 5-5 trajectories of the system are shown in a phase portrait. The trajectories shown in the plots are in response to some randomly generated bang-bang pitch command signals. The signals have a two seconds duration and are bounded by: $\theta_{ref,\min} = -8°$ and $\theta_{ref,\max} = 23°$. The left plot on the figures show trajectories of the system for some initial condition inside the safe-set, the plots on the right show trajectories for some initial condition outside the safe-set (but still inside the envelope). The backdrop shows the corresponding safe-sets. In figure 5-5(a), where the system is nominal, the trajectories do not appear to leave the safe-set once it has entered it. The safe-set is an positively invariant set(Blanchini, 1999) for the system. For the off-nominal system, the trajectories however do appear to leave the safe-set. The safe-set is no longer positively invariant and if the system starts somewhere inside the safe-set, although it is guaranteed to stay inside the envelope, the system might leave the safe-set within the next two seconds and enter the "unsafe" portion of the envelope. Hence, for the same command margins, the envelope can be violated for a larger time-horizon.

5-4-4 'Safe' command margins for the nominal and off-nominal system

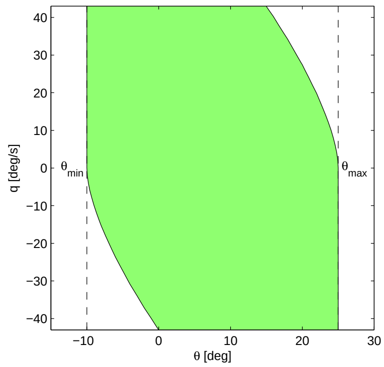
Fix the time-horizon is to some time T and evaluate J^* at some initial condition $\boldsymbol{x}_0$ for many different command margins $(\theta_{ref,\min}, \theta_{ref,\max})$. In figure 5-6 and figure 5-7 the results are shown for a time-horizon of 5 seconds. The green regions in the figures depict the "safe"[3]

[3] "Safe" refers here to the fact that the margins guarantee operation within the envelope for the specified time window

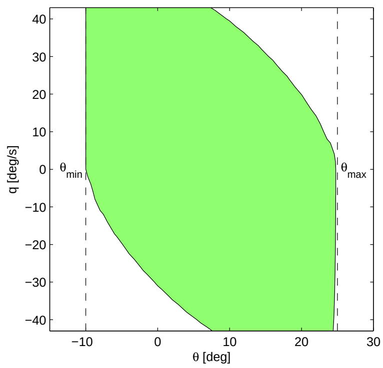
command margins for a the specified initial conditions. The command margins are strongly dependent on the initial condition and system dynamics. This dependency is illustrated also in figure 4-2.



(a) $\theta_{ref,\min} = -2°$, $\theta_{ref,\max} = 17°$

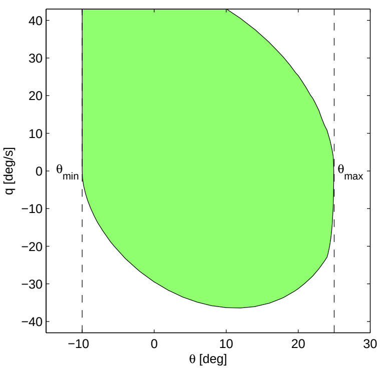


(b) $\theta_{ref,\min} = -5°$, $\theta_{ref,\max} = 20°$

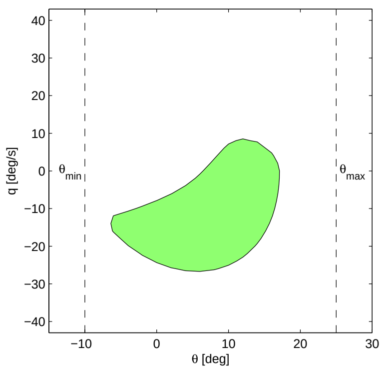


(c) $\theta_{ref,\min} = -8°$, $\theta_{ref,\max} = 23°$

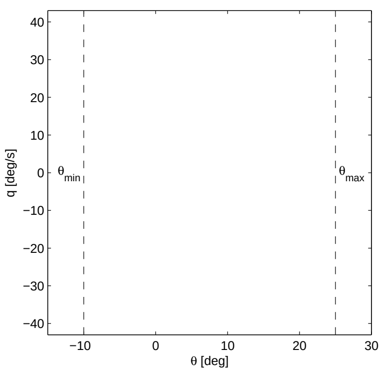
Figure 5-1: The safe-set for different command margins. The dynamics are nominal and the time-horizon is set to 3 seconds.



(a) $\theta_{ref,\min} = -2°$, $\theta_{ref,\max} = 17°$

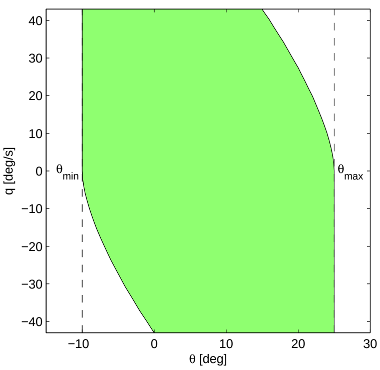


(b) $\theta_{ref,\min} = -5°$, $\theta_{ref,\max} = 20°$

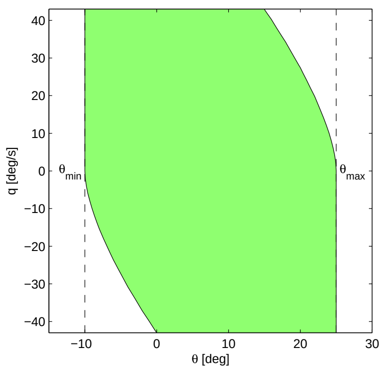


(c) $\theta_{ref,\min} = -8°$, $\theta_{ref,\max} = 23°$

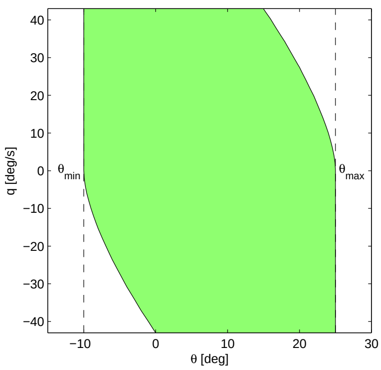
Figure 5-2: The safe-set for different command margins. The dynamics are off-nominal and the time-horizon is set to 3 seconds.



(a) $T = 1$ s

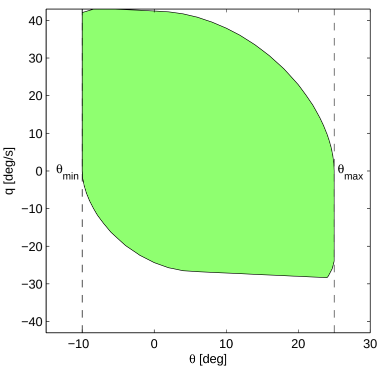


(b) $T = 3$ s

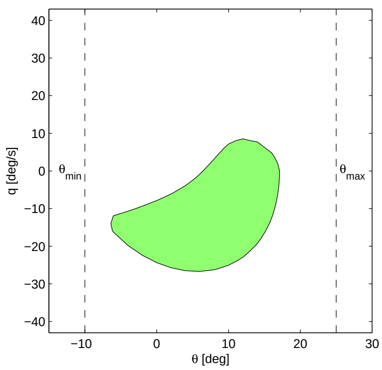


(c) $T = 5$ s

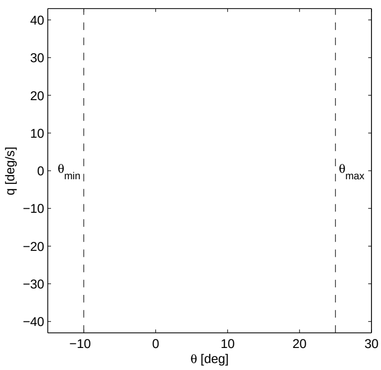
Figure 5-3: The safe-set for different time-horizons. The dynamics are nominal and the command margins are respectively $\theta_{ref,\min} = -5^\circ$ and $\theta_{ref,\max} = 20^\circ$



(a) $T = 1$ s

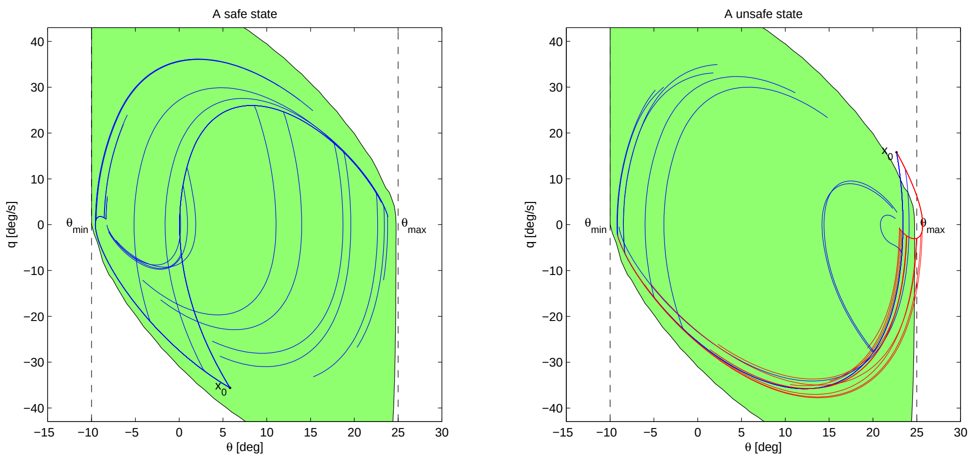


(b) $T = 3$ s

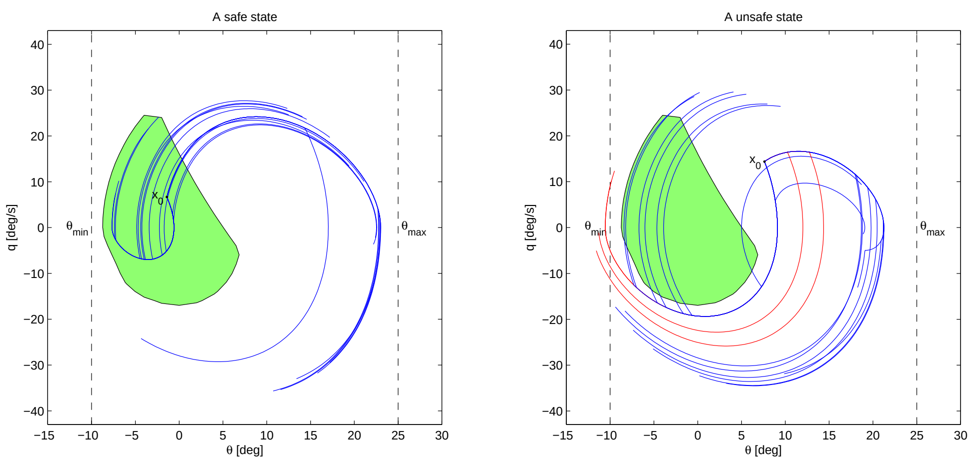


(c) $T = 5$ s

Figure 5-4: The safe-set for different time-horizons. The dynamics are off-nominal and the command margins are respectively $\theta_{ref,\min} = -5°$ and $\theta_{ref,\max} = 20°$

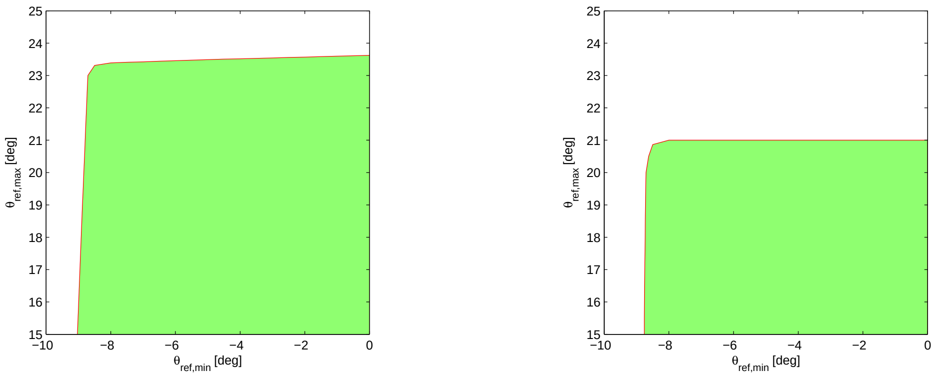


(a) Nominal case



(b) Off-nominal case

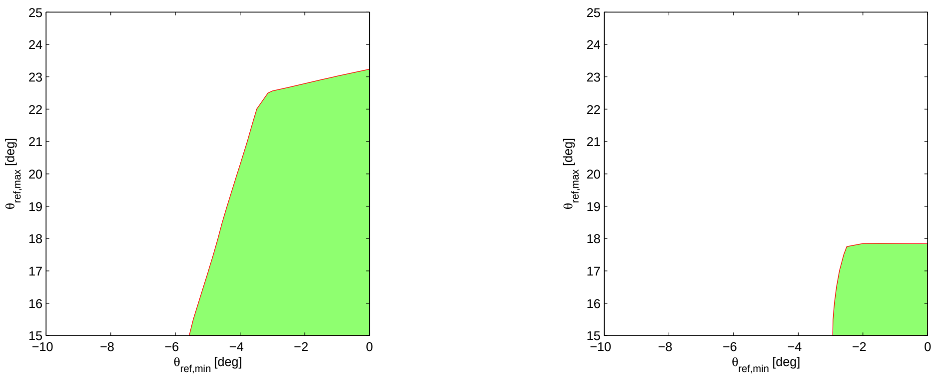
Figure 5-5: Possible state trajectories of the system plotted for some initial conditions in response to certain randomly generated bang-bang signals. The command margins are respectively $\theta_{ref,\min} = -8°$ and $\theta_{ref,\max} = 23°$. The green area indicates the safe-set for a time-horizon of 2 seconds.



(a) $\theta_0 = 0^\circ$, $q_0 = 0^\circ/\mathrm{s}$

(b) $\theta_0 = 20^\circ$, $q_0 = 25^\circ/\mathrm{s}$

Figure 5-6: The "safe" command margins (depicted in green) for the nominal system at different initial conditions. The time-horizon is set to 5 seconds.



(a) $\theta_0 = 0^\circ$, $q_0 = 0^\circ/\mathrm{s}$

(b) $\theta_0 = 20^\circ$, $q_0 = 25^\circ/\mathrm{s}$

Figure 5-7: The "safe" command margins (depicted in green) for the off-nominal system at different initial conditions. The time-horizon is set to 5 seconds.

Part III

Approximation of the value function in a real-time setting

Chapter 6

A Neuro-Dynamic Programming approach using value-iteration

6-1 Introduction

The exponential growth of the grid-size limits the applicability of level set algorithms to relatively low dimensional problems. This motivates to look into alternative approaches to solve the optimal control problem such that the framework, detailed in chapter 4, can be implemented in real-time. Neuro-Dynamic Programming (NDP) attempts to beat the curse-of-dimensionality, inherent to DP problems, by approximating the value function with regression techniques (Bertsekas & Tsitsiklis, 1996). For instance, in (Djeridane & Lygeros, 2006), an attempt was made to approximate a time-dependent value function with multi-layer perceptrons. The value function considered by the paper was nearly identical to eq(4-5), except for the supremum being replaced by an infimum. The tuning of the network parameters, however, involved solving a complex, non-smooth optimization problem which was difficult to solve in real-time.

In this chapter, a novel method is presented to find approximate solutions to eq(4-5) using generic function approximators. The method relies on finding eq(4-5) indirectly through eq(4-9) by application of eq(4-10). An algebraic expression is first obtained for eq(4-12), which then is transcribed into a non-linear optimization problem. The non-linear optimization problem is subsequently solved as a sequence of least-squares problems by means of a value-iteration scheme.

The chapter is organized as follows. Section 6-2 discusses the discretization of eq(4-12). Section 6-3 transcribes the approximate algebraic expression of eq(4-12) into a non-linear optimization problem. Section 6-4 presents the value-iteration scheme, which later is used in section 6-5 to derive the final algorithm.

6-2 Discretization of the HJB PDE

The method of discretization for eq(4-12) is motivated by the approaches taken by (Alwardi, 2010) and (Huang, Wang, Chen, & Li, 2006). The PDE is first discretized in the state with generic function approximators, converting the PDE into an Ordinary Differential Equation (ODE) with a terminal condition. The ODE is subsequently discretized in time with an implicit integration scheme, yielding an algebraic matrix expression.

6-2-1 Discretization of the state

The HJB equation is a PDE coupled with an optimization. As a result, Eq(4-12) can be rewritten into the coupled equations[1]:

$$\begin{aligned} \frac{\partial V_2(t,\boldsymbol{x})}{\partial t} + \frac{\partial V_2(t,\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}^*(t,\boldsymbol{x})) &= 0 && \text{(6-1a)} \\ V_2(T,\boldsymbol{x}) &= l(\boldsymbol{x}) && \text{(6-1b)} \end{aligned}$$

and

$$\boldsymbol{g}^*(t,\boldsymbol{x}) = \arg \max_{\boldsymbol{y}_{ref} \in \Upsilon_{ref}} \left\{ \frac{\partial V_2(t,\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{y}_{ref}\right) \right\} \tag{6-2}$$

where eq(6-1) is referred to as the Boundary value Problem (BVP) and eq(6-2) is referred to as the Optimization Problem (OP).

Let $\hat{V}_2$ denote the approximate solution for eq(4-12). The approximate solution is assumed to take the following form:

$$\hat{V}_2(t,\boldsymbol{x}) = l(\boldsymbol{x}) + \sum_{i=1}^{p} \phi_i(\boldsymbol{x}) c_i(t) \tag{6-3}$$

In eq(6-3), $\phi_i(\boldsymbol{x})$ represent a set of functions that collectively have high approximation power. Furthermore, $c_i(t)$ are the unknown time-dependent coefficients which need to be estimated. Notice also that the term: $l(\boldsymbol{x})$ denotes the boundary condition of eq(6-1). Unlike in the method of (Alwardi, 2010), the boundary term is added to the expression so that the approximate solution will satisfy the boundary condition by construction.

Several different candidate solutions could have been chosen to approximate eq(4-12). The benefit of eq(6-3) is that it describes the physics of the underlying PDE. Basically, eq(6-3) describes an evolving surface over time. However, unlike in level set methods, the surface is defined continuously by a linear combination of basis functions.

Remark: *This chapter describes the proposed method generically for any type of function approximator which is linear in the parameters. In chapter 7, multivariate simplex spline theory(Lai & Schumaker, 2007) is used to find a polynomial basis for $\phi_i(\boldsymbol{x})$.*

The approximation $\hat{V}_2$ is related with V_2 by:

$$V_2(t,\boldsymbol{x}) = \hat{V}_2(t,\boldsymbol{x}) + \epsilon(t,\boldsymbol{x}) \tag{6-4}$$

where $\epsilon(t,\boldsymbol{x})$ denotes the approximation error. The following assumption is made.

[1] Without any loss-of-generality, it is assumed that $t_0 = 0$

Assumption: *For large numbers of basis functions, i.e. when p is sufficiently large, the approximation error ϵ in eq(6-4) becomes negligible.*

The above assumption basically states that the function approximator is able to accurately describe the shape of the value function.

Next, observe that:

$$\frac{\partial V_2(t,\boldsymbol{x})}{\partial t} = \sum_{i=1}^{p} \phi_i(\boldsymbol{x})\,\dot{c}_i(t) \qquad \text{and} \qquad \frac{\partial V_2(t,\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial l(\boldsymbol{x})}{\partial \boldsymbol{x}} + \sum_{i=1}^{p} \frac{\partial \phi_i(\boldsymbol{x})}{\partial \boldsymbol{x}} c_i(t)$$

Substitution of these exprresions into eq(6-1) and eq(6-2) yield respectively:

$$\sum_{i=1}^{p} \phi_i(\boldsymbol{x})\,\dot{c}_i(t) = -\left(\frac{\partial l(\boldsymbol{x})}{\partial \boldsymbol{x}} + \sum_{i=1}^{p} \frac{\partial \phi_i(\boldsymbol{x})}{\partial \boldsymbol{x}} c_i(t)\right) \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}^*(t,\boldsymbol{x})) \tag{6-5a}$$

$$c_i(T) = 0, \qquad i = 1, 2, \ldots, p \tag{6-5b}$$

and

$$\boldsymbol{g}^*(t,\boldsymbol{x}) = \arg \max_{\boldsymbol{y}_{ref} \in \Upsilon_{ref}} \left\{ \left(\frac{\partial l(\boldsymbol{x})}{\partial \boldsymbol{x}} + \sum_{i=1}^{p} \frac{\partial \phi_i(\boldsymbol{x})}{\partial \boldsymbol{x}} c_i(t)\right) \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{y}_{ref}\right) \right\} \tag{6-6}$$

Notice that the equality sign in eq(6-5) is actually an immediate result of the assumption made on eq(6-4) regarding the error term $\epsilon(t,\boldsymbol{x})$.

Nevertheless, the following concise notations are introduced for eq(6-5):

$$\boldsymbol{F}(\boldsymbol{x})^T \dot{\boldsymbol{c}}(t) = -\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}^*(t,\boldsymbol{x}))^T \left(\frac{\partial l(\boldsymbol{x})}{\partial \boldsymbol{x}}^T + \frac{\partial \boldsymbol{F}(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \boldsymbol{c}(t)\right) \tag{6-7a}$$

$$\boldsymbol{c}(T) = 0, \tag{6-7b}$$

and eq(6-6):

$$\boldsymbol{g}^*(t,\boldsymbol{x}) = \arg \max_{\boldsymbol{y}_{ref} \in \Upsilon_{ref}} \left\{ \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{y}_{ref}\right)^T \left(\frac{\partial l(\boldsymbol{x})}{\partial \boldsymbol{x}}^T + \frac{\partial \boldsymbol{F}(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \boldsymbol{c}(t)\right) \right\} \tag{6-8}$$

where $\boldsymbol{F}(\boldsymbol{x})$ is defined as:

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{bmatrix} \phi_1(\boldsymbol{x}) & \phi_2(\boldsymbol{x}) & \cdots & \phi_p(\boldsymbol{x}) \end{bmatrix}^T \tag{6-9}$$

and $\boldsymbol{c}(t)$ as:

$$\boldsymbol{c}(t) = \begin{bmatrix} c_1(t) & c_2(t) & \cdots & c_p(t) \end{bmatrix}^T \tag{6-10}$$

6-2-2 Discretization in time

The ODE in eq(6-5) is discretized in time with an implicit integration scheme. Implicit schemes are preferred over explicit ones for their stability properties. Forward-Euler or higher-order explicit Runge-Kutta schemes are known to be only conditionally stable. In order to ensure stability, severe restrictions have to be imposed on the time step Δt in the discretization. Implicit schemes on the other hand, do not suffer from such limitations. For instance, the Backward-Euler $\mathcal{O}(\Delta t)$ and Crank-Nicolson scheme $\mathcal{O}\left(\Delta t^2\right)$ are known to be unconditionally stable. These benefits do not come without a price because implicit schemes are more complicated to solve. What follows next is a derivation of the algebraic expressions that result from the application of the Backward-Euler scheme and Crank-Nicolson scheme.

First, let the time-dependent coefficients $\boldsymbol{c}(t)$ be discretized by:

$$\boldsymbol{c}_k = \boldsymbol{c}(k\Delta t), \qquad k = 0, 1, 2, \ldots N \tag{6-11}$$

where $\Delta t = T/(N+1)$ and N some positive integer. Furthermore, define:

$$\boldsymbol{g}^*(k\Delta t, \boldsymbol{x}) \equiv \boldsymbol{g}^*(\boldsymbol{c}_k, \boldsymbol{x}) = \arg \max_{\boldsymbol{y}_{ref} \in \Upsilon_{ref}} \left\{ \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_{ref})^T \left(\frac{\partial l(\boldsymbol{x})}{\partial \boldsymbol{x}}^T + \frac{\partial \boldsymbol{F}(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \boldsymbol{c}_k \right) \right\} \tag{6-12}$$

Application of the Backward-Euler scheme to eq(6-7) yields:

$$\boldsymbol{F}(\boldsymbol{x})^T \frac{\boldsymbol{c}_{k-1} - \boldsymbol{c}_k}{\Delta t} = -\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}^*(\boldsymbol{c}_{k-1}, \boldsymbol{x}))^T \left(\frac{\partial l(\boldsymbol{x})}{\partial \boldsymbol{x}}^T + \frac{\partial \boldsymbol{F}(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \boldsymbol{c}_{k-1} \right)$$

Observing that the coefficients $\boldsymbol{c}_{k-1}$ cannot be expressed explicitly as a relation of $\boldsymbol{c}_k$, application of the Backward-Euler scheme gives rise to the following implicit sequence:

$$\alpha_1(\boldsymbol{x}, \boldsymbol{c}_{k-1}) \boldsymbol{c}_{k-1} + \alpha_2(\boldsymbol{x}) \boldsymbol{c}_k = \beta(\boldsymbol{x}, \boldsymbol{c}_{k-1}), \qquad \boldsymbol{c}_N = 0 \tag{6-13}$$

where:

$$\alpha_1(\boldsymbol{x}, \boldsymbol{c}_{k-1}) = \boldsymbol{F}(\boldsymbol{x})^T + \Delta t \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}^*(\boldsymbol{c}_{k-1}, \boldsymbol{x}))^T \frac{\partial \boldsymbol{F}(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \tag{6-14a}$$

$$\alpha_2(\boldsymbol{x}) = -\boldsymbol{F}(\boldsymbol{x})^T \tag{6-14b}$$

$$\beta(\boldsymbol{x}, \boldsymbol{c}_{k-1}) = -\Delta t (\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}^*(\boldsymbol{c}_{k-1}, \boldsymbol{x})))^T \frac{\partial l(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \tag{6-14c}$$

Similarly, direct application of the Cranck-Nicolson scheme to eq(6-7) will yield:

$$\begin{aligned} \boldsymbol{F}(\boldsymbol{x})^T \frac{\boldsymbol{c}_{k-1} - \boldsymbol{c}_k}{\Delta t} = -\frac{1}{2} \Bigg\{ & \left[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}^*(\boldsymbol{c}_{k-1}, \boldsymbol{x}))^T \left(\frac{\partial l(\boldsymbol{x})}{\partial \boldsymbol{x}}^T + \frac{\partial \boldsymbol{F}(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \boldsymbol{c}_{k-1} \right) \right] \\ & + \left[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}^*(\boldsymbol{c}_k, \boldsymbol{x}))^T \left(\frac{\partial l(\boldsymbol{x})}{\partial \boldsymbol{x}}^T + \frac{\partial \boldsymbol{F}(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \boldsymbol{c}_k \right) \right] \Bigg\} \end{aligned}$$

leading to the following implicit sequence:

$$\alpha_1(\boldsymbol{x}, \boldsymbol{c}_{k-1}) \boldsymbol{c}_{k-1} + \alpha_2(\boldsymbol{x}, \boldsymbol{c}_k) \boldsymbol{c}_k = \beta(\boldsymbol{x}, \boldsymbol{c}_{k-1}, \boldsymbol{c}_k), \qquad \boldsymbol{c}_N = 0 \tag{6-15}$$

where:

$$\alpha_1(\boldsymbol{x}, \boldsymbol{c}_{k-1}) = \boldsymbol{F}(\boldsymbol{x})^T + \frac{\Delta t}{2} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}^*(\boldsymbol{c}_{k-1}, \boldsymbol{x}))^T \frac{\partial \boldsymbol{F}(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \tag{6-16a}$$

$$\alpha_2(\boldsymbol{x}, \boldsymbol{c}_k) = -\boldsymbol{F}(\boldsymbol{x})^T + \frac{\Delta t}{2} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}^*(\boldsymbol{c}_k, \boldsymbol{x}))^T \frac{\partial \boldsymbol{F}(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \tag{6-16b}$$

$$\beta(\boldsymbol{x}, \boldsymbol{c}_{k-1}, \boldsymbol{c}_k) = -\frac{\Delta t}{2} (\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}^*(\boldsymbol{c}_{k-1}, \boldsymbol{x})) + \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}^*(\boldsymbol{c}_k, \boldsymbol{x})))^T \frac{\partial l(\boldsymbol{x})}{\partial \boldsymbol{x}}^T \tag{6-16c}$$

In (Alwardi, 2010), the term: $\boldsymbol{g}^*(\boldsymbol{c}_{k-1}, \boldsymbol{x})$ in eq(6-14) and eq(6-16) was approximated with $\boldsymbol{g}^*(\boldsymbol{c}_k, \boldsymbol{x})$, so that the coefficients $\boldsymbol{c}_k$ can found through backward recursion of the sequences in eq(6-13) or eq(6-15).

An alternative approach is to place the sequences in one big system of algebraic equations and find the coefficients through that route. For in the case Back-Euler discretization, this system of algebraic equations would be:

$$\begin{aligned}
\alpha_1(\boldsymbol{x}, \boldsymbol{c}_0)\,\boldsymbol{c}_0 + \alpha_2(\boldsymbol{x})\,\boldsymbol{c}_1 &= \beta(\boldsymbol{x}, \boldsymbol{c}_0) \\
\alpha_1(\boldsymbol{x}, \boldsymbol{c}_1)\,\boldsymbol{c}_1 + \alpha_2(\boldsymbol{x})\,\boldsymbol{c}_2 &= \beta(\boldsymbol{x}, \boldsymbol{c}_1) \\
&\vdots \\
\alpha_1(\boldsymbol{x}, \boldsymbol{c}_{N-2})\,\boldsymbol{c}_{N-2} + \alpha_2(\boldsymbol{x})\,\boldsymbol{c}_{N-1} &= \beta(\boldsymbol{x}, \boldsymbol{c}_{N-2}) \\
\alpha_1(\boldsymbol{x}, \boldsymbol{c}_{N-1})\,\boldsymbol{c}_{N-1} + \alpha_2(\boldsymbol{x})\,0 &= \beta(\boldsymbol{x}, \boldsymbol{c}_{N-1})
\end{aligned}$$

which is expressed in a matrix-vector notation:

$$\mathcal{A}(\boldsymbol{x}, \boldsymbol{C})\,\boldsymbol{C} = \mathcal{B}(\boldsymbol{x}, \boldsymbol{C}) \tag{6-17}$$

where $\boldsymbol{C} \in \mathbb{R}^{pN}$ denotes:

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}_0 \\ \vdots \\ \boldsymbol{c}_{N-2} \\ \boldsymbol{c}_{N-1} \end{bmatrix} \tag{6-18}$$

and $\mathcal{A} \in \mathbb{R}^{N \times pN}$, $\mathcal{B} \in \mathbb{R}^N$ denotes:

$$\mathcal{A}(\boldsymbol{x}, \boldsymbol{C}) = \begin{bmatrix} \alpha_1(\boldsymbol{x}, \boldsymbol{c}_0) & \alpha_2(\boldsymbol{x}) & & \\ & \ddots & \ddots & \\ & & \alpha_1(\boldsymbol{x}, \boldsymbol{c}_{N-2}) & \alpha_2(\boldsymbol{x}) \\ & & & \alpha_1(\boldsymbol{x}, \boldsymbol{c}_{N-1}) \end{bmatrix} \tag{6-19a}$$

$$\mathcal{B}(\boldsymbol{x}, \boldsymbol{C}) = \begin{bmatrix} \beta(\boldsymbol{x}, \boldsymbol{c}_0) \\ \vdots \\ \beta(\boldsymbol{x}, \boldsymbol{c}_{N-2}) \\ \beta(\boldsymbol{x}, \boldsymbol{c}_{N-1}) \end{bmatrix} \tag{6-19b}$$

A similar expression expression for the Crank-Nicolson discretization can be found. Infact, the notation would be identical to eq(6-17), except that $\mathcal{A} \in \mathbb{R}^{N \times pN}$ and $\mathcal{B} \in \mathbb{R}^N$ are now defined by:

$$\mathcal{A}(\boldsymbol{x}, \boldsymbol{C}) = \begin{bmatrix} \alpha_1(\boldsymbol{x}, \boldsymbol{c}_0) & \alpha_2(\boldsymbol{x}, \boldsymbol{c}_1) & & \\ & \ddots & \ddots & \\ & & \alpha_1(\boldsymbol{x}, \boldsymbol{c}_{N-2}) & \alpha_2(\boldsymbol{x}, \boldsymbol{c}_{N-1}) \\ & & & \alpha_1(\boldsymbol{x}, \boldsymbol{c}_{N-1}) \end{bmatrix} \tag{6-20a}$$

$$\mathcal{B}(\boldsymbol{x}, \boldsymbol{C}) = \begin{bmatrix} \beta(\boldsymbol{x}, \boldsymbol{c}_0, \boldsymbol{c}_1) \\ \vdots \\ \beta(\boldsymbol{x}, \boldsymbol{c}_{N-2}, \boldsymbol{c}_{N-1}) \\ \beta(\boldsymbol{x}, \boldsymbol{c}_{N-1}, 0) \end{bmatrix} \tag{6-20b}$$

where the terms: α_1, α_2, and β refer to the ones in eq(6-16).

Remark: *In comparison to Backward-Euler, the Crank-Nicolson scheme is of higher-order. However, the additional computational requirements associated with this scheme are negligible. Therefore, the remainder of this thesis considers only the Crank-Nicolson discretization for further studies.*

6-3 Transcription into a non-linear optimization problem

The algebraic expression in eq(6-17) can be transcribed into an optimization problem in the following sense.

Define the error function as:

$$\boldsymbol{e}\left(\boldsymbol{x}, \boldsymbol{C}\right) = \mathcal{A}\left(\boldsymbol{x}, \boldsymbol{C}\right) \boldsymbol{C} - \mathcal{B}\left(\boldsymbol{x}, \boldsymbol{C}\right) \tag{6-21}$$

Then eq(6-17) is transcribed into an optimization problem by minimization of the following integral[2]:

$$\min_{\boldsymbol{C} \in \mathbb{R}^{pN}} \int_{\mathbb{R}^n} \left\| \boldsymbol{e}\left(\eta, \boldsymbol{C}\right) \right\|^2 \mathrm{d}\eta$$

Observe that the above integral requires an integration over the entire state-space. This follows from the fact that V_2 is formally defined over the domain: $[0, T] \times \mathbb{R}^n$ (see theorem 4.3). In practice, one is interested in knowing the solution of the PDE only for a subset of this domain: $[0, T] \times K$.

Remark: *Strictly speaking, solving eq(4-12) for a local domain will require additional information on what happens at the boundary of this local domain (see also figure 6-1). The evolution of the surface is governed by the dynamics of the system. Consequently, the characteristic information that emanate from outer regions of the state-space propagate also into the solution of V_2 inside K. Henceforth, solving eq(4-12) over a local state domain automatically introduces errors in the numerical solution. As in (Alwardi, 2010), V_2 is solved over an extended region: $\Omega \supset K$ in order to minimize these artificially introduced errors.*

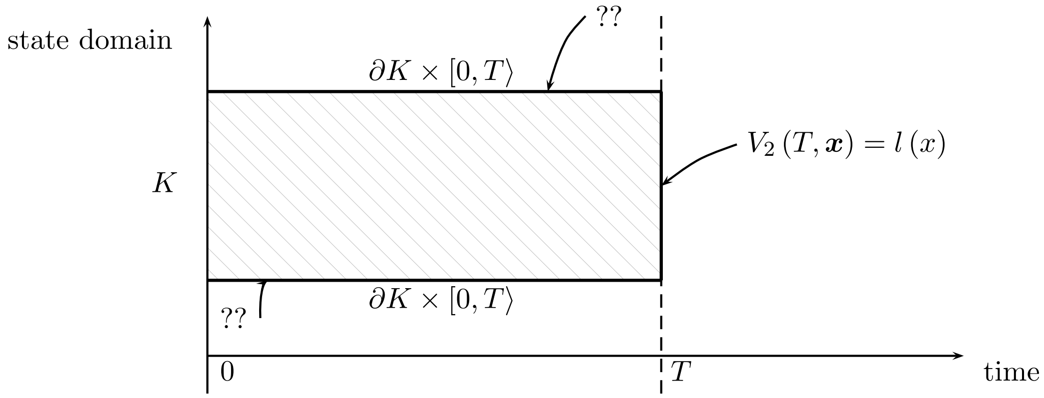


Figure 6-1: The boundary conditions which are needed to solve the HJB PDE locally. In the figure, ∂K denotes the boundary of the closed set K.

[2] $\|\cdot\|$ denotes here the standard 2-norm.

Apart from the local domain $[0, T] \times \Omega$, the optimization is done only for a selective number of L sample points in the state-space, i.e.

$$\boldsymbol{x}_l \in \Omega, \quad l = 1, 2, \ldots, L$$

In the literature, these sample points are referred to as *collocation points* and the technique is referred to as the *collocation method.* The collocation method effectively reduces the integral cost cost function into a summation:

$$O(\boldsymbol{C}) = \sum_{l=1}^{L} \|\boldsymbol{e}(\boldsymbol{x}_l, \boldsymbol{C})\|^2$$

where $O : \mathbb{R}^{pN} \mapsto \mathbb{R}$.

The cost function can be expressed also in a matrix-vector notation:

$$O(\boldsymbol{C}) = \|\mathrm{X}(\boldsymbol{C})\,\boldsymbol{C} - \mathrm{Y}(\boldsymbol{C})\|^2 \tag{6-22}$$

where $\mathrm{X} \in \mathbb{R}^{LN \times pN}$ and $\mathrm{Y} \in \mathbb{R}^{LN \times 1}$ defined by:

$$\mathrm{X}(\boldsymbol{C}) = \begin{bmatrix} \mathcal{A}(\boldsymbol{x}_1, \boldsymbol{C}) \\ \mathcal{A}(\boldsymbol{x}_2, \boldsymbol{C}) \\ \vdots \\ \mathcal{A}(\boldsymbol{x}_L, \boldsymbol{C}) \end{bmatrix} \tag{6-23a}$$

$$\mathrm{Y}(\boldsymbol{C}) = \begin{bmatrix} \mathcal{B}(\boldsymbol{x}_1, \boldsymbol{C}) \\ \mathcal{B}(\boldsymbol{x}_2, \boldsymbol{C}) \\ \vdots \\ \mathcal{B}(\boldsymbol{x}_L, \boldsymbol{C}) \end{bmatrix} \tag{6-23b}$$

Notice that since eq(6-22) is non-affine in terms of the arguments, the optimization problem:

$$\min_{\boldsymbol{C} \in \mathbb{R}^{pN}} O(\boldsymbol{C}) \tag{6-24}$$

will also be non-linear. The non-linearity is basically caused by the coupling of eq(6-1) with eq(6-2). Value-iteration can be used decouple this non-linear optimization problem into a series of least-squares problems.

6-4 Value-iteration for time-dependent HJB PDEs

Consider again BVP eq(6-1) and OP eq(6-2). Observe that $\boldsymbol{g}^*$ is the *optimal feed-back control law* associated with V_2, i.e. for any other feedback law $\boldsymbol{y}_{ref} = \boldsymbol{g}(t, \boldsymbol{x})$ the following holds:

$$V'_2(t, \boldsymbol{x}) \leq V_2(t, \boldsymbol{x}), \qquad \forall (t, \boldsymbol{x}) \in [0, T] \times \mathbb{R}^n$$

where $V'_2 : [0, T] \times \mathbb{R}^n \mapsto \mathbb{R}$ is the viscosity solution of:

$$\begin{aligned} \frac{\partial V'_2(t, \boldsymbol{x})}{\partial t} + \frac{\partial V'_2(t, \boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}(t, \boldsymbol{x})) &= 0 \\ V'_2(T, \boldsymbol{x}) &= l(\boldsymbol{x}) \end{aligned}$$

In other words, solving eq(4-12) is equivalent to finding an (not necessarily unique!) optimal feedback control law $\boldsymbol{g}^{*}(t,\boldsymbol{x})$ for eq(3-1).

Iterative approaches can be employed to decouple the BVP from the OP. In value-iteration, the decoupling is done by starting with some initial function $V_2^{(0)}$ for which a (sub-optimal) feedback law $\boldsymbol{y}_{ref} = \boldsymbol{g}^{(0)}(t,\boldsymbol{x})$ is determined via eq(6-2). This feedback law is subsequently used to compute an updated value function $V_2^{(1)}$. The process can be repeated indefinitely, which leads to the following iteration.

Definition 6.1: ***value-iteration (continuous time)***
Set $V_2^{(0)}(t,\boldsymbol{x}) = l(\boldsymbol{x})$*. For* $i = 1,2,3,\ldots$*, do:*

1. *Define:*

$$\boldsymbol{g}^{(i)}(t,x) = \arg \max_{\boldsymbol{y}_{ref}\in \Upsilon_{ref}} \left\{ \frac{\partial V_2^{(i-1)}(t,\boldsymbol{x})}{\partial x} \boldsymbol{f}\left(\boldsymbol{x},\boldsymbol{y}_{ref}\right) \right\} \tag{6-25}$$

2. *Define* $V_2^{(i)} : [0,T] \times \mathbb{R}^n \mapsto \mathbb{R}$ *as the viscosity solution of:*

$$\frac{\partial V_2^{(i)}(t,\boldsymbol{x})}{\partial t} + \frac{\partial V_2^{(i)}(t,\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}\left(\boldsymbol{x},\boldsymbol{g}^{(i)}(t,x)\right) = 0 \tag{6-26a}$$

$$V_2^{(i)}(T,\boldsymbol{x}) = l(\boldsymbol{x}) \tag{6-26b}$$

The iteration procedure is also illustrated in figure 6-2.

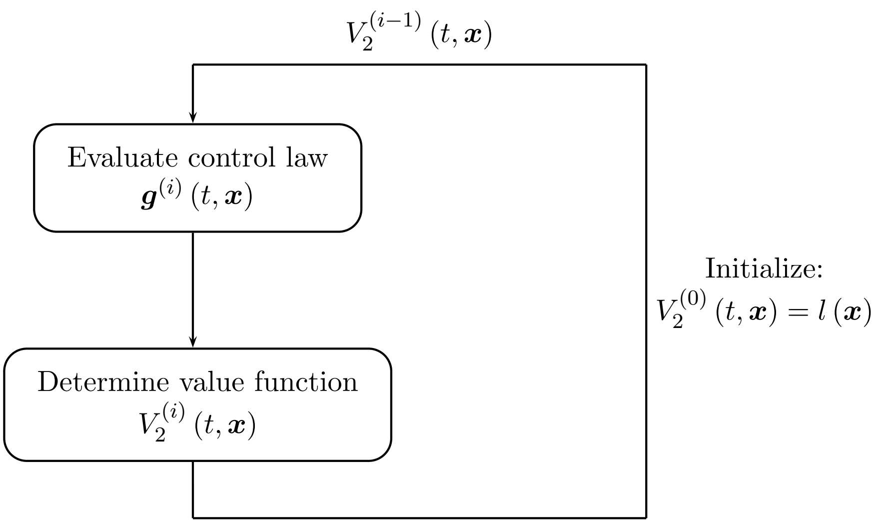


Figure 6-2: The value-iteration procedure as described in definition 6.1.

The question which now arises is whether $V_2^{(i)}$ in definition 6.1 will approach V_2 in the limit case. Starting with $V_2^{(0)}(t,\boldsymbol{x}) = l(\boldsymbol{x})$ at least guarantees that the feedback law: $\boldsymbol{g}^{(1)}(t,x)$

will be optimal at the end-time, i.e. at $t = T$. *But what does this imply for the consecutive iterations?* The next sub-section analyzes the convergence properties of definition 6.1 in greater detail.

6-4-1 Theoretical convergence analysis

Convergence analysis of value-iteration schemes in the continuous-time setting is complicated by the involvement of PDEs. For discrete-time systems, the PDEs are replaced by recursive relations and the convergence analysis can be simplified significantly. Therefore, the proposal is to study the convergence of the discrete-time equivalent of definition 6.1 and then draw connections with the continuous-time version. Although this will not prove anything, at least the convergence of the discrete-time case can be used as a motivation to assert the likelihood of convergence for the continuous-time case.

Consider the *discrete-time* dynamical system, given by the recurrence relation:

$$\boldsymbol{x}(k+1) = \boldsymbol{f_d}\left(\boldsymbol{x}(k), \boldsymbol{y}_{ref}(k)\right) \tag{6-27}$$

where t is replaced by k to denote a discrete-time instance, and the subscript d is added to $\boldsymbol{f}$ to distinguish eq(6-27) from the continuous-time system. Let $\boldsymbol{y}_{ref}(\cdot)$ be the concise notation for a sequence of reference command values[3] and define:

$$\boldsymbol{\mathcal{Y}_{ref}}_{\{k,\cdots,k_0+N\}} := \left\{\boldsymbol{y}_{ref}(\cdot) : \{k,\ldots,k_0+N\} \mapsto \mathrm{Y}_{ref}\right\} \tag{6-28}$$

as the analogue of the command signal space, with N as in eq(6-11). The equivalent of eq(4-9) for the discrete-time system is:

$$V_2(k,\boldsymbol{x}) = \sup_{\boldsymbol{y}_{ref}(\cdot)\in \mathbf{Y}_{\boldsymbol{ref}[k,k_0+N]}} l\left(\boldsymbol{\phi}\left(k_0+N;\boldsymbol{x},k,\boldsymbol{y}_{ref}(\cdot)\right)\right) \tag{6-29}$$

The Bellman equation (i.e. the discrete-time equivalent of the HJB PDE) associated with eq(6-29) is given by:

$$V_2(k,\boldsymbol{x}) = \max_{\boldsymbol{y}_{ref}\in \mathrm{Y}_{ref}} V_2\left(k+1, \boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{y}_{ref}\right)\right) \tag{6-30}$$

Similar to eq(6-1) and eq(6-2), eq(6-30) can be reformulated into two coupled equations. Without loss-of-generality, assuming that $k_0 = 0$, these equations are respectively:

$$V_2(k,\boldsymbol{x}) = V_2\left(k+1, \boldsymbol{f_d}\left(\boldsymbol{x}, \boldsymbol{g}^*(k,\boldsymbol{x})\right)\right) \tag{6-31a}$$

$$V_2(N,\boldsymbol{x}) = l(\boldsymbol{x}) \tag{6-31b}$$

and

$$\boldsymbol{g}^*(k,\boldsymbol{x}) = \arg\max_{\boldsymbol{y}_{ref}\in \mathrm{Y}_{ref}} V_2\left(k+1, \boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{y}_{ref}\right)\right) \tag{6-32}$$

The value-iteration scheme for the discrete-time case can subsequently be defined as follows.

Definition 6.2: ***value-iteration (discrete-time)***
Set $V_2^{(0)}(k,\boldsymbol{x}) = l(\boldsymbol{x})$. *For* $i = 1,2,3,\ldots,$ *do:*

[3] Instead of a signal!

1. Define:

$$\boldsymbol{g}^{(i)}(k,x)=\arg\max_{\boldsymbol{y}_{ref}\in\Upsilon_{ref}} V_2^{(i-1)}\left(k+1,\boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{y}_{ref}\right)\right) \tag{6-33}$$

2. Define $V_2^{(i)}(k,\boldsymbol{x})$ as the solution to:

$$V_2^{(i)}(k,\boldsymbol{x}) = V_2^{(i)}\left(k+1,\boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{g}^{(i)}(k,\boldsymbol{x})\right)\right) \tag{6-34a}$$

$$V_2^{(i)}(N,\boldsymbol{x}) = l(\boldsymbol{x}) \tag{6-34b}$$

Next will be shown that the scheme in definition 6.2 actually converges to V_2, as in eq(6-29), in a finite number of steps.

Theorem 6.1: ***Convergence of value-iteration for a discrete-time system***
Consider definition 6.2. $V_2^{(i)}$ converges to V_2, as in eq(6-29), after at most N iterations, i.e.

$$V_2^{(N)}(k,\boldsymbol{x})=V_2(k,\boldsymbol{x}) \tag{6-35}$$

Proof. First of all, observe that:

$$V_2^{(0)}(k,\boldsymbol{x})=V_2(k,\boldsymbol{x}), \quad \text{for } k=N \tag{6-36}$$

This fact follows straightforwardly from eq(6-31b) and eq(6-34b). Next, it is shown that:

$$V_2^{(1)}(k,\boldsymbol{x})=V_2(k,\boldsymbol{x}), \quad \text{for } k=N,N-1 \tag{6-37}$$

For $k=N$, the result is trivial. However, to see that the equality also holds for $k=N-1$, notice at first that:

$$\begin{aligned}
\boldsymbol{g}^{(1)}(N-1,x) &= \arg\max_{\boldsymbol{y}_{ref}\in\Upsilon_{ref}} V_2^{(0)}\left(N,\boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{y}_{ref}\right)\right)\\
&= \arg\max_{\boldsymbol{y}_{ref}\in\Upsilon_{ref}} V_2\left(N,\boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{y}_{ref}\right)\right)\\
&= \boldsymbol{g}^{*}(N-1,x)
\end{aligned}$$

From which one can verify that:

$$\begin{aligned}
V_2^{(1)}(N-1,\boldsymbol{x}) &= V_2^{(1)}\left(N,\boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{g}^{(1)}(N-1,\boldsymbol{x})\right)\right)\\
&= V_2\left(N,\boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{g}^{(1)}(N-1,\boldsymbol{x})\right)\right)\\
&= V_2\left(N,\boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{g}^{*}(N-1,\boldsymbol{x})\right)\right)\\
&= V_2(N-1,\boldsymbol{x})
\end{aligned}$$

On a similar note, it can be show that:

$$V_2^{(2)}(k,\boldsymbol{x})=V_2(k,\boldsymbol{x}), \quad \text{for } k=N,N-1,N-2 \tag{6-38}$$

Again, the result for $k=N$ is trivial. To show that eq(6-38) holds true for $k=N-1$, one first shows that $\boldsymbol{g}^{(2)}(N-1,x)=\boldsymbol{g}^{(1)}(N-1,x)$ from which one proves that $V_2^{(2)}(N-1,\boldsymbol{x})=$

$V_2^{(1)}(N-1,\boldsymbol{x})$. Then by eq(6-37), it follows that $V_2^{(2)}(N-1,\boldsymbol{x}) = V_2(N-1,\boldsymbol{x})$. To show for $k = N-2$, take first note of the following:

$$\begin{aligned}
\boldsymbol{g}^{(2)}(N-2,x) &= \arg\max_{\boldsymbol{y}_{ref}\in \mathrm{Y}_{ref}} V_2^{(1)}\left(N-1, \boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{y}_{ref}\right)\right) \\
&= \arg\max_{\boldsymbol{y}_{ref}\in \mathrm{Y}_{ref}} V_2\left(N-1, \boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{y}_{ref}\right)\right) \\
&= \boldsymbol{g}^*(N-2,x)
\end{aligned}$$

From which follows that:

$$\begin{aligned}
V_2^{(2)}(N-2,\boldsymbol{x}) &= V_2^{(2)}\left(N-1, \boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{g}^{(2)}(N-2,\boldsymbol{x})\right)\right) \\
&= V_2\left(N-1, \boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{g}^{(2)}(N-2,\boldsymbol{x})\right)\right) \\
&= V_2\left(N-1, \boldsymbol{f_d}\left(\boldsymbol{x},\boldsymbol{g}^{*}(N-2,\boldsymbol{x})\right)\right) \\
&= V_2(N-2,\boldsymbol{x})
\end{aligned}$$

By induction, the following can be stated:

$$V_2^{(l)}(k,\boldsymbol{x}) = V_2(k,\boldsymbol{x}), \quad \text{for } k = N, N-1, \ldots, N-l \tag{6-39}$$

Hence, proving the claim made in eq(6-35). □

A well known fact is that eq(3-1) can be approximated by a discrete-time model using a zero-order hold approximation:

$$\boldsymbol{f_d}\left(\boldsymbol{x}(k),\boldsymbol{y}_{ref}(k)\right) = \boldsymbol{x}(k) + \Delta t \boldsymbol{f}\left(\boldsymbol{x}(k),\boldsymbol{y}_{ref}(k)\right) \tag{6-40}$$

In the limit case of $\Delta t \to 0$ (and $N \to \infty$), eq(6-27) becomes equivalent to eq(3-1). This hints at the possibility that convergence will also hold for the continuous-time case.

6-4-2 Empirical study of the convergence

To further examine the convergence properties of definition 6.1, empirical studies are conducted on the example problem of chapter 5. Because of the nonexistence of analytical solutions, the PDEs are solved numerically with level set algorithms which are known to converge to the correct viscosity solution in the limit case.

The empirical study considers only the nominal system for which the command margins are set to $\theta_{ref,\min} = -8°$ and $\theta_{ref,\max} = 23°$. Furthermore, the boundary condition is set to Eq(5-6). The value functions are solved over the domain $\Omega = [-0.3491, 0.8727] \times [-1.51.5]$ rad, and the grid sizes are set to $\Delta\theta = \frac{2}{180}\pi$ rad and $\Delta q = \frac{2}{180}\pi$ rad. For computing V_2, the spatial derivatives are approximated with a *first-order* scheme, the Hamiltonian is approximated with a *LF scheme*, and the value function is integrated in time with *explicit forward Euler*. For computing $V_2^{(i)}$, the spatial derivatives are approximated with a *first-order* scheme, the convective term is approximated with an *upwind scheme*, and the value function is integrated in time with *explicit forward Euler*.

To study the convergence, the Residual Sum-of-Squares (RSS) is also computed over the entire solution grid, i.e.

$$\mathrm{RSS} = \sum \left(V_2^{(i)}\left(t, \boldsymbol{x}\right) - V_2\left(t, \boldsymbol{x}\right) \right)^2$$

Table 6-1 shows the RSS values for the first 10 iterations. Results indicate that the value-iteration scheme converges to true value function for this particular example problem.

Iteration	RSS	RSS normalized
1	1.15958	1
2	0.02912	0.02511
3	0.00641	0.00553
4	0.00227	0.00195
5	0.00107	0.00092
6	0.00070	0.00060
7	0.00055	0.00048
8	0.00049	0.00042
9	0.00046	0.00040
10	0.00045	0.00039

Table 6-1: RSS as a function of number of iterations.

Concluding remarks on the convergence

To the best of the authors knowledge, no specific results were found in the literature regarding the convergence of the value-iteration scheme for the particular time-dependent, continuous-time case. Consequently, the following assertion has to be made in the research.

Assertion:
Consider the iteration described in definition 6.1. The following claim is made:

$$V_2^{(i)}\left(t, \boldsymbol{x}\right) \to V_2\left(t, \boldsymbol{x}\right), \qquad \text{as } i \to \infty$$

where V_2 is as in eq(4-9).

This assertion can be backed by the following arguments:

- Convergence holds for the discrete-time equivalent of the value-iteration scheme.
- Empirical results indicate that convergence also appears to occur for the continuous-time case.

6-5 The value function approximation algorithm

The value-iteration scheme, as in definition 6.1, is used to convert the non-linear optimization problem of section 6-3 into a sequence of least-squares problems.

Let $\boldsymbol{c}_0^{(i)}, \ldots, \boldsymbol{c}_{N-1}^{(i)}$ denote the coefficients of the value function for the i-th iteration. From eq(6-12) and eq(6-25) it follows that:

$$\boldsymbol{g}^{(i)}\left(\boldsymbol{c}_k^{(i-1)}, \boldsymbol{x}\right) = \arg \max_{\boldsymbol{y}_{ref} \in \mathrm{Y}_{ref}} \left\{ \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{y}_{ref}\right)^T \left(\frac{\partial l\left(\boldsymbol{x}\right)^T}{\partial x} + \frac{\partial \boldsymbol{F}\left(\boldsymbol{x}\right)}{\partial x} \boldsymbol{c}_k^{(i-1)} \right) \right\}$$

Furthermore, eq(6-15) changes into:

$$\alpha_1\left(\boldsymbol{x}, \boldsymbol{c}_{k-1}^{(i-1)}\right)\boldsymbol{c}_{k-1}^{(i)} + \alpha_2\left(\boldsymbol{x}, \boldsymbol{c}_k^{(i-1)}\right)\boldsymbol{c}_k^{(i)} = \beta\left(\boldsymbol{x}, \boldsymbol{c}_{k-1}^{(i-1)}\right), \qquad \boldsymbol{c}_N^{(i)} = 0$$

The value-iteration procedure for the approximate expression of eq(6-17) can effectively described by the recursion:

$$\mathcal{A}\left(x, \boldsymbol{C}^{(i-1)}\right)\boldsymbol{C}^{(i)} = \mathcal{B}\left(x, \boldsymbol{C}^{(i-1)}\right), \qquad \boldsymbol{C}^{(0)} = 0 \tag{6-41}$$

with $\mathcal{A}$ and $\mathcal{B}$ defined as in eq(6-20).

The introduction of value-iteration, transforms the cost function into eq(6-22) a least-squares problem:

$$O\left(\boldsymbol{C}^{(i)}\right) = \left\| \mathrm{X}\left(\boldsymbol{C}^{(i-1)}\right)\boldsymbol{C}^{(i)} - \mathrm{Y}\left(\boldsymbol{C}^{(i-1)}\right) \right\|^2 \tag{6-42}$$

where $\mathrm{X} \in \mathbb{R}^{LN \times pN}$ and $\mathrm{Y} \in \mathbb{R}^{LN \times 1}$ are given by the expressions:

$$\mathrm{X}\left(\boldsymbol{C}^{(i-1)}\right) = \begin{bmatrix} \mathcal{A}\left(\boldsymbol{x}_1, \boldsymbol{C}^{(i-1)}\right) \\ \mathcal{A}\left(\boldsymbol{x}_2, \boldsymbol{C}^{(i-1)}\right) \\ \vdots \\ \mathcal{A}\left(\boldsymbol{x}_L, \boldsymbol{C}^{(i-1)}\right) \end{bmatrix} \tag{6-43}$$

and

$$\mathrm{Y}\left(\boldsymbol{C}^{(i-1)}\right) = \begin{bmatrix} \mathcal{B}\left(\boldsymbol{x}_1, \boldsymbol{C}^{(i-1)}\right) \\ \mathcal{B}\left(\boldsymbol{x}_2, \boldsymbol{C}^{(i-1)}\right) \\ \vdots \\ \mathcal{B}\left(\boldsymbol{x}_L, \boldsymbol{C}^{(i-1)}\right) \end{bmatrix} \tag{6-44}$$

Given the coefficients of a previous value-iteration cycle (i.e. $\boldsymbol{C}^{(i-1)}$), the objective is to determine the coefficients for the next value-iteration $\boldsymbol{C}^{(i)}$ by:

$$\arg \min_{\boldsymbol{C}^{(i)} \in \mathbb{R}^{pN}} O\left(\boldsymbol{C}^{(i)}\right)$$

for which the solution is:

$$\boldsymbol{C}^{(i)} = \left(\mathrm{X}^T\left(\boldsymbol{C}^{(i-1)}\right)\mathrm{X}\left(\boldsymbol{C}^{(i-1)}\right)\right)^{-1}\mathrm{X}^T\left(\boldsymbol{C}^{(i-1)}\right)\mathrm{Y}\left(\boldsymbol{C}^{(i-1)}\right) \tag{6-45}$$

Note that, in order for the inverse to exist, $\mathrm{X}^T\mathrm{X}$ has to be non-singular. This is the case when X has full column rank, imposing the necessary condition that the number of collocation points has to be greater than the number of unknown coefficients (i.e. $L \geq p$).

One of the benefits of value-iteration is that it allows for a framework in which the value function can be adapted to changing system dynamics. Unlike in level set methods, the adaption process avoids a complete re-computation of the value function. The algorithm for approximating the value function is summarized here below.

Algorithm 6.1: ***Adaptive value function approximation scheme***
Initialize $\boldsymbol{C}^{(0)} = 0$. *Given the coefficients of a previous value-iteration cycle (i.e.* $\boldsymbol{C}^{(i)}$*), do the following:*

1. *Construct matrix* X *and vector* Y *based on* $\boldsymbol{C}^{(i-1)}$ *and present dynamics* $\boldsymbol{f}$.

2. *Obtain the new coefficients* $\boldsymbol{C}^{(i)}$ *through:*

$$\boldsymbol{C}^{(i)} = \left(\mathrm{X}^T\left(\boldsymbol{C}^{(i-1)}\right)\mathrm{X}\left(\boldsymbol{C}^{(i-1)}\right)\right)^{-1}\mathrm{X}^T\left(\boldsymbol{C}^{(i-1)}\right)\mathrm{Y}\left(\boldsymbol{C}^{(i-1)}\right)$$

The algorithm is illustrated in figure 6-3.

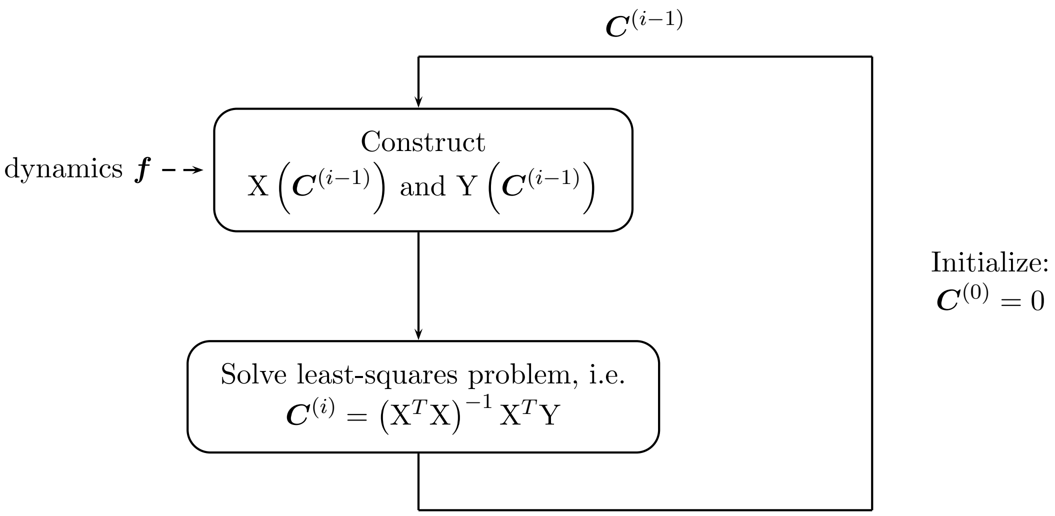


Figure 6-3: The value-iteration procedure as described in definition 6.1.

Chapter 7

Approximation of the value function with simplex splines

7-1 Introduction

The previous chapter introduced an alternative method to approximate the solution of eq(4-12) using regressional techniques. The chapter described the proposed method in a generic fashion by disclosing the type of function approximator used in the scheme. There are several options available for the selection of suitable basis functions, so many, that analyzing the performance of every individual function approximator goes beyond the scope of this thesis. For instance, the proposed scheme may be used along with Radial Basis Functions (RBFs), as it was done in (Huang et al., 2006) and (Alwardi, 2010). Apart from the heuristics in selecting suitable centers, RBFs are linear in the parameters making them a compatible choice for algorithm 6.1. Another function approximator that has the affine property is the multivariate simplex spline. In (Awanou & Lai, 2004), (Awanou, Lai, & Wenston, 2005) and (Hu, Han, & Lai, 2007), simplex splines were already used to solve variants of the Navier-Stokes equations.

Motivated by these papers, this chapter aims to apply multivariate simplex spline theory together with algorithm 6.1 to find polynomial-based approximations of eq(4-12). The "spline method" is tested on the simplified pitch dynamics model. The performance of the method is assessed by comparing the quality of the solutions with level set algorithms.

The chapter is organized as follows. Section 7-2 discusses details concerning the implementation of simplex splines in algorithm 6.1. Section 7-3 presents some preliminary results of the spline method for the simplified pitch dynamics model. Section 7-4 further analyzes the results by comparing the performance of the method to level set algorithms.

7-2 The spline method

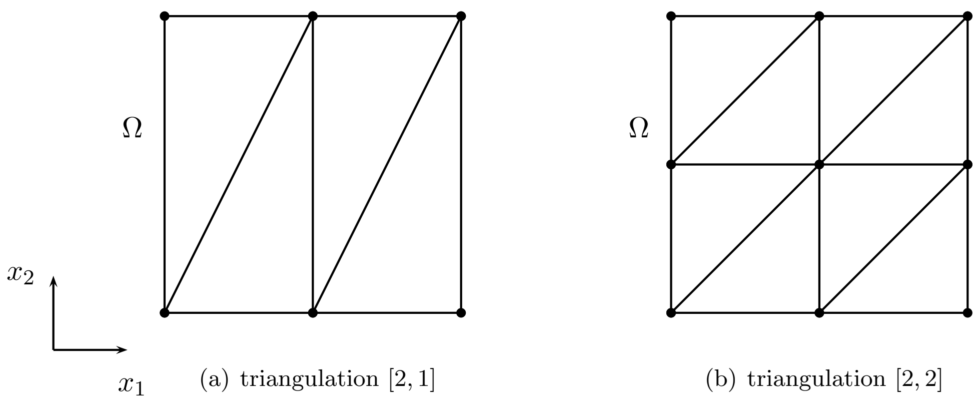
A multivariate simplex spline is a piecewise-polynomial function defined over a special geometric structure called a triangulation. Since simplex spline functions are affine in the parameters, they are directly usable in algorithm 6.1. Appendix B provides a short introduction to the theory of multivariate simplex splines. Readers not familiar with the subject are recommended to read the appendix first before proceeding any further.

The following subsections discuss aspects concerning the implementation of splines in algorithm 6.1. This includes matters such as the selection of suitable spline model parameters and collocation points. Note that the coding of the algorithms is all done in Matlab. A copy of the source code may be obtained under special request from the author.

7-2-1 Spline model selection

The first step in algorithm 6.1 is to select a suitable spline model. This involves the selection of a *triangulation* $\mathcal{T}$ for the computational domain Ω, a *degree* d for the polynomial functions, and a *continuity order* r for the simplex interfaces. The exact effects of the spline model parameters on the quality of the solution is difficult to predict. However, an increase in the number of simplices, or an increase in the polynomial degree, commonly tends to improve the quality of the solutions as the approximation power of the spline is increased. Mind that this increase comes at the price of higher computational costs during estimation and evaluation. Therefore, a trade-off has to be made in the selection of suitable spline model parameters. The optimization of the model parameters can go rather involved, even up to the extend that a separate research project needs to be dedicated to the topic itself.

Consequently, this thesis take only a simplistic approach to triangulate the rectangular domains of the examples. The triangulation is done as follows: the set Ω is partitioned into an arbitrary number of equally sized sub-boxes. Every sub-box is then further divided into two triangles leading to the triangulation depicted in figure 7-1. The triangulation may be parameterized by a two dimensional row vector. The components of the vector denote respectively the number of boxes in the x_1-direction and x_2-direction.



(a) triangulation $[2, 1]$ (b) triangulation $[2, 2]$

Figure 7-1: Triangulation of the computational domain Ω.

The preference is to use spline models with lower degree polynomials. This is done in order to minimize Runge's phenomenon associated with higher-order polynomial approximations. Furthermore, the smoothness of the spline functions is limited to *0th order continuity.* The continuity order affects the level of propagation of characteristic information across simplex boundaries during estimation. Due to time constraints and other priorities during research, experimentation with higher-order continuous splines has not been attempted.

7-2-2 Selection of suitable collocation points

The estimation procedure requires a set of collocation points to be chosen over the computational domain Ω. These points are necessary to construct the regression matrices: eq(6-43) and eq(6-44). The proposal is to use the spatial locations associated with the B-coefficients as collocation points for the algorithm. The spatial distribution of the B-coefficients, also referred to as the B-net in (de Visser et al., 2009), is described by eq(B-10). In this configuration, the collocation points are evenly scattered over the simplex and seem to give reasonable results.

7-2-3 Numerical implementation of the least-squares solution

Let the computational domain Ω be partitioned into J simplexes, the number of collocation points in algorithm 6.1 will then equal $L = J\hat{d}$, where $\hat{d}$ is defined in eq(B-8). In order to enforce continuity in the spline function, $\boldsymbol{C}^{(i)}$, as in eq(6-45), has to meet some linear equality constraints. The following constrained optimization problem needs to be solved in that regard:

$$\min_{\boldsymbol{C}^{(i)} \in \mathbb{R}^{pN}} O\left(\boldsymbol{C}^{(i)}\right) \qquad \text{subject to constraint: } \mathrm{H}_{global}\boldsymbol{C}^{(i)} = 0$$

where $\mathrm{H}_{global} \in \mathbb{R}^{Ns \times J\hat{d}N}$ is the smoothness matrix, given by:

$$\mathrm{H}_{global} = \begin{bmatrix} \mathrm{H}_{t_0} & & \\ & \ddots & \\ & & \mathrm{H}_{t_N} \end{bmatrix} \tag{7-1}$$

where H_{t_k} is as in eq(B-11).

The solution to this constrained optimization problem is obtained by solving the Karun-Kuhn-Tucker (KKT) system:

$$\begin{bmatrix} \mathrm{Q}^{(i-1)} & \mathrm{H}_{global}^{T} \\ \mathrm{H}_{global} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{C}^{(i)} \\ \boldsymbol{\lambda}^{(i)} \end{bmatrix} = \begin{bmatrix} \mathrm{R}^{(i-1)} \\ 0 \end{bmatrix} \tag{7-2}$$

where $\mathrm{Q}^{(i-1)} = \mathrm{X}^T\left(\boldsymbol{C}^{(i-1)}\right)\mathrm{X}\left(\boldsymbol{C}^{(i-1)}\right)$ and $\mathrm{R}^{(i-1)} = \mathrm{X}^T\left(\boldsymbol{C}^{(i-1)}\right)\mathrm{Y}\left(\boldsymbol{C}^{(i-1)}\right)$ refer to the matrices defined in eq(6-43) and eq(6-44).

Eq(7-2) can be solved by a variety of methods. An effective approach is to use the matrix iterative solver presented in (Awanou & Lai, 2004). The iterative solver computes $\boldsymbol{C}^{(i)}$ by the recursion:

$$\boldsymbol{d}^{(k+1)} = \left(\mathrm{Q}^{(i-1)} + \frac{1}{\epsilon}\mathrm{H}_{global}^{T}\mathrm{H}_{global}\right)^{-1}\mathrm{Q}^{(i-1)}\boldsymbol{d}^{(k)} \tag{7-3}$$

where $\epsilon > 0$, and eq(7-3) is initialized by either:

$$\boldsymbol{d}^{(0)} = \left(\mathrm{Q}^{(i-1)} + \frac{1}{\epsilon} \mathrm{H}_{global}^T \mathrm{H}_{global} \right)^{-1} \mathrm{R}^{(i-1)} \tag{7-4}$$

or:

$$\boldsymbol{d}^{(0)} = \boldsymbol{C}^{(i-1)} \tag{7-5}$$

The iteration is known to converge to $\boldsymbol{C}^{(i)}$ as $k \to \infty$. However, from a practical sense, convergence is obtained relatively quickly in three or four steps. The iterative solver is more efficient than solving eq(7-2) directly, because a significantly smaller matrix needs to be inverted in the process. Furthermore, the sparse matrix $\mathrm{Q}^{(i-1)} + \frac{1}{\epsilon} \mathrm{H}_{global}^T \mathrm{H}_{global}$ is known to be positive definite, hence the highly effective Cholesky factorization can be used to solve the system of equations.

Any algorithm that requires solving a system of linear equations is undesirable in a real-time application. Preferably, an effective gradient descent method should be used instead. With the availability of past solutions (i.e. $\boldsymbol{C}^{(i-1)}$), hopes are that convergence can be obtained relatively quickly in a real-time setting. Recommendations are to further study the feasibility of such an approach.

7-3 Results for the simplified pitch dynamics model

The performance of the spline method is tested on the nominal case of the simplified pitch dynamics model. The value function V_1 is approximated for a time-horizon of 3 seconds and the margins are set to respectively: $-8°$ and $23°$ degrees. Since no analytic solutions are available for the problem, the results are compared to high-accuracy level set approximations.

7-3-1 Two different methods to find the safe-set

The safe-set for eq(5-5) can be obtained in several ways. One approach (*method A*) is to compute the value function for the boundary condition in eq(5-6), i.e.

$$l\left(\boldsymbol{x}\right) = \max\left\{-\theta - 10°, \theta - 25°\right\}$$

This approach was taken also in chapter 5 and uses one implicit surface function to describe all constraints at once.

An alternative approach (*method B*) is to evaluate separate value functions for the lower and upper constraint respectively. The intersection of the safe-sets associated to these value functions, describes the safe set for eq(5-5). That is to say, if $V_{1,a}$ and $V_{1,b}$ denote respectively the value functions for the boundary conditions:

$$l_a\left(\boldsymbol{x}\right) = -\theta - 10° \tag{7-6}$$

and

$$l_b\left(\boldsymbol{x}\right) = \theta - 25° \tag{7-7}$$

Then:

$$\left\{ \boldsymbol{x} \in \mathbb{R}^2 \;\middle|\; \max\left\{V_{1,a}\left(0, \boldsymbol{x}\right), V_{1,b}\left(0, \boldsymbol{x}\right)\right\} \leq 0 \right\}$$

is the safe-set for eq(5-5).

The following analysis presents results for both methods. Based upon earlier results (table 6-1), the number of iterations in algorithm 6.1 is fixed to 10, irrespective of the convergence of the scheme. The value functions are approximated for the computational domain Ω : $[-0.2269, 0.5672] \times [-0.9750, 0.9750]$ rad.

7-3-2 Performance metrics for quality assessment

The following performance metrics are introduced to asses the quality of the solutions in greater detail. The first metric ϵ_{norm} denotes the magnitude of the error normalized with respect to a reference solution:

$$\epsilon_{norm}\left(\boldsymbol{x}\right) = \frac{\left\| V_1(0,\boldsymbol{x}) - \hat{V}_1(0,\boldsymbol{x}) \right\|}{\| V_1(0,\boldsymbol{x}) \| + c} \tag{7-8}$$

The normalization is done in order to magnify errors near the zero-level set where accurate results are more critical. The positive constant $c > 0$ is added to prevent division by zero. In this thesis, c was set to 10^{-3}. The second metric η denotes the classification error, and states whether the approximate solution has the same sign as the reference solution for a particular locations in Ω:

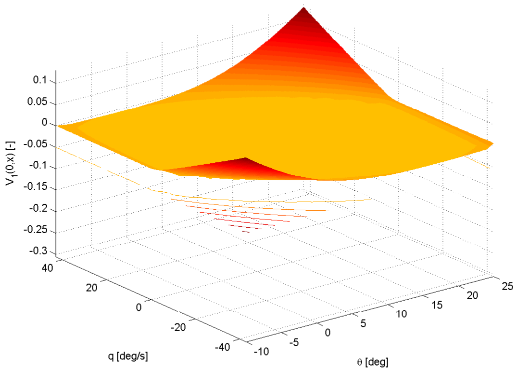
$$\eta\left(\boldsymbol{x}\right) = \begin{cases} 1 & \text{if } \mathrm{sgn} V_1(0,\boldsymbol{x}) = \mathrm{sgn} \hat{V}_1(0,\boldsymbol{x}) \\ 0 & \text{otherwise} \end{cases} \tag{7-9}$$

The reference solutions used in the subsequent results are high-accuracy level set approximations of the solution. The spatial derivatives are approximated with a third-order ENO scheme, a LF approximation is used for the Hamiltonian, and the value function is integrated in time with an explicit third-order RK method. The value function is solved over the domain $\Omega = [-0.3491, 0.8727] \times [-1.5, 1.5]$ rad. The grid accuracy is set to respectively $\Delta\theta = \frac{2}{180}\pi$ rad and $\Delta q = \frac{2}{180}\pi$ rad.

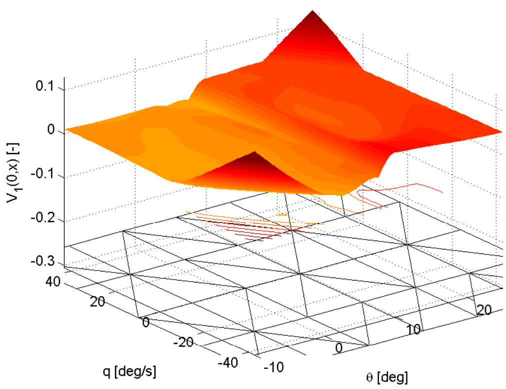
7-3-3 Analysis of the results

Figure 7-2 displays results using method A for two different spline model configurations. The associated performance metrics: ϵ_{norm} and η are shown in figures 7-5 and 7-6. Results illustrate that the approximations get gradually better once the complexity of the spline model is increased. Apart from this basic observation, a jump appears to also occur in the solutions at approximately $\theta = 7.5°$. Further analysis in figures 7-4 and 7-3 clarify that these jumps are caused by the shortcomings of the spline function to approximate the non-smooth solution region caused by the kink at the boundary condition.

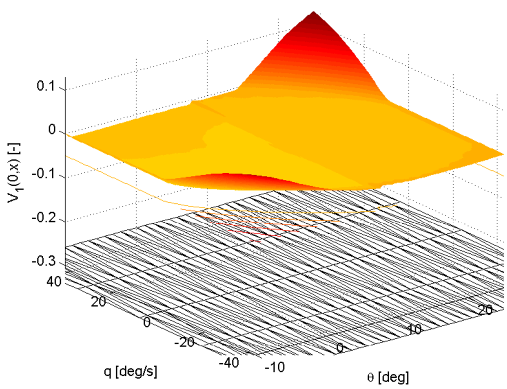
The performance is significant improved when method B is used to obtain the safe-set. In figure 7-7 results are shown for method B using a relatively simple spline model with triangulation: $[2, 2]$, $d = 3$, $r = 0$, $\Delta t = 0.2$ seconds. The spline functions no longer have to approximate the non-smooth solution region caused by the kink in the boundary condition. As a consequence, significantly better approximations are obtained for relatively "less complex" spline model configurations. Remember that figure 7-7 shows the results after 10 iterations. However, judging from figure 7-8, the quality of the solutions do not improve significantly after two iterations.



(a) reference solution

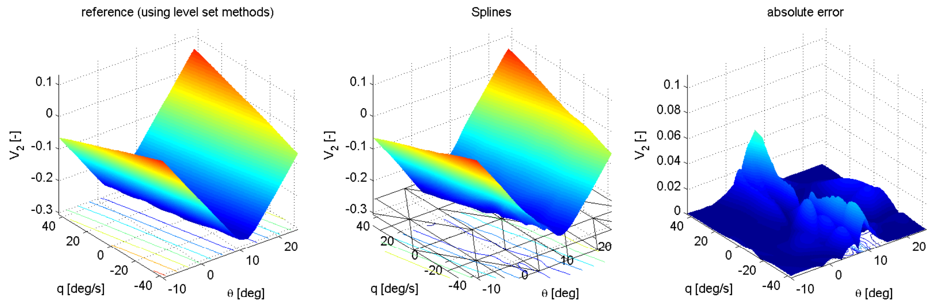


(b) triangulation $[6, 4]$, $d = 3$, $r = 0$, $\Delta t = 0.1$ seconds

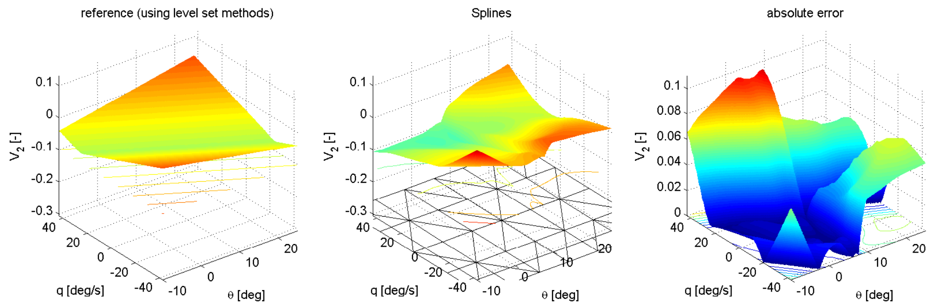


(c) triangulation $[40, 4]$, $d = 3$, $r = 0$, $\Delta t = 0.1$ seconds

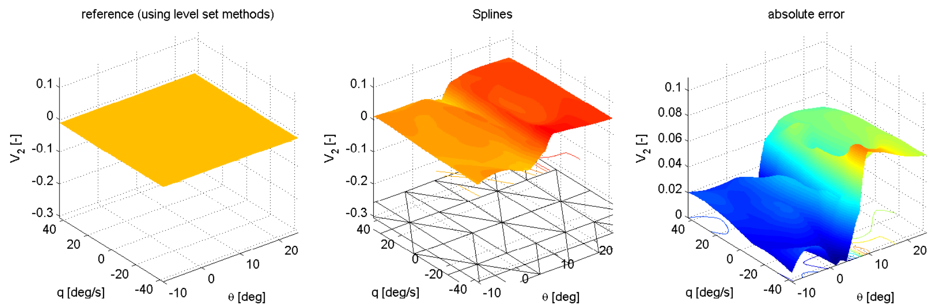
Figure 7-2: Spline approximations of $V_1(0, \boldsymbol{x})$ using method A.



(a) $t = 2.9$ seconds

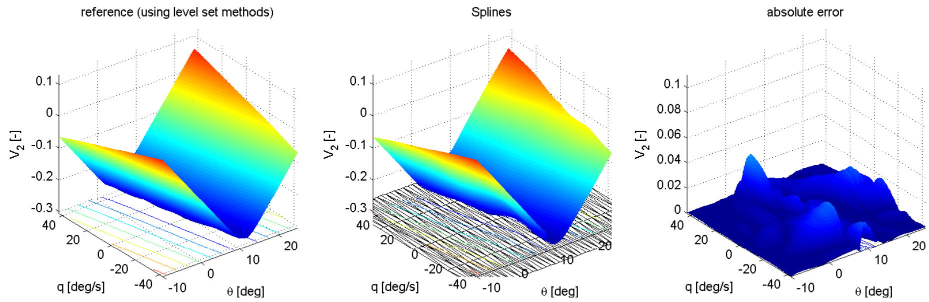


(b) $t = 2.0$ seconds

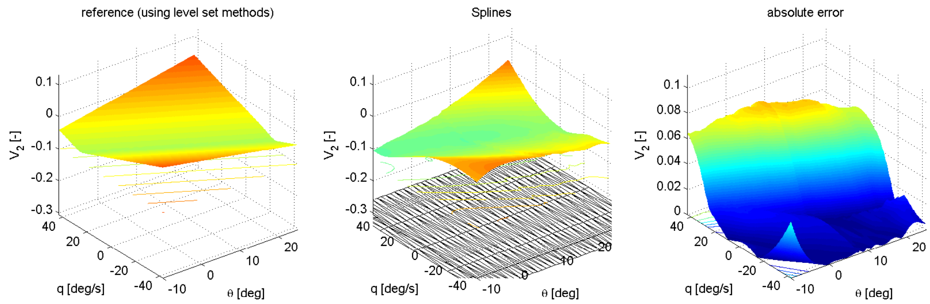


(c) $t = 0.0$ seconds

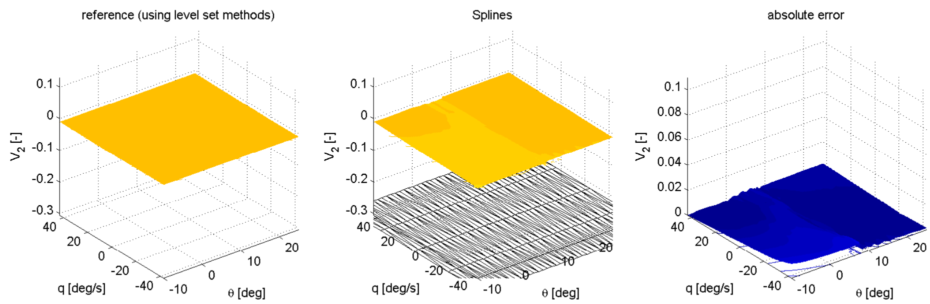
Figure 7-3: V_2 at different time instances for the spline approximation in figure 7-2(b), i.e. with triangulation $[6, 4]$, $d = 3$, $r = 0$, $\Delta t = 0.1$ seconds.



(a) $t = 2.9$ seconds

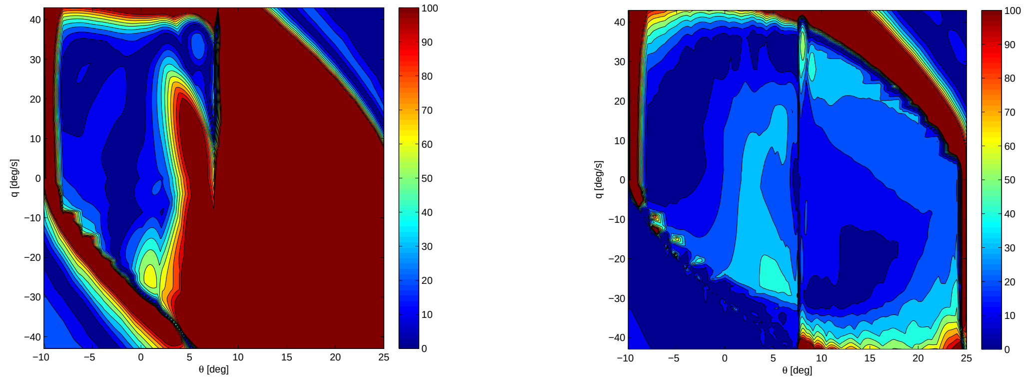


(b) $t = 2.0$ seconds



(c) $t = 0.0$ seconds

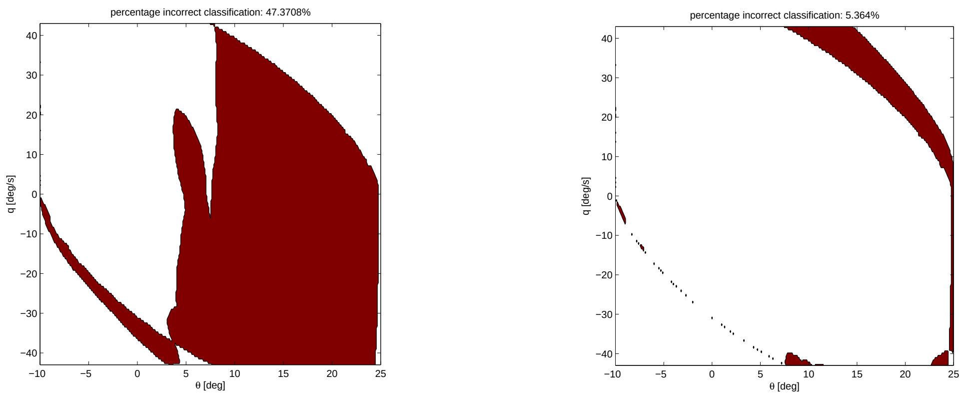
Figure 7-4: V_2 at different time instances for the spline approximation in figure 7-2(c), i.e. with triangulation $[6, 4]$, $d = 3$, $r = 0$, $\Delta t = 0.1$ seconds.



(a) triangulation $[6, 4]$, $d = 3$, $\Delta t = 0.1$ seconds

(b) triangulation $[40, 4]$, $d = 3$, $\Delta t = 0.1$ seconds

Figure 7-5: The normalized error ϵ_{norm} (in %) for the spline approximations in figure 7-2.



(a) triangulation $[6, 4]$, $d = 3$, $r = 0$, $\Delta t = 0.1$ seconds

(b) triangulation $[40, 4]$, $d = 3$, $r = 0$, $\Delta t = 0.1$ seconds

Figure 7-6: The classification error η for the spline approximations in figure 7-2.

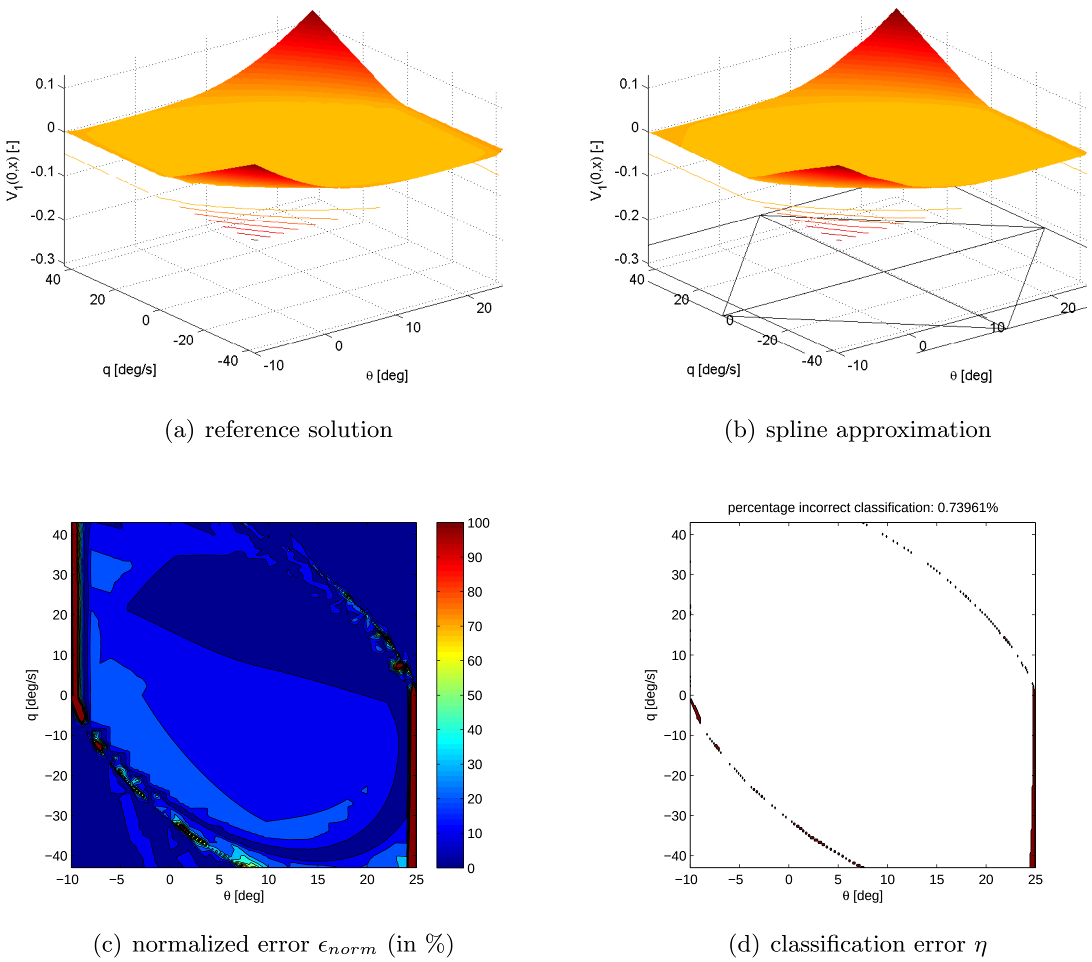
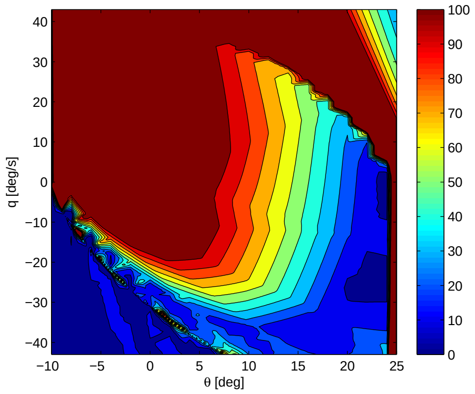
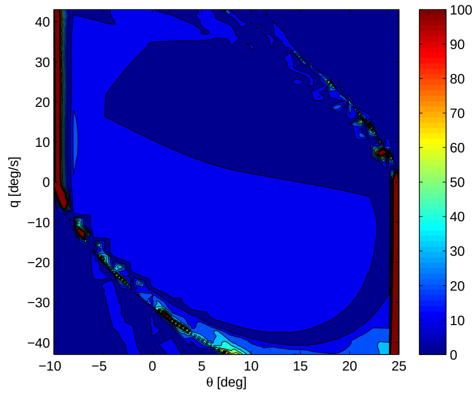


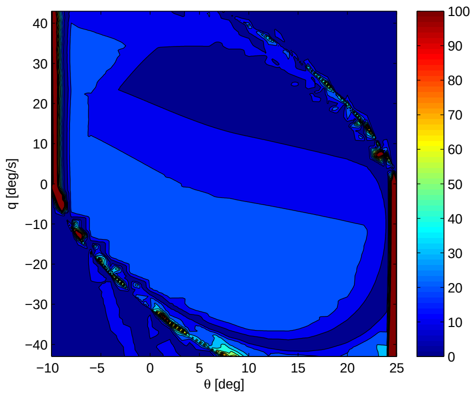
Figure 7-7: Spline approximation of $V_1(0, \boldsymbol{x})$ using method B with triangulation $[2, 2]$, $d = 3$, $r = 0$, $\Delta t = 0.2$ seconds.



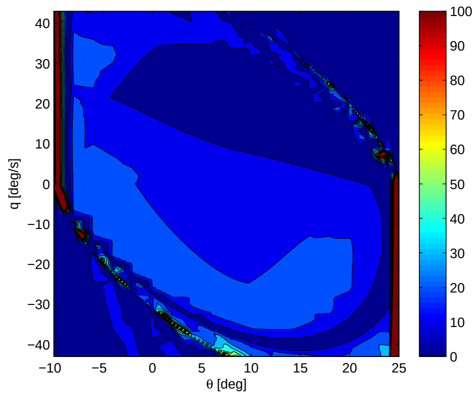
(a) After one iteration.



(b) After two iterations.



(c) After three iterations.



(d) After four iterations.

Figure 7-8: The normalized error ϵ_{norm} (in %) for the spline approximation using method B with triangulation $[2, 2]$, $d = 3$, $r = 0$ $\Delta t = 0.2$ seconds.

7-4 Comparison to level set algorithms

The spline method has some major shortcomings when compared to the level set algorithms. Level set methods are known to converge to the correct viscosity solution as the grid gets more refined. On the other hand, there are no theoretical convergence proofs for the spline method. However, results do indicate that the approximations get gradually better once the complexity of the spline model is increased. Furthermore, the splines inheritently have difficulties approximating non-smooth solution regions of the value function. Judging from figures 7-4 and 7-3, non-smooth boundary conditions cause severe distortions in the spline solution. These distortions mellow down, once the spline model complexity is increased, however at the expense of higher computational costs.

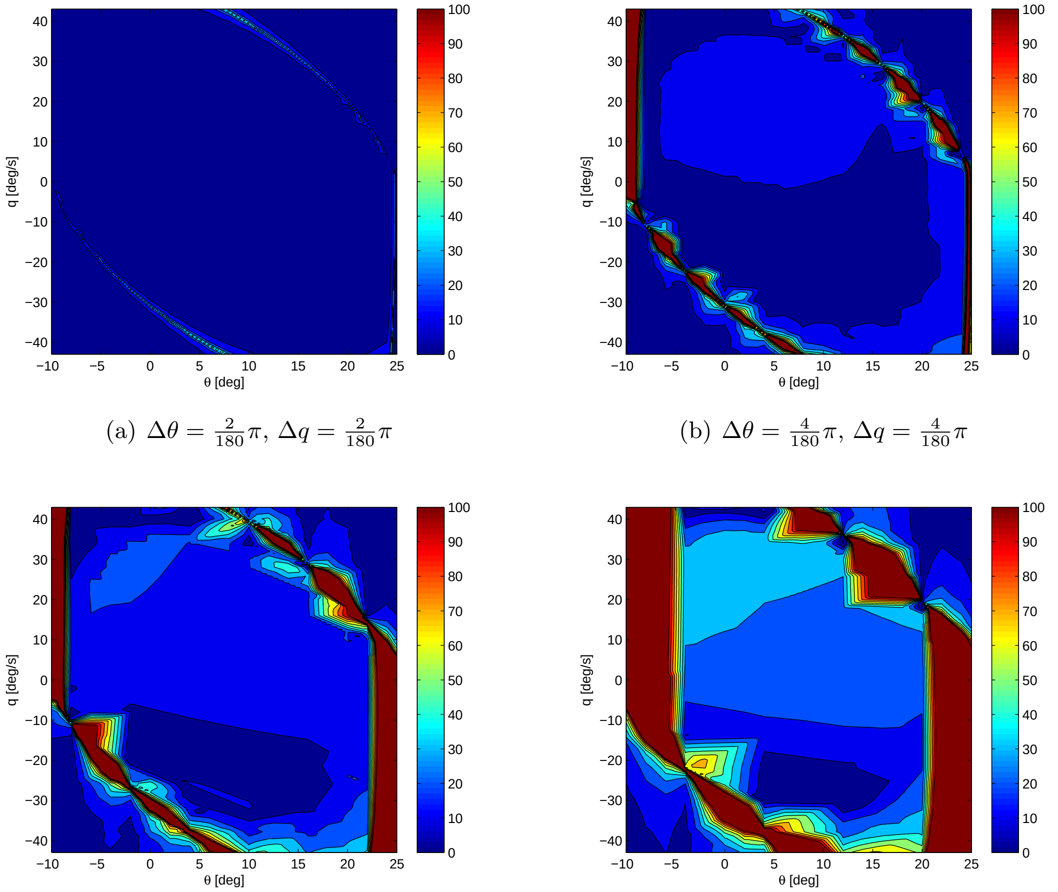
The spline method also exhibit several benefits over the level set algorithms. In the absence of large non-smooth solution regions, the spline method is capable of giving good approximations to the value function at relatively small computation costs. This claim is made based on the observations of Method B to approximate the value function. Table 7-2 shows the computation time[1] needed for five iteration cycles with different spline model settings. Table 7-1 shows the computation times for different grid settings of the level set method. In the tables, the performance parameter: $\epsilon_{norm_{avg}}$ is defined as the mean of eq(7-8) over the entire computational domain, whereas η_{total} denotes the overall classification error.

Note that the above results refer to just one specific example problem. The methods so strongly differ from each other that comparison is very difficult to make. More studies are necessary to further analyze the computational complexity of the methods.

Grid accuracy	$\epsilon_{norm_{avg}}$ *[-]*	η_{total} *[%]*	*CPU time[seconds]*
$\Delta\theta = \frac{2}{180}\pi$, $\Delta q = \frac{2}{180}\pi$	0.0473	2.70	7.75
$\Delta\theta = \frac{4}{180}\pi$, $\Delta q = \frac{4}{180}\pi$	0.2140	3.27	2.73
$\Delta\theta = \frac{6}{180}\pi$, $\Delta q = \frac{6}{180}\pi$	0.4574	7.25	1.65
$\Delta\theta = \frac{8}{180}\pi$, $\Delta q = \frac{8}{180}\pi$	0.9370	18.43	1.23

Table 7-1: Performance of level set algorithm using method B. The spatial derivatives are approximated with a upwind first-order scheme, a LF approximation is used for the Hamiltonian, and the value function is integrated in time using forward Euler.

[1]Note that these times are obtained on a laptop with Intel Core Duo 2.1 GHz processor and 4 GB RAM.



(a) $\Delta\theta = \frac{2}{180}\pi$, $\Delta q = \frac{2}{180}\pi$

(b) $\Delta\theta = \frac{4}{180}\pi$, $\Delta q = \frac{4}{180}\pi$

(c) $\Delta\theta = \frac{6}{180}\pi$, $\Delta q = \frac{6}{180}\pi$

(d) $\Delta\theta = \frac{8}{180}\pi$, $\Delta q = \frac{8}{180}\pi$

Figure 7-9: The normalized error ϵ_{norm} (in %) for the level set algorithm using method B. The spatial derivatives are approximated with a upwind first-order scheme, a LF approximation is used for the Hamiltonian, and the value function is integrated in time using forward Euler.

(a) Performance of the spline approximations using method B with different triangulation configurations. The other settings are: $d = 3$, $r = 0$, $\Delta t = 0.2$ seconds.

triangulation	$\epsilon_{norm_{avg}}$ *[-]*	η_{total} *[%]*	*CPU time [seconds]*
$[1, 1]$	0.3012	2.58	0.078
$[2, 2]$	0.1597	0.74	0.50
$[3, 3]$	0.2743	1.33	1.43
$[4, 4]$	0.2355	1.13	2.46

(b) Performance of the spline approximations using method B with different polynomial degrees. The other settings are: triangulation $[2, 2]$, $r = 0$, $\Delta t = 0.2$ seconds.

degree d	$\epsilon_{norm_{avg}}$ *[-]*	η_{total} *[%]*	*CPU time [seconds]*
2	1.4072	79.99	0.28
3	0.1597	0.74	0.51
4	0.2201	1.48	1.09
5	0.2215	1.40	1.90

(c) Performance of the spline approximations using method B with different time-steps. The other settings are: triangulation $[2, 2]$, $d = 3$, $r = 0$.

time-step Δt [seconds]	$\epsilon_{norm_{avg}}$ *[-]*	η_{total} *[%]*	*CPU time [seconds]*
0.375	0.8043	28.22	0.31
0.3	0.4047	1.58	0.36
0.2	0.1597	0.74	0.51
0.1	0.1509	1.22	1.12

Table 7-2: Performance of the spline approximations using method B with different model configurations. The number of value-iteration cycles is fixed to 5.

Part IV

In-flight command margin estimation

Chapter 8

A step towards real-time command margin estimation

8-1 Introduction

The objective in this thesis project is to determine margins for reference command signals of aircraft control systems, such that predefined state constraints are never violated. The margins are determined through the procedure outlined in figure 4-2, and involves solving a optimal control problem in real-time. So-far, the previous chapters have addressed how the optimization of the cost functional is done.

In this chapter, initial steps are taken to develop algorithms that close-the-loop in figure 4-2. A simple algorithm is presented to determine margins for the pitch command reference of the simplified pitch dynamics model in an iterative procedure. Furthermore, simulations are performed to study the properties of the algorithm with respect to abrupt changes in the dynamics and flight envelope.

The chapter is organized as follows. Section 8-2 presents the algorithm developed to determine the margins of the pitch dynamics model. Section 8-3 presents simulation results with the algorithm for some representative failure scenarios.

8-2 A simple algorithm for the pitch dynamics model

The margins for the simplified pitch dynamics model are found by making incremental changes to the upper and lower limits of the pitch command reference. The margins are adjusted according to the sign of J_a^* and J_b^*, which respectively denote the maxima of the cost functionals:

$$J_a\left(\boldsymbol{x}_0, \theta_{ref}\left(\cdot\right)\right) = \max_{\tau \in [t_0, t_0+T]} l_a\left(\boldsymbol{\phi}\left(\tau; \boldsymbol{x}_0, \theta_{ref}\left(\cdot\right)\right)\right) \tag{8-1}$$

and

$$J_b\left(\boldsymbol{x}_0, \theta_{ref}\left(\cdot\right)\right) = \max_{\tau \in [t_0, t_0+T]} l_b\left(\boldsymbol{\phi}\left(\tau; \boldsymbol{x}_0, \theta_{ref}\left(\cdot\right)\right)\right) \tag{8-2}$$

where l_a and l_b are defined by eq(7-6) and eq(7-7).

Given the dependency of J_a^* and J_b^* on the margins: $(\theta_{ref,\min}, \theta_{ref,\max})$, the following may be stated:

$$J_a^* = J_a^*\left(\boldsymbol{x}_0, \theta_{ref,\min}, \theta_{ref,\max}\right), \quad J_b^* = J_b^*\left(\boldsymbol{x}_0, \theta_{ref,\min}, \theta_{ref,\max}\right)$$

Hence, the objective is to adjust $(\theta_{ref,\min}, \theta_{ref,\max})$ such that:

$$J_a^*\left(\boldsymbol{x}_0\right) \leq 0 \text{ and } J_b^*\left(\boldsymbol{x}_0\right) \leq 0 \tag{8-3}$$

The relative simplicity of the pitch dynamics model allows one to take a heuristic approach to adjust the margins. The heuristic approach involves making incremental one degree changes to $\theta_{ref,\min}$ and $\theta_{ref,\max}$, until the criteria shown in eq(8-3) are all satisfied. The entire heuristic procedure is described here below in algorithmic form.

Algorithm 8.1
Let $\boldsymbol{x}_0$ denote the current state. Initialize $\theta_{ref,\min}$ and $\theta_{ref,\max}$. For every time-step, do the following iteration:

repeat

    Compute $J_a^*\left(\boldsymbol{x}_0\right)$ *and* $J_b^*\left(\boldsymbol{x}_0\right)$.
    if $k > 1$ ***then***

        if $J_a^*\left(\boldsymbol{x}_0\right) \leq 0$ *and* $J_b^*\left(\boldsymbol{x}_0\right) \leq 0$ ***then***
            break;
        else if $\theta_{ref,\min} = \theta_{ref,\max}$ *and (* $J_a^*\left(\boldsymbol{x}_0\right) > 0$ *or* $J_b^*\left(\boldsymbol{x}_0\right) > 0$ *)* ***then***
            break;
        end if

    end if
    if $J_a^*\left(\boldsymbol{x}_0\right) > 0$ ***then***

        if $\theta_{ref,\min} < \theta_{ref,\max}$ ***then***
            $\theta_{ref,\min} \leftarrow \theta_{ref,\min} + 1^\circ$.
        else if $\theta_{ref,\min} = \theta_{ref,\max}$ ***then***
            $\theta_{ref,\min} \leftarrow \theta_{ref,\min} + 1^\circ$.
            $\theta_{ref,\max} \leftarrow \theta_{ref,\max} + 1^\circ$.
        end if
    else
        $\theta_{ref,\min} \leftarrow \theta_{ref,\min} - 1^\circ$.
    end if

    Compute $J_a^*\left(\boldsymbol{x}_0\right)$ *and* $J_b^*\left(\boldsymbol{x}_0\right)$.
    if $k > 1$ ***then***

        if $J_a^*\left(\boldsymbol{x}_0\right) \leq 0$ *and* $J_b^*\left(\boldsymbol{x}_0\right) \leq 0$ ***then***
            break;
        else if $\theta_{ref,\min} = \theta_{ref,\max}$ *and (* $J_a^*\left(\boldsymbol{x}_0\right) > 0$ *or* $J_b^*\left(\boldsymbol{x}_0\right) > 0$ *)* ***then***

break;
end if

end if
if $J_b^*(\boldsymbol{x}_0) > 0$ ***then***

if $\theta_{ref,\max} > \theta_{ref,\min}$ ***then***
$\theta_{ref,\max} \leftarrow \theta_{ref,\max} - 1°$.
else if $\theta_{ref,\max} = \theta_{ref,\min}$ ***then***
$\theta_{ref,\max} \leftarrow \theta_{ref,\max} - 1°$.
$\theta_{ref,\min} \leftarrow \theta_{ref,\min} - 1°$.
end if
else
$\theta_{ref,\max} \leftarrow \theta_{ref,\max} + 1°$.
end if

until *iteration terminates*

For damaged aircraft, the actual envelope has the potential to shrink. It is possible to adapt algorithm 8.1, such that sudden changes in the envelope can be automatically included in the analysis. This adaption is done by modifying eq(7-6) and eq(7-7) by respectively:

$$l_a(\boldsymbol{x}) = -\theta \tag{8-4}$$

and

$$l_b(\boldsymbol{x}) = \theta \tag{8-5}$$

To find the margins, the following conditions need then to be satisfied:

$$J_a^*(\boldsymbol{x}_0) \leq -\theta_{\min} \text{ and } J_b^*(\boldsymbol{x}_0) \leq \theta_{\max}$$

where $\theta_{\min}$ and $\theta_{\max}$ denote respectively the lower and upper limit of the envelope. The modified procedure is described here below in algorithmic form.

Algorithm 8.2
Let $\boldsymbol{x}_0$ denote the current state. Initialize $\theta_{ref,\min}$ and $\theta_{ref,\max}$. For every time-step, do the following iteration:

repeat

Compute $J_a^*(\boldsymbol{x}_0)$ and $J_b^*(\boldsymbol{x}_0)$.
if $k > 1$ ***then***

if $J_a^*(\boldsymbol{x}_0) \leq -\theta_{ref,\min}$ *and* $J_b^*(\boldsymbol{x}_0) \leq \theta_{ref,\max}$ ***then***
break;
else if $\theta_{ref,\min} = \theta_{ref,\max}$ *and (* $J_a^*(\boldsymbol{x}_0) > -\theta_{ref,\min}$ *or* $J_b^*(\boldsymbol{x}_0) > \theta_{ref,\max}$ *)* ***then***
break;
end if

end if
if $J_a^*(\boldsymbol{x}_0) > -\theta_{ref,\min}$ ***then***

  if $\theta_{ref,\min} < \theta_{ref,\max}$ ***then***
    $\theta_{ref,\min} \leftarrow \theta_{ref,\min} + 1°.$
  else if $\theta_{ref,\min} = \theta_{ref,\max}$ ***then***
    $\theta_{ref,\min} \leftarrow \theta_{ref,\min} + 1°.$
    $\theta_{ref,\max} \leftarrow \theta_{ref,\max} + 1°.$
  end if
else
  $\theta_{ref,\min} \leftarrow \theta_{ref,\min} - 1°.$
end if

Compute $J_a^*(\boldsymbol{x}_0)$ *and* $J_b^*(\boldsymbol{x}_0)$.
if $k > 1$ ***then***

  if $J_a^*(\boldsymbol{x}_0) \leq -\theta_{ref,\min}$ *and* $J_b^*(\boldsymbol{x}_0) \leq \theta_{ref,\max}$ ***then***
    break;
  else if $\theta_{ref,\min} = \theta_{ref,\max}$ *and (* $J_a^*(\boldsymbol{x}_0) > -\theta_{ref,\min}$ *or* $J_b^*(\boldsymbol{x}_0) > \theta_{ref,\max}$ *)* ***then***
    break;
  end if

end if
if $J_b^*(\boldsymbol{x}_0) > \theta_{ref,\max}$ ***then***

  if $\theta_{ref,\max} > \theta_{ref,\min}$ ***then***
    $\theta_{ref,\max} \leftarrow \theta_{ref,\max} - 1°.$
  else if $\theta_{ref,\max} = \theta_{ref,\min}$ ***then***
    $\theta_{ref,\max} \leftarrow \theta_{ref,\max} - 1°.$
    $\theta_{ref,\min} \leftarrow \theta_{ref,\min} - 1°.$
  end if
else
  $\theta_{ref,\max} \leftarrow \theta_{ref,\max} + 1°.$
end if

until *iteration terminates*

Algorithm 8.2 is implemented in a Matlab/Simulink environment. The spline method is used to approximate the value function associated with the cost functionals in eq(8-1) and eq(8-2). A copy of the Matlab and Simulink files can be obtained under special request from the author.

8-3 Simulations

Simulations are conducted to study the behavior of the margins in a dynamic environment. Given that the pilot has final authority over the control system, the following two cases can be distinguished:

- Case I: the pilot *follows* the command margins, and provides a reference command signal *within* the margins.
- Case II: the pilot *ignores* the command margins, and provides a reference command signal *outside* of the margins.

The response of the margins is studied for both cases. Simulations are conducted with a duration of 50 seconds for the following "failure" scenarios:

- Scenario A: a *sudden degradation of the system dynamics*, in which the system becomes off-nominal, as in eq(5-4), after $t = 10$ seconds.
- Scenario B: a *sudden degradation of the envelope*, in which the envelope limits change according to:

$$\theta_{\min}(t) = \begin{cases} -10^\circ & t < 20 \text{ seconds} \\ -5^\circ & t \geq 20 \text{ seconds} \end{cases}, \quad \theta_{\max}(t) = \begin{cases} 25^\circ & t < 20 \text{ seconds} \\ 20^\circ & t \geq 20 \text{ seconds} \end{cases} \tag{8-6}$$

- Scenario C: a *superposition of the previous two scenarios* in which the system becomes off-nominal after $t = 10$ seconds and the envelope shrinks according to eq(8-6).

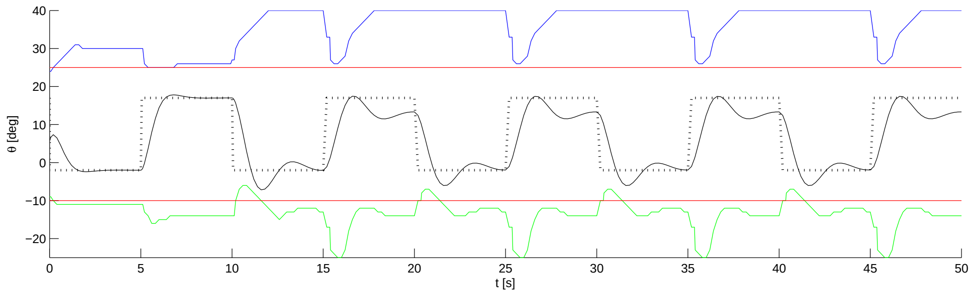
Section 8-3-1 analyzes results for case I, whereas section 8-3-2 analyzes the results for case II.

8-3-1 Case I: Pilot follows the margins

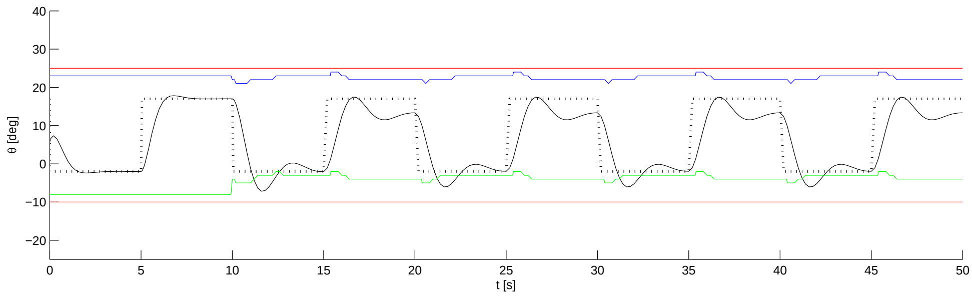
Figure 8-1 shows simulation results for scenario A where the computed margins are displayed along the trajectory of the system. The margins are computed for different time-horizons. Results indicate that the margins tend to become larger when the time-horizon is reduced. In fact, for $T = 1$ second (figure 8-1(a)), the margins even exceed the flight envelope. Apparently, the inputs of the system do not have enough control power to steer the aircraft out of the envelope within an one second horizon. The time-scale of the system requires one to compute the margins for a larger time-window in order to account for all transient effects. For the type of dynamics in the example problem, a time-horizon of $T = 5$ seconds is a more suitable indication of safety. For $T = 5$ seconds, the margins clearly shrink once the dynamics become off-nominal, which gives the correct indication that the aircraft performance has been degraded.

In general, small time-horizons can give misleading (i.e. large) margins to a pilot, especially if the time window is not able to capture all the transients. On the other hand, a large time-horizon T is of no practical use as one is uncertain about the dynamics in the distant future. Aircraft dynamics are highly non-linear, and one typically has only local approximations of the dynamics available, especially during off-nominal conditions. Consequently, a trade-off needs to be made in the selection of a suitable time-horizon. A suggestion is to fix T to two or three times the time-constant of the nominal system.

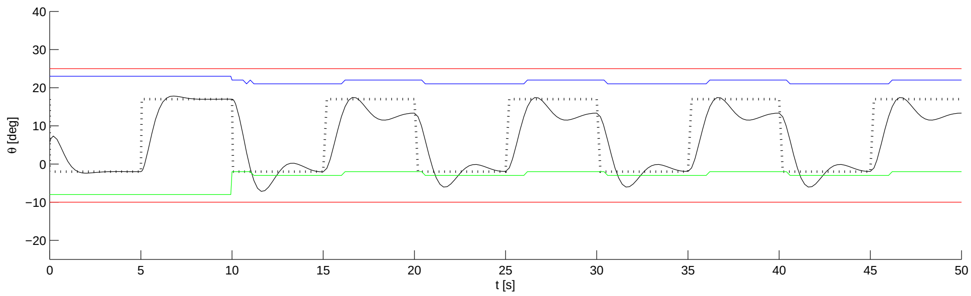
In figure 8-2 and 8-3, simulation results are shown for respectively scenario B and C. The margins displayed in the figures are for a time-horizon of $T = 5$ seconds. As in figure 8-1, the pilot again provides a reference command signal in the form of a block function. As expected, results show that the margins shrink once the flight envelope also shrinks. The margins shrink even further when the system also becomes off-nominal. The state trajectories appear to also never violate the flight envelope. This observation is consistent with the theory because the pilot continuously provides a reference command signal within the margins.



(a) Results for $T = 1$ seconds.



(b) Results for $T = 3$ seconds.



(c) Results for $T = 5$ seconds.

Figure 8-1: The command margins estimated for scenario A. The upper and lower limits of the margins are shown respectively in blue and green. The envelope limits are denoted in red. The state trajectory is denoted by the black continuous line. The reference command signal is denoted by the black dotted line.

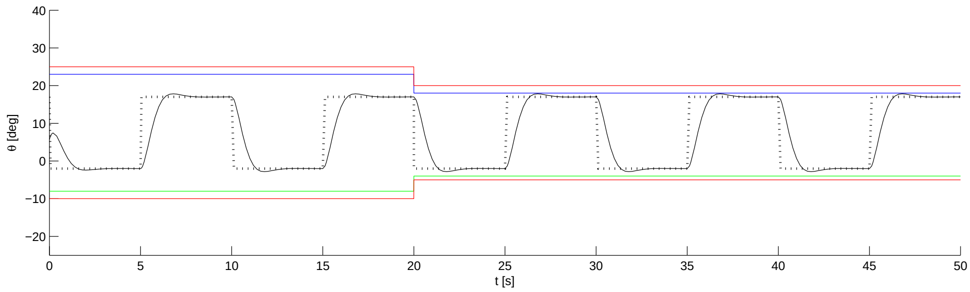


Figure 8-2: The command margins estimated for scenario B with $T = 5$ seconds. The upper and lower limits of the margins are shown respectively in blue and green. The envelope limits are denoted in red. The state trajectory is denoted by the black continuous line. The reference command signal is denoted by the black dotted line.

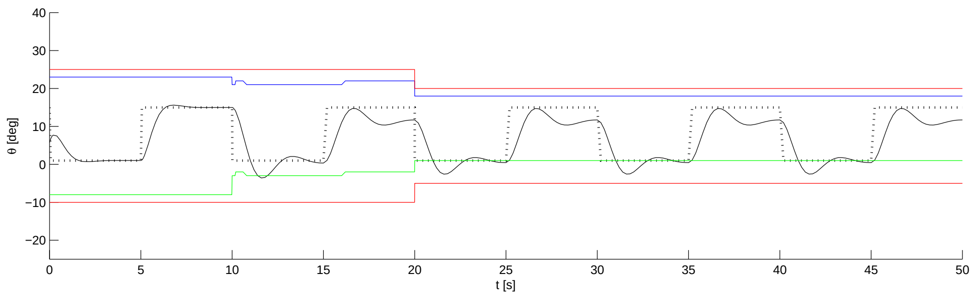
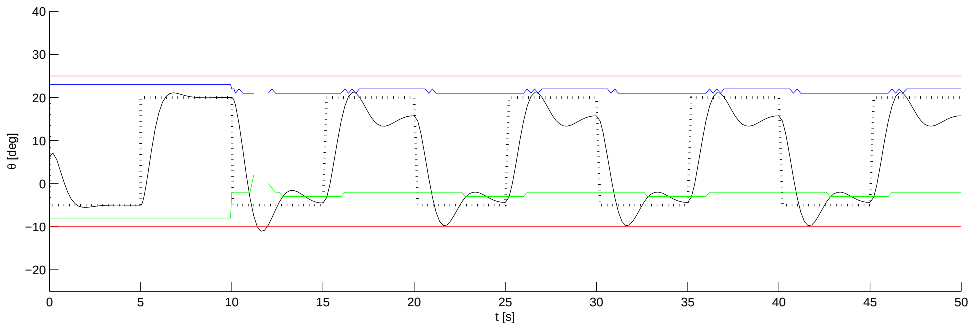


Figure 8-3: The command margins estimated for scenario C with $T = 5$ seconds. The upper and lower limits of the margins are shown respectively in blue and green. The envelope limits are denoted in red. The state trajectory is denoted by the black continuous line. The reference command signal is denoted by the black dotted line.

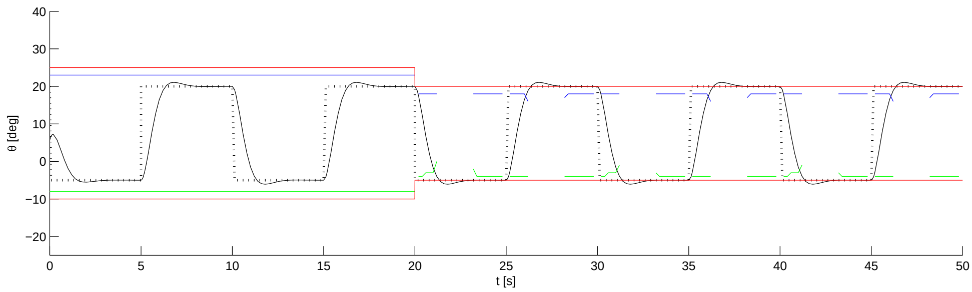
8-3-2 Case II: Pilot ignores the margins

Figure 8-4 shows more simulation results for scenario A, B, and C. The margins are again estimated for a time-horizon of $T = 5$ seconds. However, this time the pilot provides a reference command signal that occasionally *violates* the margins. In compliance with expectations, results indicate that envelope excursions may occur under prolonged neglect of the margins. However, mind that such a neglect does not imply an inevitable violation of the envelope. For example, in figure 8-4(a) the pilot repeatedly violates the margins after $t = 20$ seconds, but the system never violates the envelope.

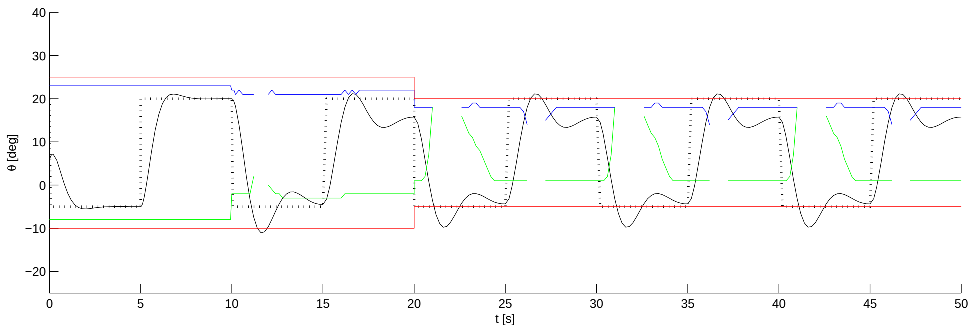
An envelope excursion is commonly preceded by a rapid shrinkage of the margins, which indicates that the aircraft is fast approaching the edge of the envelope. This shrinkage can especially be noticed in figure 8-4(c) after $t = 20$ seconds. An envelope excursion can be prevented if the pilot reacts in time to the changes in the margins. This is illustrated in figure 8-5 where some more simulations results are presented for scenario A. Results are zoomed-in to the time interval: $[9, 11.5]$ seconds when the transition occurs from nominal to off-nominal dynamics. In figure 8-5(b) and 8-5(c), the pilot avoids an envelope excursion by changing his reference command at approximately $t = 11.2$ seconds. Note however that in reality an envelope violation could not have been avoided, given the reaction time of a pilot. The proposed framework for envelope protection will be for suitable for outer-loop control variables of aircraft control systems (e.g. airspeed, flight-path angle). The dynamics of these variables evolve at a slower pace, giving enough time for the pilot to take countermeasures.



(a) Results for scenario A.

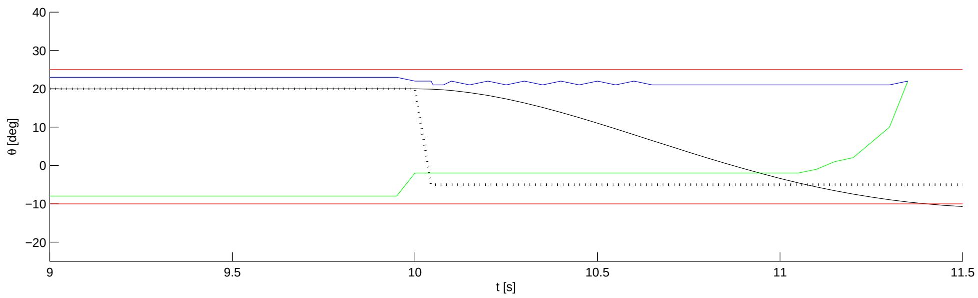


(b) Results for scenario B.

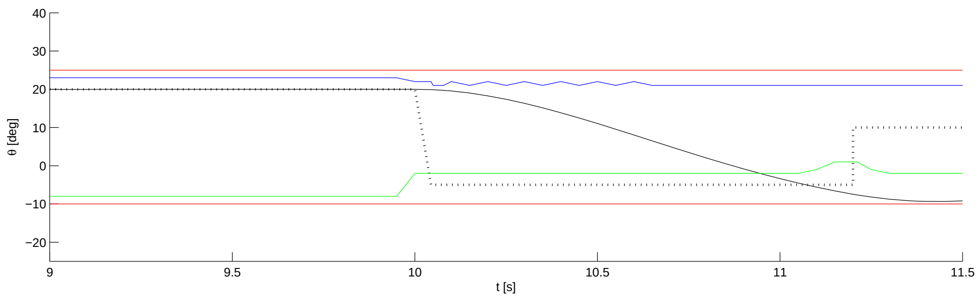


(c) Results for scenario C.

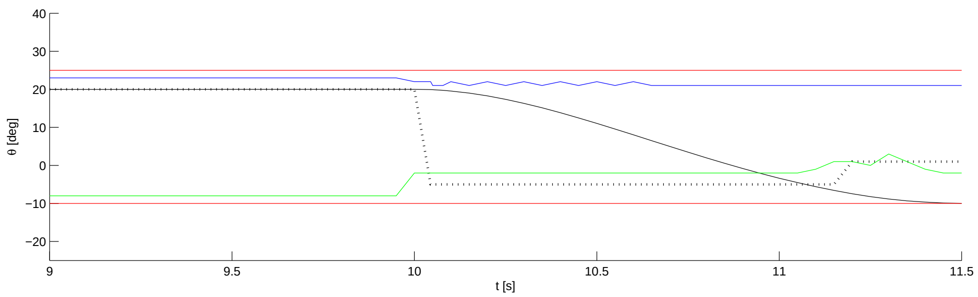
Figure 8-4: Command margins estimated for different scenarios with $T = 5$ seconds. This time the reference command signals of the pilot do not always satisfy the margins. The upper and lower limits of the margins are shown respectively in blue and green. The envelope limits are denoted in red. The state trajectory is denoted by the black continuous line. The reference command signal is denoted by the black dotted line.



(a) The pilot ignores the margins continuously. The margins shrink rapidly followed by a envelope violation.



(b) The pilot responds to the changes in the margins. Consequently, the margins recover to their original state and a envelope violated is prevented.



(c) The pilot responds to the changes in the margins, but this time, a reference command is given which barely satisfies the margins. Consequently, the aircraft briefly touches the envelope boundaries, but an violation does not occur.

Figure 8-5: Command margins estimated for scenario A with $T = 5$ seconds. The results are shown for the time interval: $[9, 11.5]$ seconds where the transition occurs from nominal to off-nominal dynamics. The upper and lower limits of the margins are shown respectively in blue and green. The envelope limits are denoted in red. The state trajectory is denoted by the black continuous line. The reference command signal is denoted by the black dotted line.

Chapter 9

Command margin estimation with uncertain dynamics

9-1 Introduction

In reality, the aircraft dynamics are known only for the nominal case. During off-nominal conditions, the command margins have to be estimated with approximations of the dynamics instead. The detection and identification of anomalies in-flight has to be done extremely fast in order to prevent LOC incidents. LOC may happen within a matter of seconds and without any prior notice. The pilot needs to be warned well in advance on how to adapt to the degrading condition.

This chapter discusses how safe command margins can be estimated for aircraft when the dynamics are not known entirely. The goal is to estimate the correct margins without exciting the inputs of the system. The proposal is to use the prediction model in (Stepanyan et al., 2011) to obtain real-time approximations of the system dynamics. The prediction model is chosen over traditional parameter estimation techniques for its properties regarding parameter convergence. Simulations with the simplified pitch dynamics model are conducted in order to study the effects of this approach.

The chapter is organized as follows. Section 9-2 introduces the prediction model for a Linear Time-Invariant (LTI) system. Furthermore, the section discusses how the prediction model is integrated with the command margin estimation procedure. Section 8-3 presents results obtained with the prediction model.

9-2 The prediction model

The prediction model is a state predictor that runs parallel to the flight control system (see figure 9-1). The prediction model follows the input-output behavior of the plant by adaption of certain parameters. The adaptive laws are a function of the predicted state, actual state and input of the system. These laws are designed such that the error between the predicted state and actual state converges to zero.

What follows hereafter is a description of the prediction model for a LTI system. Extensions to more general classes of dynamical systems are discussed in (Stepanyan et al., 2011) or (Stepanyan & Krishnakumar, 2011), and go beyond the scope of this thesis.

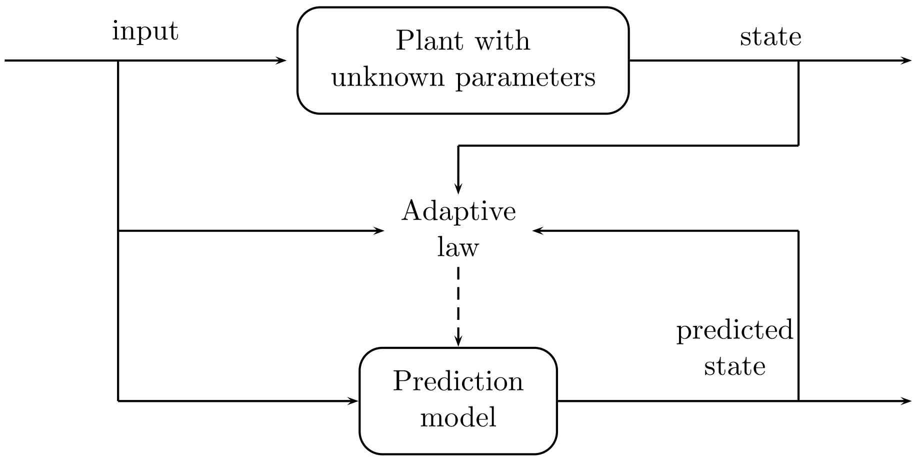


Figure 9-1: Prediction model

9-2-1 The Linear Time-Invariant case

Assume the nominal dynamics of the aircraft to be known entirely. Denote the nominal system by:

$$\dot{\boldsymbol{x}}(t) = \mathrm{A}_0\boldsymbol{x}(t) + \mathrm{B}_0\boldsymbol{u}(t) \tag{9-1}$$

where A_0 is a Hurwitz matrix (i.e. all eigenvalues of A_0 have a strictly negative real part). Since A_0 is Hurwitz, there exists for any given positive definite matrix $\mathrm{Q} = \mathrm{Q}^T > 0$, a positive definite matrix $\mathrm{P} = \mathrm{P}^T > 0$ such that:

$$\mathrm{AP} + \mathrm{AP}^T = -\mathrm{Q} \tag{9-2}$$

where eq(9-2) is known as the Lyapunov equation.

Now consider the off-nominal system:

$$\dot{\boldsymbol{x}}(t) = \mathrm{A}\boldsymbol{x}(t) + \mathrm{B}\boldsymbol{u}(t) \tag{9-3}$$

where:

$$\mathrm{A} = \mathrm{A}_0 + \Delta\mathrm{A}, \quad \mathrm{B} = \mathrm{B}_0 + \Delta\mathrm{B}$$

The off-nominal system may also be expressed by:

$$\dot{\boldsymbol{x}}(t) = \mathrm{A}_0\boldsymbol{x}(t) + \mathrm{B}_0\boldsymbol{u}(t) + \Delta\mathrm{A}\boldsymbol{x}(t) + \Delta\mathrm{B}\boldsymbol{u}(t) \tag{9-4}$$

where $\Delta\mathrm{A}$ and $\Delta\mathrm{B}$ are unknown terms.

Remark: *Notice that when* $\Delta \mathrm{A}, \Delta \mathrm{B} = 0$*, the system is behaving nominally.*

The prediction model for eq(9-4) is described in the following definition.

Definition 9.1: ***Prediction model for a LTI system***
Let $\hat{\boldsymbol{x}}$ *denote the predicted state. The prediction model is defined as:*

$$\dot{\hat{\boldsymbol{x}}}(t) = \mathrm{A}_0 \hat{\boldsymbol{x}}(t) + \mathrm{B}_0 \boldsymbol{u}(t) + \Delta\hat{\mathrm{A}}(t)\,\boldsymbol{x}(t) + \Delta\hat{\mathrm{B}}(t)\,\boldsymbol{u}(t) + \lambda\left(\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\right) \tag{9-5}$$

where λ *is a design parameter and* $\Delta\hat{\mathrm{A}}(t)$, $\Delta\hat{\mathrm{B}}(t)$ *are parameters updated by the laws:*

$$\Delta\dot{\hat{\mathrm{A}}}(t) = \gamma \mathrm{P}\left(\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\right)\boldsymbol{x}^T(t) \tag{9-6a}$$

$$\Delta\dot{\hat{\mathrm{B}}}(t) = \gamma \mathrm{P}\left(\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\right)\boldsymbol{u}^T(t) \tag{9-6b}$$

with γ *defined as the adaption rate, and* P *a solution of eq(9-2) for some given* $\mathrm{Q} = \mathrm{Q}^T > 0$.

The predicted state in definition 9.1 closely follows the actual state of the system under specific conditions for λ and γ. Define the prediction error as:

$$\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t) \tag{9-7}$$

The prediction error dynamics are given by:

$$\dot{\tilde{\boldsymbol{x}}}(t) = \left(\mathrm{A}_0 - \lambda I\right)\tilde{\boldsymbol{x}}(t) + \Delta\tilde{\mathrm{A}}(t)\,\boldsymbol{x}(t) + \Delta\tilde{\mathrm{B}}(t)\,\boldsymbol{u}(t) \tag{9-8}$$

where $\Delta\tilde{\mathrm{A}}(t) = \Delta\mathrm{A} - \Delta\hat{\mathrm{A}}(t)$ and $\Delta\tilde{\mathrm{B}}(t) = \Delta\mathrm{B} - \Delta\hat{\mathrm{B}}(t)$. Since $\Delta\mathrm{A}$ and $\Delta\mathrm{B}$ are constants, it follows that:

$$\Delta\dot{\tilde{\mathrm{A}}}(t) = -\gamma \mathrm{P}\tilde{\boldsymbol{x}}(t)\,\boldsymbol{x}^T(t) \tag{9-9a}$$

$$\Delta\dot{\tilde{\mathrm{B}}}(t) = -\gamma \mathrm{P}\tilde{\boldsymbol{x}}(t)\,\boldsymbol{u}^T(t) \tag{9-9b}$$

The prediction-error dynamics are known to be asymptotically stable for $\gamma > 0, \lambda > 0$. This is proven in the next theorem using Lyapunov stability theory.

Theorem 9.1: ***Asymptotic properties of the prediction error***
Consider the LTI system eq(9-4) along with the prediction model eq(9-5) and update laws eq(9-6).

Let $\gamma > 0, \lambda > 0$*, then the following asymptotic relations hold:*

$$\tilde{\boldsymbol{x}}(t) \rightarrow 0 \tag{9-10a}$$

$$\Delta\dot{\hat{\mathrm{A}}}(t) \rightarrow 0 \tag{9-10b}$$

$$\Delta\dot{\hat{\mathrm{B}}}(t) \rightarrow 0 \tag{9-10c}$$

as $t \rightarrow \infty$.

Proof. Take a $\gamma > 0$ and consider the following candidate Lyapunov function:

$$V(t) = \tilde{\boldsymbol{x}}^T(t)\,\mathrm{P}\tilde{\boldsymbol{x}}(t) + \frac{1}{\gamma}\,\mathrm{trace}\left(\Delta\tilde{\mathrm{A}}^T(t)\,\Delta\tilde{\mathrm{A}}(t) + \Delta\tilde{\mathrm{B}}^T(t)\,\Delta\tilde{\mathrm{B}}(t)\right) \tag{9-11}$$

The derivative of eq(9-11) with respect to time is:

$$\dot{V}(t) = 2\tilde{\boldsymbol{x}}^T(t)\,\mathrm{P}\dot{\tilde{\boldsymbol{x}}}(t) + \frac{2}{\gamma}\ \text{trace}\left(\Delta\tilde{\mathrm{A}}^T(t)\,\Delta\dot{\tilde{\mathrm{A}}}(t) + \Delta\tilde{\mathrm{B}}^T(t)\,\Delta\dot{\tilde{\mathrm{B}}}(t)\right) \tag{9-12}$$

Direct substitution of eq(9-8) , eq(9-9a) and eq(9-9b) yields:

$$\begin{aligned}\dot{V}(t) &= 2\tilde{\boldsymbol{x}}^T(t)\,\mathrm{P}\left((\mathrm{A}_0 - \lambda I)\,\tilde{\boldsymbol{x}}(t) + \Delta\tilde{\mathrm{A}}(t)\,\boldsymbol{x}(t) + \Delta\tilde{\mathrm{B}}(t)\,\boldsymbol{u}(t)\right) + \\ &\quad \frac{2}{\gamma}\ \text{trace}\left(\Delta\tilde{\mathrm{A}}^T(t)\left(-\gamma\mathrm{P}\tilde{\boldsymbol{x}}(t)\,\boldsymbol{x}^T(t)\right) + \Delta\tilde{\mathrm{B}}^T(t)\left(-\gamma\mathrm{P}\tilde{\boldsymbol{x}}(t)\,\boldsymbol{u}^T(t)\right)\right) \\ &= 2\tilde{\boldsymbol{x}}^T(t)\,\mathrm{P}(\mathrm{A}_0 - \lambda I)\,\tilde{\boldsymbol{x}}(t) + 2\tilde{\boldsymbol{x}}^T(t)\,\mathrm{P}\Delta\tilde{\mathrm{A}}(t)\,\boldsymbol{x}(t) + 2\tilde{\boldsymbol{x}}^T(t)\,\mathrm{P}\Delta\tilde{\mathrm{B}}(t)\,\boldsymbol{u}(t) - \\ &\quad 2\ \text{trace}\left(\Delta\tilde{\mathrm{A}}^T(t)\,\mathrm{P}\tilde{\boldsymbol{x}}(t)\,\boldsymbol{x}^T(t)\right) - 2\ \text{trace}\left(\Delta\tilde{\mathrm{B}}^T(t)\,\mathrm{P}\tilde{\boldsymbol{x}}(t)\,\boldsymbol{u}^T(t)\right)\end{aligned}$$

By using the property: $\boldsymbol{q}^T\boldsymbol{r} = \text{trace}(\boldsymbol{r}\boldsymbol{q}^T) = \text{trace}(\boldsymbol{q}\boldsymbol{r}^T)$, the following can be shown:

$$\tilde{\boldsymbol{x}}^T(t)\,\mathrm{P}\Delta\tilde{\mathrm{B}}(t)\,\boldsymbol{u}(t) = \text{trace}\left(\Delta\tilde{\mathrm{B}}^T(t)\,\mathrm{P}\tilde{\boldsymbol{x}}(t)\,\boldsymbol{u}^T(t)\right)$$

and

$$\tilde{\boldsymbol{x}}^T(t)\,\mathrm{P}\Delta\tilde{\mathrm{A}}(t)\,\boldsymbol{x}(t) = \text{trace}\left(\Delta\tilde{\mathrm{A}}^T(t)\,\mathrm{P}\tilde{\boldsymbol{x}}(t)\,\boldsymbol{x}^T(t)\right)$$

Hence,

$$\dot{V}(t) = 2\tilde{\boldsymbol{x}}^T(t)\,\mathrm{P}(\mathrm{A}_0 - \lambda I)\,\tilde{\boldsymbol{x}}(t) \tag{9-13}$$

By re-arrangement of the terms, eq(9-13) can be expressed as:

$$\dot{V}(t) = \tilde{\boldsymbol{x}}^T(t)\left(\mathrm{P}\mathrm{A}_0 + \mathrm{A}_0^T\mathrm{P}\right)\tilde{\boldsymbol{x}}(t) - 2\lambda\tilde{\boldsymbol{x}}^T(t)\,\mathrm{P}\tilde{\boldsymbol{x}}(t) \tag{9-14}$$

Furthermore, application of eq(9-2) yields:

$$\dot{V}(t) = -\tilde{\boldsymbol{x}}^T(t)\,\mathrm{Q}\tilde{\boldsymbol{x}}(t) - 2\lambda\tilde{\boldsymbol{x}}^T(t)\,\mathrm{P}\tilde{\boldsymbol{x}}(t) \tag{9-15}$$

Since $\mathrm{P} > 0$, $\mathrm{Q} > 0$, Lyapunov stability theory guarantees eq(9-10a). Eq(9-10b) and eq(9-10c) follow directly from eq(9-9), hence proving the claims made in the theorem. □

9-2-2 Estimating command margins with the prediction model

The parameters $\Delta\hat{\mathrm{A}}(t)$, $\Delta\hat{\mathrm{B}}(t)$ are used to approximate the system dynamics. Essentially, the input-output behavior is approximated with the Linear Time-Variant (LTV) system:

$$\dot{\boldsymbol{x}}(t) = \hat{\mathrm{A}}(t)\,\boldsymbol{x}(t) + \hat{\mathrm{B}}(t)\,\boldsymbol{u}(t) \tag{9-16}$$

where:

$$\hat{\mathrm{A}}(t) = \mathrm{A}_0 + \Delta\hat{\mathrm{A}}(t)\,, \quad \hat{\mathrm{B}}(t) = \mathrm{B}_0 + \Delta\hat{\mathrm{B}}(t)$$

The integration of the prediction model with the command margin estimation procedure is illustrated in figure 9-2. The estimates: $\hat{\mathrm{A}}(t)$ and $\hat{\mathrm{B}}(t)$, are used in the optimal control formulation. Instead of eq(9-3), the margins are computed for the LTI system:

$$\dot{\boldsymbol{x}}(t) = \hat{\mathrm{A}}(t_0)\,\boldsymbol{x}(t) + \hat{\mathrm{B}}(t_0)\,\boldsymbol{u}(t) \tag{9-17}$$

where t_0 presents the current time. In order for the margins to be estimated correctly, eq(9-17) has to converge to eq(9-3) as time progresses. This convergence is however an issue, since the statements of eq(9-10b) and eq(9-10c) in theorem 9.1 do not immediately imply that: $\tilde{\mathrm{A}}(t) \to 0$ and $\tilde{\mathrm{B}}(t) \to 0$, as $t \to \infty$.

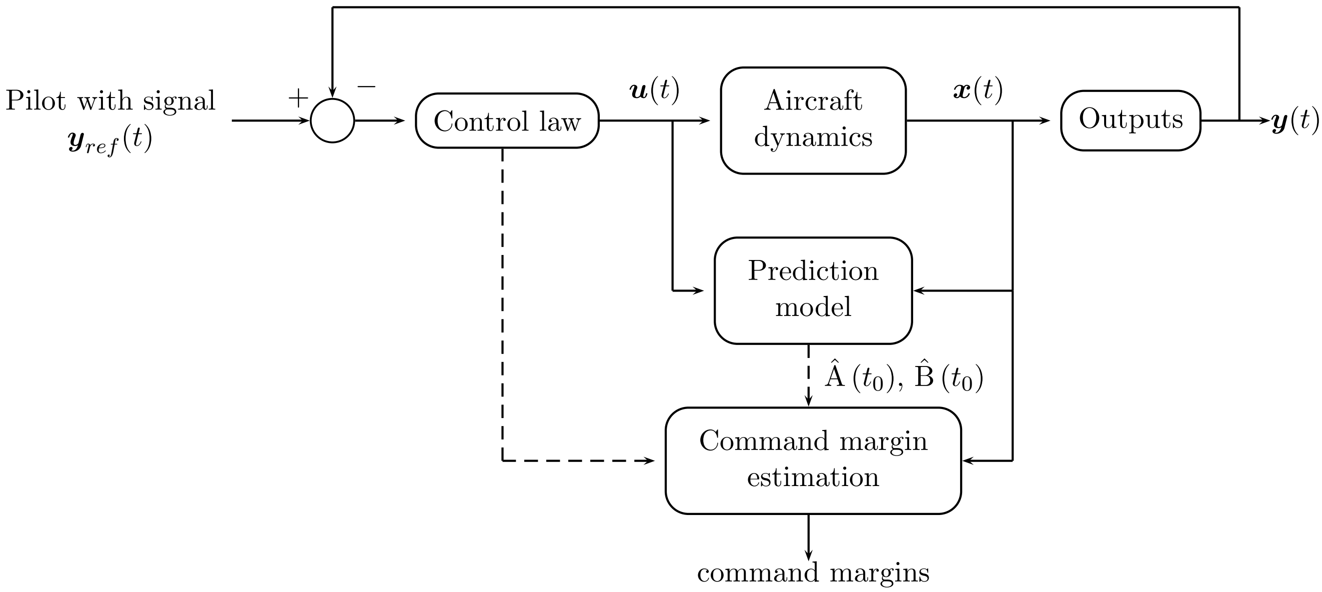


Figure 9-2: Real-time implementation

9-3 Simulations

Simulations are conducted to the study performance of the prediction model in conjunction with the set-up shown in figure 9-2. Only scenario A, as in section 8-3, is considered in the analysis.

9-3-1 Effects of the prediction model parameters

The parameters: γ and λ have the following effect on the behavior of the prediction model. Roughly stated, γ dictates the convergence rate of the prediction error signal, and λ influences the transient behavior. The adaption rate γ must be set as high as possible, in order to improve the rate of convergence of the prediction error signal $\tilde{\boldsymbol{x}}(t)$. The design criterion γ is then chosen accordingly to "tame" the large transients in the prediction error signal, but also in the estimates: $\hat{A}(t)$ and $\hat{B}(t)$. (Stepanyan & Krishnakumar, 2011) proposed the following for λ:

$$\lambda(t) = 2\sqrt{\gamma}\sqrt{\|\boldsymbol{x}(t)\|^2 + \|\boldsymbol{u}(t)\|^2} \tag{9-18}$$

Figures 9-3 and 9-5 illustrate the effects of λ on the transient behavior of $\tilde{\boldsymbol{x}}(t)$, $\hat{A}(t)$, and $\hat{B}(t)$. The adaption rate $\gamma = 10000$ and $\mathrm{P} = I$. As can be seen in the figures, the performance of the prediction model strongly depends on the setting of λ. When $\lambda = 0$, large oscillations occur in $\tilde{\boldsymbol{x}}(t)$, $\hat{A}(t)$, and $\hat{B}(t)$, as shown by the blue lines in the figures. When λ is set according to eq(9-18), the high frequency oscillations are reduced.

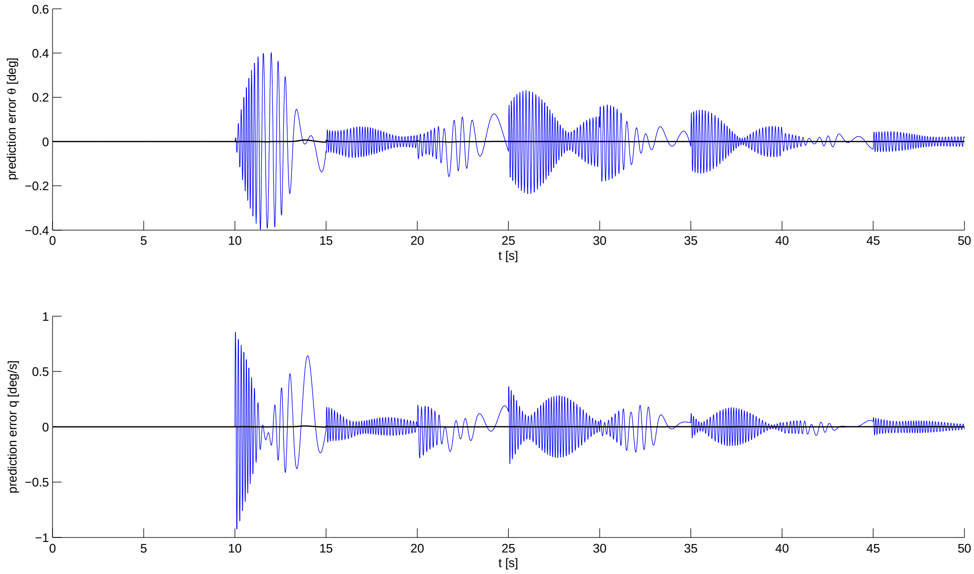


Figure 9-3: The prediction error $\tilde{x}(t)$ for the state trajectory shown in figure 9-4. The blue lines denote the prediction error for the settings: $\gamma = 1000$, $P = I$ and $\lambda = 0$. The black lines denote the prediction error for the settings: $\gamma = 1000$, $P = I$ and λ set according to eq(9-18).

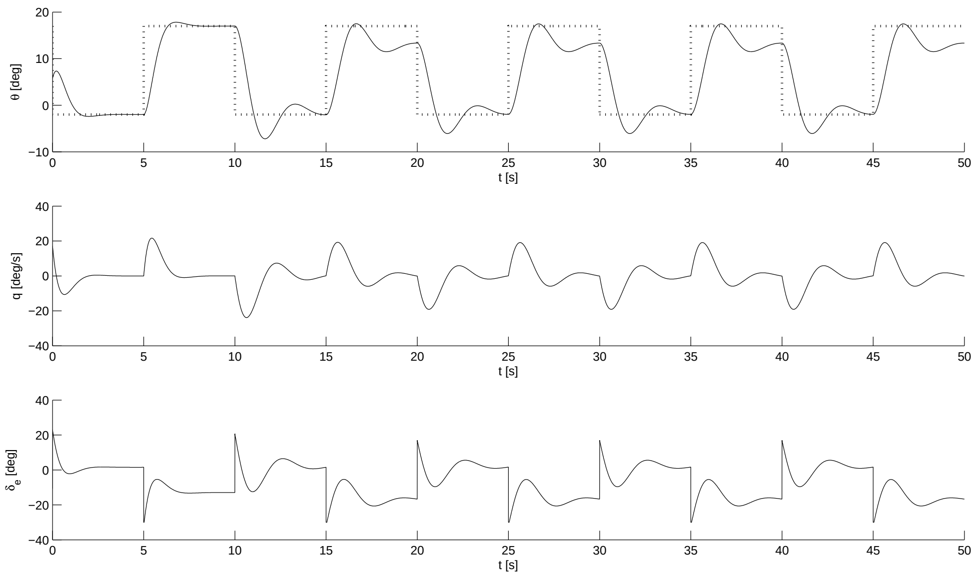


Figure 9-4: The state and input as a function of time. The top graph displays the pitch angle θ and reference pitch attitude θ_{ref} versus time. The center graph shows the pitch rate q versus time. The bottom graph shows the elevator deflection versus time.

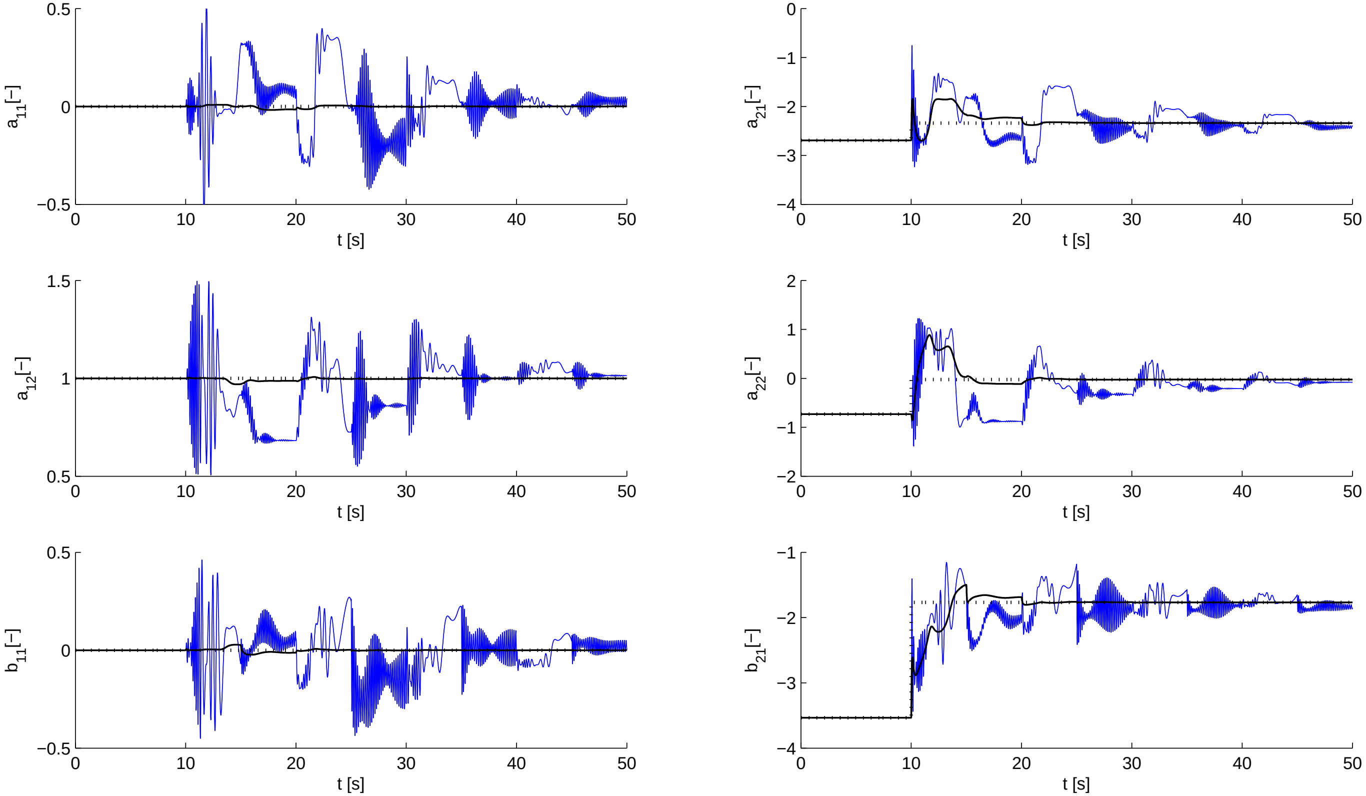


Figure 9-5: The adaptive parameters $\hat{\mathrm{A}}(t)$ and $\hat{\mathrm{B}}(t)$ for the state trajectory shown in figure 9-4. The blue lines denote the estimates obtained with the settings: $\gamma = 1000$, $P = I$ and $\lambda = 0$. The black continuous lines denote the estimates obtained with the settings: $\gamma = 1000$, $P = I$ and λ set according to eq(9-18). The black dotted lines denote the actual values of the elements in A and B.

9-3-2 Convergence of the adaptive parameters

The system identification approach using the prediction model stands different from ordinary parameter estimation techniques. The prediction model does not attempt to estimate the unknown parameters of the off-nominal model directly. Instead, it only adapts the parameters: $\Delta\hat{\mathrm{A}}(t)$ and $\Delta\hat{\mathrm{B}}(t)$, such that the predicted state follows the actual state. Although eq(9-16) mimics the input-output behavior of eq(9-3), internally they may be very different. For example, when certain modes of the system are not excited, the prediction model will not detect certain anomalies in the system. Consequently, the parameters of the prediction model will not "learn" the off-nominal conditions required to estimate the correct margins.

In figure 9-6, the adaptive parameters: $\hat{\mathrm{A}}(t)$ and $\hat{\mathrm{B}}(t)$ are shown along with the true values of A and B. The blue lines denote the estimates obtained with the trajectory shown in figure 9-7(a). The black lines denote the estimates obtained with the trajectory of figure 9-7(b). When the reference command is constant for a prolonged period, the prediction model tends to not learn anything new about the system. This can especially be noticed for the time period from 10 to 20 seconds where $\hat{\mathrm{A}}(t)$ and $\hat{\mathrm{B}}(t)$ settle down to a value, other than A and B. The estimates obtained with the trajectory of figure 9-7(b) appear to converge faster to the true values of the system. This is caused by the fact that the reference command in figure 9-7(a) switches from value at a higher frequency. Every time the reference command changes from value, the system gets excited and the prediction model gets to "learn" more about the off-nominal conditions.

Judging from the simulations, the plant needs to be sufficiently excited in order for $\hat{\mathrm{A}}(t)$ and $\hat{\mathrm{B}}(t)$ to converge to respectively A and B. Consequently, there seems to be no real benefit of using the prediction model as opposed to other system identification methods. Although the author of the report feels that no strong conclusions can be made on this matter, because not enough time has been invested to study the properties of the prediction model thoroughly.

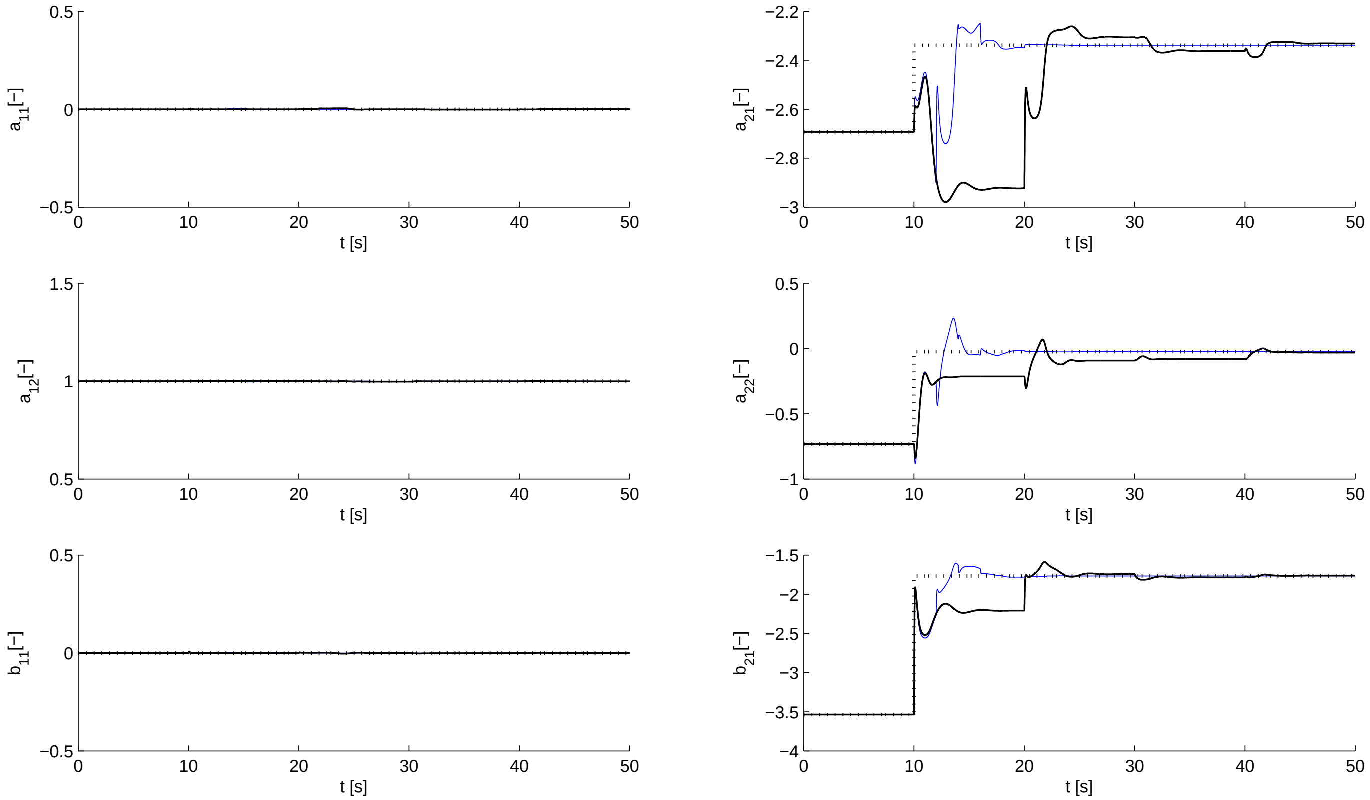
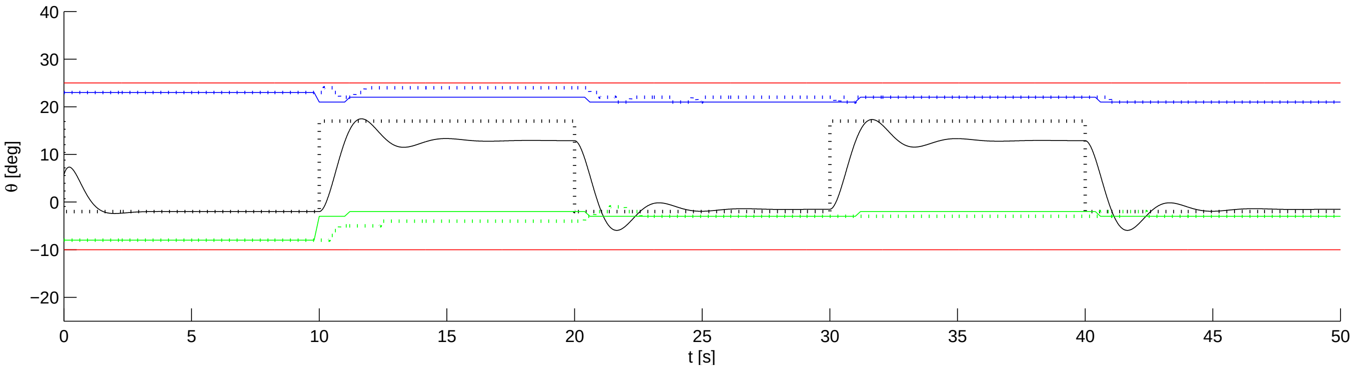
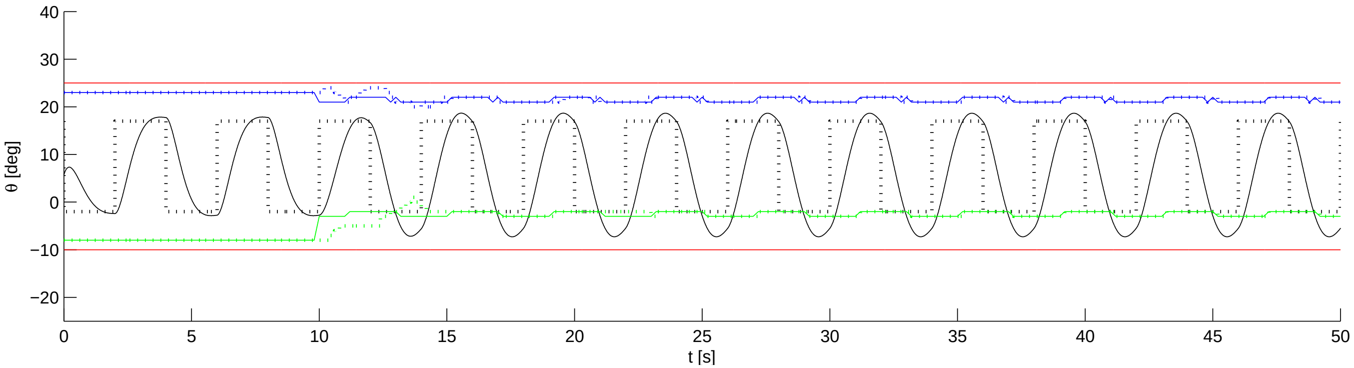


Figure 9-6: The adaptive parameters $\hat{\mathrm{A}}(t)$ and $\hat{\mathrm{B}}(t)$ for the state trajectory shown in figure 9-7. The blue lines denote the estimates obtained with the trajectory in figure 9-7(a). The black (continuous) lines denote the estimates obtained with the trajectory in figure 9-7(b). The black dotted lines denote the actual values of the elements in A and B. The prediction model settings are: $\gamma = 1000$, $P = I$ and λ set according to eq(9-18).



(a) A low frequency block signal is provided as a reference command.



(b) A high frequency block signal is provided as a reference command.

Figure 9-7: Margins estimated with the prediction model. The upper and lower limits of the margins are denoted respectively by the blue and green continuous lines. The dotted blue and green lines denote the actual margins (estimated with the true model parameter). The envelope limits are denoted in red. The state trajectory is denoted by the black continuous line. The reference command signal is denoted by the black dotted line.

Chapter 10

Conclusions

The goal of this thesis project was to develop a procedure to compute "safety margins" for reference command signals of aircraft control systems, such that predefined state constraints are never violated. Furthermore, the margins have to be estimated in real-time for system with uncertain plant dynamics. An optimal control framework was proposed to estimate these margins. The framework requires complete information on the system, for which a prediction model was used to obtain approximation of the dynamics during off-nominal conditions. The global concept of estimating margins was illustrated on a simplified pitch dynamics model with state limitations on the pitch attitude.

The optimal control framework involved optimizing a cost functional over a space of admissible command signals. The sign of the cost functional indicates whether a state trajectory of the system can violate the envelope in a given time window. This property was exploited to find margins for the reference command signals in an iterative procedure. The optimization of the cost functional was done using DP principles. Essentially, a time-dependent Hamilton-Jacobi-Bellman PDE had to be solved in order to find the extrema of the cost functional. The so-called viscosity solution of this PDE described the value function of a terminal cost optimal control problem.

Well established numerical schemes, collectively referred to as level set methods, exist to solve PDEs of the kind addressed in this thesis. However, they are computationally expensive, making them unsuitable for real-time applications. This motivated to look into alternative solution methods. A scheme was developed to approximate the solution of the PDE using regressional techniques. The PDE was transcribed into a non-linear optimization problem which subsequently was solved as a sequence of least-squares problems through the employment of a value-iteration scheme. A shortcoming of the proposed method is the unknown convergence properties of the value-iteration scheme. Although an independent proof was derived for the convergence of the discrete-time equivalent of the scheme, this was insufficient to claim also convergence for the continuous-time case. Hence, empirical studies were needed to further back the use of the scheme in the algorithm.

The new methodology to solve the PDE was tested on the simplified pitch dynamics model with multivariate simplex splines as regressors. Results showed that the splines have difficulties approximating non-smooth parts of the viscosity solution. Especially in the presence of non-smooth boundary conditions, the approximations tend to get heavily distorted. Otherwise, sufficiently accurate results may be obtained with significant reductions in computational time.

A simple algorithm was developed to determine pitch reference margins for the simplified pitch dynamics model. Experiments were conducted to study the behavior of the margins in response to abrupt changes in the dynamics and flight envelope. Simulations confirmed that envelope excursions are avoided when the reference command signal remains within the margins. On the other hand, a prolonged neglect of the margins is capable of steering the aircraft out of the flight envelope. The excursions can be anticipated by a pilot through the rapid shrinkage of the margins prior to an envelope violation. However, given the time-scale in which these events occur, it is postulated that the proposed framework for envelope protection will be more suitable for outer-loop control variables of aircraft control systems. Variables such as airspeed and flight-path angle evolve at a slower pace, giving enough response-time for the pilot to take countermeasures.

To estimate command margins during off-nominal conditions, a system identification procedure had to be implemented to determine the uncertain plant dynamics. The dynamics needed to be approximated with a method that circumvents the persistence of excitation requirement, because deliberate excitation of the controls is not recommended during a failure condition. The idea was to use a prediction model that follows the input-output behavior of the plant by adaption of certain parameters. The adaptive laws are designed such that the predicted error converges to zero, irrespective of the input excitation given to the system. However, it was discovered that this convergence was not sufficient for estimating the margins correctly. The adaptive parameters need to first converge to the true model parameters, however that appeared to occur only when the plant got sufficiently excited. Judging from simulation results, there seems to be no real benefit of using the prediction model as opposed to other system identification methods. Mind however that no strong conclusions can be made on this matter because insufficient time has been invested to study the prediction model thoroughly.

Chapter 11

Recommendations

The priority in this thesis was to illustrate the entire concept of estimating aircraft command margins on a simplified dynamical system. There are still major hurdles to be overcome, before the proposed concept can be implemented on realistic aircraft systems. These hurdles are respectively the high computational costs for solving the optimal control problem, and the challenges associated with on-line identification of off-nominal conditions. The following recommendations are given for further research.

The first recommendation is to explore other methods for solving the optimal control problem. It is important to recognize that the cost functional needs to be optimized only along the trajectory of the system. In DP, the extremum of a cost functional is found for an entire sub-region of the state-space. This makes DP somewhat cumbersome for the application. Recommended is to exploit the connections of eq(4-1) with standard optimal control problems, such that other algorithms from the literature can be used instead. The papers of (Betts, 1998) and (Rao, 2009) provide a good survey on the various numerical methods in optimal control. The papers classify the numerical methods into direct and indirect methods. Whereas the former aims to solve the optimal control problems through first-order necessary conditions in calculus of variations and Pontryagin's maximum principle, the latter proposes a direct discretization of the cost functional. A particularly interesting direct approach is the Gauss-Pseudospectral method of (Benson, 2005). The method proposes to approximate the state and input trajectory of a dynamical system with Legendre polynomials at so-called Gauss collocation points. The cost functional is effectively transcribed into a non-linear program where the resulting KKT conditions are exactly equivalent to the discretized version of the first-order necessary conditions for optimality. As a consequence, the method shares the accuracy of indirect methods, while preserving the robustness of direct methods.

The second recommendation is to make further improvements to the spline method of chapter 7. The performance of the method can be further enhanced by optimizing the configuration of the spline model. A suggestion is to develop an adaptive scheme which re-configures the parameters of the spline based on the residual approximation error. The computational costs can be further reduced by implementing a gradient descent method for solving the system of algebraic equations. Direct elimination of the smoothness constraints in eq(7-1) may also be useful for reducing computational costs. More studies need to be conducted to compare the computational complexity of the spline method with level set algorithms. The convergence of the value-iteration procedure also needs to be proven. A recommendation is to first conduct more empirical studies for value-iteration, especially with nonlinear systems. Researchers are encouraged to study the survey paper of (Wang, Zhang, & Liu, 2009) in order to gain additional insight into the field of "adaptive dynamic programming". The survey gives references to work such as (Vamvoudakis, 2011), where on-line algorithms were developed to find the value function of an infinite-horizon integral cost functional.

The final recommendation is to perform more studies to improve the detection & identification procedure of off-nominal conditions. Given the nature of LOC incidents, accurate estimates of the off-nominal dynamics need to be obtained in a very short time-span. In addition, deliberate excitation of the inputs has to be avoided, which further complicates matters. In this thesis, little has been done to address these aspects of the problem. The majority of the work focused on finding an effective method to solve the optimal control problem. Nevertheless, a first step is to conduct more simulations with the prediction model.

Appendix A

Level set methods

A-1 Introduction

Analytic solutions are rarely found for the PDEs in eq(4-7) and eq(4-12). Furthermore, classical solutions often do not exist for these PDEs as eq(4-5) and eq(4-9) may form discontinuities in their derivatives. Anyhow, well established methods do exist to solve these HJB PDEs numerically. The numerical schemes go under the name of level set methods, and are specifically designed to obtain viscosity solutions.

This appendix chapter aims to give a brief introduction to level set algorithms. The appendix primarily focuses on the algorithms used to obtain solutions for eq(4-12). A more comprehensive treatment of level set methods can be found in the books of (Osher & Fedkiw, 2003) and (Sethian, 1999). The Level Set Toolbox documentation of (Mitchell, 2007) is also a good source of information, especially when considering the practical implementation of the schemes.

The appendix is organized as follows. Section A-2 discusses the basic considerations for approximating the value functions on a finite grid. Section A-3 discusses the finite difference methods used to evaluate the separate PDE terms. Section A-4 describes the overall algorithm. Section A-5 discusses the general shortcomings of the algorithms.

A-2 Numerically approximating the value function on a finite grid

Level set algorithms aim to solve PDEs, like the one in eq(4-12), over a fixed, finite grid of the state-space. The spatial derivatives are evaluated with finite difference methods and the grid values are evolved over time with an explicit integration scheme.

Note that subsequently, the value function V_2 is approximated for only a local region of the state-space, i.e. for some $\Omega \subset \mathbb{R}^n$. Since V_2 is formally defined over the entire state-space, errors will be introduced automatically in the numerical solutions due to incorrect boundary conditions. Consequently, a sufficiently large computational domain has to be taken to marginalize these errors. The size of this computational domain will vary per problem and depends on how far the characteristic waves propagate in the state-space.

For illustration purposes, the working principles of the level set algorithms is described for the one dimensional case. Extension to the multivariate case is relatively straightforward and can be found in the aforementioned literature.

Figure A-1 shows the grid over which eq(4-12) is solved. Notice that every point (t_j, x_i) within the grid, has to satisfy the following relation[1]:

$$\frac{\partial V\left(t_j, x_i\right)}{\partial t} + H\left(x_i, \frac{\partial V\left(t_j, x_i\right)}{\partial x}\right) = 0 \tag{A-1}$$

Furthermore, the grid points at the boundary are equal to:

$$V\left(t_N, x_i\right) = l\left(x_i\right) \tag{A-2}$$

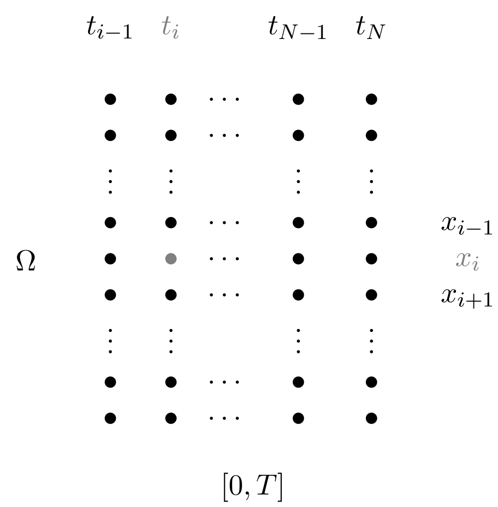


Figure A-1: The grid for solving V_2 over the domain: $[0, T] \times \Omega$

A-3 Approximation of the PDE terms

The PDE terms are approximated with finite difference methods. The manner in which these approximations have to be made requires some consideration in order to ensure stability of the overall algorithm.

A-3-1 The spatial derivatives

The spatial derivatives of the value function at a certain grid point is approximated from the left and right side of the respective node. In the case of eq(A-1), the left approximation is

[1] For sake of brevity, the subscript 2 is omitted and t_0 is set to 0

given by:

$$\frac{\partial V^{-}(t_j, x_i)}{\partial x} = \frac{V(t_j, x_i) - V(t_j, x_{i-1})}{\Delta x} \tag{A-3}$$

whereas the right approximation is given by:

$$\frac{\partial V^{+}(t_j, x_i)}{\partial x} = \frac{V(t_j, x_{i+1}) - V(t_j, x_i)}{\Delta x} \tag{A-4}$$

Note that eq(A-3) and eq(A-4) are both first-order $\mathcal{O}(\Delta x)$ approximations of the gradient. Higher order approximations for $\frac{\partial V^{-}}{\partial x}$ and $\frac{\partial V^{+}}{\partial x}$ can be obtained with *Essentially Non-Oscillatory* (ENO) and *Weighted Essentially Non-Oscillatory* (WENO) schemes. The Level Set Toolbox, as in (Mitchell, 2007), has predefined functions which allows one to implement these higher-order approximations.

A-3-2 The temporal derivative

The temporal derivative is evaluated with an explicit integration scheme. The explicit scheme allows one to express the values of the nodes at t_{j-1} explicitly as relation of the nodes at t_j. The most simplest version of such a scheme is the forward Euler method which is of firs-order accuracy $\mathcal{O}(\Delta t)$. Given that the integration is done backwards in time, the forward Euler approximation is given by:

$$\frac{\partial V(t_j, \boldsymbol{x}_i)}{\partial t} = \frac{V(t_j, x_i) - V(t_{j-1}, x_i)}{\Delta t} \tag{A-5}$$

Higher order Runge-Kutta (RK) schemes also exist for the temporal derivative. The Level Set Toolbox allows one to implement these higher order approximations.

A-3-3 The Hamiltonian term

In order to ensure numerical stability, the Hamiltonian term is either approximated in the upwind direction, or artificial diffusion is added to the dampen the system. For systems with no inputs, it is more practical to use upwind differencing. For the generic case however, the Lax-Friedrich (LF) approximation should be used.

Upwind differencing

When the system is autonomous, eq(A-1) reduces to the convection equation:

$$\begin{aligned} \frac{\partial V(t_j, x_i)}{\partial t} + \frac{\partial V(t_j, x_i)}{\partial x} f(x_i) &= 0 && \text{(A-6a)} \\ V(t_N, x_i) &= l(x_i) && \text{(A-6b)} \end{aligned}$$

where $f(x)$ is the externally generated velocity field. The term $\frac{\partial V(t_j,x_i)}{\partial x}^T f(x_i)$ is evaluated with an upwind scheme. The upwind scheme approximates the spatial derivatives by biasing the finite difference stencil in the direction where the characteristic information is coming from. If $f(x_i) > 0$, the right approximation $\frac{\partial V^{+}(t_j,x_i)}{\partial x}$ is used. Similarly, the left approximation $\frac{\partial V^{-}(t_j,x_i)}{\partial x}$ is used when $f(x_i) < 0$.

Lax-Friedrichs approximation

For the more generic case, the *Lax-Friedrichs* (LF) approximation is used instead:

$$\hat{H} = H\left(\boldsymbol{x}, \frac{\boldsymbol{p}^{+} + \boldsymbol{p}^{-}}{2}\right) - \boldsymbol{\alpha}^{T}\frac{\boldsymbol{p}^{+} - \boldsymbol{p}^{-}}{2} \tag{A-7}$$

In eq(A-8), the second term represents the artificial dissipation needed to stabilize the numerical scheme. For the one dimensional case, the LF approximation of the Hamiltonian is given by:

$$\hat{H} = H\left(x_i, \frac{p^{+} + p^{-}}{2}\right) - \alpha\frac{p^{+} - p^{-}}{2} \tag{A-8}$$

where $p^{+} = \frac{\partial V^{+}(t_j,x_i)}{\partial x}$ and $p^{-} = \frac{\partial V^{-}(t_j,x_i)}{\partial x}$.

The dissipation coefficient α increases the amount of artificial dissipation and reduces the quality of the solution. Ideally, the objective it is to chose α as small as possible without inducing oscillations (or other nonphysical phenomena) into the numerical solution. In the simulations performed in this thesis, the dissipation coefficient is obtained through maximizing:

$$\alpha = \max\left|\frac{\partial H}{\partial p}\right| \tag{A-9}$$

over the entire computational domain.

A-4 The overall numerical scheme

The algorithms for solving eq(4-12) are summarized here below.

Algorithm A.1: ***Level set algorithm using upwind scheme***
Consider eq(A-6). Initialize grid points at t_N with $l(x_i)$, i.e. $V(t_N, x_i) = l(x_i)$. Repeat the following steps until $t_j = 0$:

1. *Evaluate $\partial V/\partial \boldsymbol{x}$ in the upwind direction for all grid points at time t_j , i.e.*

   if $f(x_i) > 0$, then $\frac{\partial V^{+}(t_j,x_i)}{\partial t} \rightarrow \frac{\partial V(t_j,x_i)}{\partial t}$

   and,

   if $f(x_i) < 0$, then $\frac{\partial V^{-}(t_j,x_i)}{\partial t} \rightarrow \frac{\partial V(t_j,x_i)}{\partial t}$

2. *Evaluate V for all grid points at time t_{j-1} with:*

$$V(t_{j-1}, x_i) = V(t_j, x_i) - \Delta t\frac{\partial V(t_j, x_i)}{\partial x} f(x_i)$$

Algorithm A.2: ***Level set algorithm with LF approximation***
Consider eq(A-1). Initialize grid points at t_N with $l(x_i)$, i.e. $V(t_N, x_i) = l(x_i)$. Repeat the following steps until $t_j = 0$:

1. *Evaluate the LF approximation for all grid points at time t_j:*

$$\hat{H}\left(t_j, x_i\right) = H\left(x_i, \frac{1}{2}\left(\frac{\partial V^{+}\left(t_j, x_i\right)}{\partial x} + \frac{\partial V^{-}\left(t_j, x_i\right)}{\partial x}\right)\right) - \frac{1}{2}\alpha\left(\frac{\partial V^{+}\left(t_j, x_i\right)}{\partial x} - \frac{\partial V^{-}\left(t_j, x_i\right)}{\partial x}\right)$$

   where the artificial dissipation coefficient α is obtained through eq(A-9).

2. *Evaluate V for all grid points at time t_{j-1} with:*

$$V\left(t_{j-1}, x_i\right) = V\left(t_j, x_i\right) - \Delta t \hat{H}\left(t_j, x_i\right)$$

A-4-1 Stability and convergence properties

The algorithms A.1 and A.2 converge to the correct solution if, and only if, the algorithms are both consistent and stable. Since the approximation error converges to zero as $\Delta t \rightarrow 0$ and $\Delta x \rightarrow 0$, the algorithms are known to be consistent. Stability, on the other hand, needs to be enforced through the Courant-Friedrichs-Lewy (CFL) condition. The CLF condition states that the numerical waves: $\Delta x / \Delta t$ have to propagate as fast as the physical waves. For the univariate case, the CLF condition is:

$$\Delta t < \frac{\Delta x}{\max\left\{\left|\frac{\partial H}{\partial p}\right|\right\}} \tag{A-10}$$

where $\max\left\{\left|f\left(x_i\right)\right|\right\}$ is taken over the entire computational domain. For the autonomous case in eq(A-6), $\frac{\partial H}{\partial p} = f\left(x_i\right)$. Notice also when the grid becomes more dense, the CLF condition imposes a more severe restriction on the maximum allowable time-step. For the practical implementation, this restriction enforces a lower limit on the grid accuracy Δx.

A-5 Drawbacks of level set algorithms

The main drawback of level set algorithms is the *curse-of-dimensionality*, which is inherent to all DP problems. Since the grid size grows exponentially with the state dimension, the algorithms become intractable as the size of state vector grows. Consequently, level set methods are applicable to only relatively small-scale problems. In order to extend their applicability to more real-life problems, (Kitsios & Lygeros, 2005) proposed to use time-scale separation to divide the computation into several parts. In his paper, this approach was used to compute the final glide back envelope of a reusable launch vehicle. Regardless, the algorithms are still unsuitable for real-time applications.

Appendix B

Multivariate simplex splines

B-1 Introduction

A spline is a piecewise polynomial function with predefined continuity between the separate pieces. They are known to be highly effective tools for function approximation as they allow one to fit data which otherwise is too complex for one single function. In the multivariate case, splines can for instance be defined over special geometric structures called triangulations. These so-called simplex splines are used in chapter 7 to approximate the value function in eq(4-9).

This appendix chapter aims to give a brief introduction to multivariate simplex spline theory. The appendix acts as complement to chapter 7 for readers who are not familiar with the subject. For a more thorough understanding however, the author of this report recommends the PhD thesis of (de Visser, 2011) and the references therein.

The appendix is organized as follows. Section B-2 introduces the basic theory of multivariate simplex splines. Section B-3 discusses their application to non-linear function approximation.

B-2 Simplex spline theory

A multivariate simplex spline can be described as a piecewise-polynomial function of certain *continuity order* defined over a *triangulation*. The polynomial functions within each *simplex* of the triangulation are furthermore expressed in *B-form* using *barycentric coordinates*.

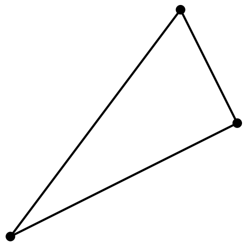
What follows next is a further elaboration on the terminology: simplex, triangulation, barycentric coordinates, B-form polynomial, and continuity order.

B-2-1 Preliminaries on simplices, triangulations and Barycentric coordinates

An *n-simplex* is an n-dimensional polytope with $n + 1$ vertices. As illustrated in figure B-1, a line segment is a simplex in one dimensional space. In two dimensional space, the simplex becomes equivalent to a triangle, which subsequently turns into a tetrahedron in three dimensional space. Formally, the simplex can be defined as follows.

Definition B.1: ***Simplex***
An n-simplex is an n-dimensional polytope with $n+1$ vertices. Let V *be a set of* $n+1$ *unique, non-degenerate, points in n-dimensional space, i.e.* $\mathrm{V} := \{\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\} \in \mathbb{R}^n$*. Then the n-simplex* Δ *is the convex hull of* V*.*

(a) 1-D simplex: a line segment.


(b) 2-D simplex: a triangle.

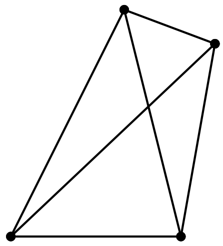

(c) 3-D simplex: a tetrahedron.

Figure B-1: Examples of simplices

For a simplex, a local coordinate system can be defined in terms of *Barycentric coordinates*. In the Barycentric coordinate system, every point $\boldsymbol{x} \in \mathbb{R}^n$ is expressed in terms of a unique, normalized, weighted vector sum of the simplex vertices:

$$\boldsymbol{x} = \sum_{i=0}^{n} \boldsymbol{v}_i b_i, \quad \sum_{i=0}^{n} b_i = 1 \tag{B-1}$$

The mapping between Barycentric and Cartesian coordinates is linear and invertible. The transformation from Cartesian to Barycentric coordinates: $\mathrm{T}_{C\mapsto B}$ can be denoted by:

$$\boldsymbol{b} = \mathrm{T}_{C\mapsto B}\left(\boldsymbol{x}\right) = \mathcal{A}\boldsymbol{x} + \mathcal{B} \tag{B-2}$$

where $\mathcal{A} \in \mathbb{R}^{(n+1)\times n}$ and $\mathcal{B} \in \mathbb{R}^{(n+1)\times 1}$ are obtained through:

$$\begin{bmatrix} \mathcal{A} & \mathcal{B} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_0 & \cdots & \boldsymbol{v}_n \\ 1 & \cdots & 1 \end{bmatrix}^{-1} \tag{B-3}$$

and $\boldsymbol{b} = \begin{bmatrix} b_1 & b_2 & \ldots & b_n \end{bmatrix}^T$. The transformation from Cartesian to Barycentric coordinates is, in principle, a linear mapping from a n-dimensional Euclidian space ($\mathbb{R}^n$) on to a n-dimensional hyper plane in $(n+1)$-dimensional Euclidian space ($\mathbb{R}^{n+1}$). The columns of $\mathcal{A}$ basically represent the partial derivatives of the Barycentric coordinates w.r.t. the Cartesian coordinates, i.e.

$$\mathcal{A} := \begin{bmatrix} \frac{\partial \boldsymbol{b}}{\partial x_1} & \cdots & \frac{\partial \boldsymbol{b}}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{(n+1)\times n}$$

A polygonal domain in $\mathbb{R}^n$ can be partitioned into many non-overlapping simplices. Such a partition is called a *triangulation*. Figure B-2 shows a triangulation in two dimensional space. Formally, triangulations can be defined as follows.

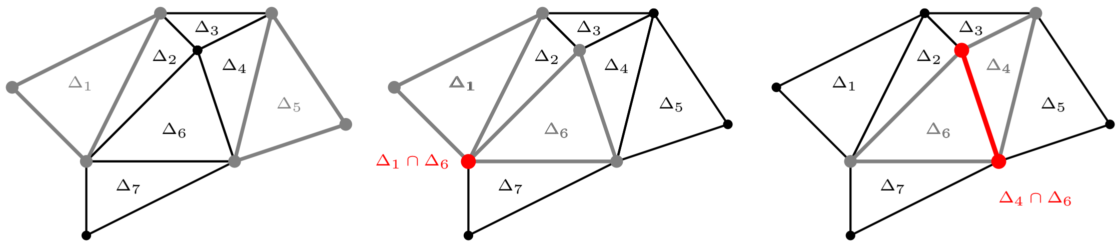
Definition B.2: ***Triangulation***
A triangulation $\mathcal{T}$ is a special decomposition of a domain in $\mathbb{R}^n$ into a set of J non-overlapping simplices:

$$\mathcal{T} := \bigcup \{\Delta_i,\ i = 1, 2, \ldots, J\} \tag{B-4}$$

in which $\forall \Delta_i, \Delta_j \in \mathcal{T}$, $i \neq j$, it holds that

$$\Delta_i \cap \Delta_j = \{\emptyset, \bar{\Delta}\} \tag{B-5}$$

where $\bar{\Delta}$ is a k-simplex with $0 \leq k \leq n-1$ (de Visser et al., 2009).



(a) $\Delta_1 \cap \Delta_5 = \emptyset$ (b) $\Delta_1 \cap \Delta_6 = \{$a 0-D simplex$\}$ (c) $\Delta_3 \cap \Delta_6 = \{$a 1-D simplex$\}$

Figure B-2: A triangulation in two dimensional space.

B-2-2 The B-form polynomial

The polynomial functions within every simplex of the triangulation are expressed in *B-form*. The B-form polynomial expression follows from the multinomial theorem. The multinomial theorem, which basically is a generalization of Newton's binomial theorem, states that a sum of $n+1$ numbers raised to the d-th power can be expanded into $\hat{d} = \frac{(d+n)!}{d!n!}$ terms by the relation:

$$(b_0 + b_1 + \ldots + b_n)^d = \sum_{\kappa_1+\kappa_2+\ldots+\kappa_n=d} \left(\frac{d!}{\kappa_1!\kappa_2!\ldots\kappa_n!} \prod_{i=0}^{n} b_i^{\kappa_i} \right) \tag{B-6}$$

The B-form polynomial is then defined as follows.

Definition B.3: ***B-form polynomial***
Let $b_0, b_1, \ldots, b_n$ in eq(B-6) be the barycentric coordinates with respect to some simplex Δ for a point $\boldsymbol{x} \in \mathbb{R}^n$. A polynomial function $p : \mathbb{R}^n \mapsto \mathbb{R}$ is expressed in B-form when the polynomial is expressed as a linear combination of the multinomial terms:

$$p(\boldsymbol{b}) = \sum_{\kappa_1+\kappa_2+\ldots+\kappa_n=d} c_{\boldsymbol{\kappa}} B^d_{\boldsymbol{\kappa}}(\boldsymbol{b}) \tag{B-7}$$

where:

$$B^d_{\boldsymbol{\kappa}}(\boldsymbol{b}) := \frac{d!}{\kappa_1!\kappa_2!\ldots\kappa_n!} \prod_{i=0}^{n} b_i^{\kappa_i}$$

and

$$\boldsymbol{\kappa} := (\kappa_0, \kappa_1, \ldots, \kappa_n) \in \mathbb{N}^{n+1}$$

In definition B.3, $c_{\boldsymbol{\kappa}}$ are referred to as the B-coefficients. The number of terms in the summation is given by

$$\hat{d} = \frac{(d+n)!}{d!n!} \tag{B-8}$$

For sake of convenience, the following vectorized notation is introduced [1]:

$$p(\boldsymbol{b}) = \boldsymbol{F}(\boldsymbol{b})^T \boldsymbol{c}_{\Delta_j} \tag{B-9}$$

where the terms in $\boldsymbol{F}(\boldsymbol{b}) \in \mathbb{R}^{\hat{d}}$ and $\boldsymbol{c}_{\Delta_j} \in \mathbb{R}^{\hat{d}}$ are lexicographically sorted by:

$$\boldsymbol{F}(\boldsymbol{b}) = \begin{bmatrix} B^d_{(d,0,0,\ldots,0)}(\boldsymbol{b}) \\ B^d_{(d-1,1,0\ldots,0)}(\boldsymbol{b}) \\ B^d_{(d-1,0,1,0\ldots,0)}(\boldsymbol{b}) \\ \vdots \\ B^d_{(0,\ldots,0,1,d-1)}(\boldsymbol{b}) \\ B^d_{(0,\ldots,0,0,d)}(\boldsymbol{b}) \end{bmatrix}, \quad \boldsymbol{c}_{\Delta_j} = \begin{bmatrix} c_{(d,0,0,\ldots,0)} \\ c_{(d-1,1,0\ldots,0)} \\ c_{(d-1,0,1,0\ldots,0)} \\ \vdots \\ c_{(0,\ldots,0,1,d-1)} \\ c_{(0,\ldots,0,0,d)} \end{bmatrix}$$

The B-form polynomial has several interesting properties. First of all, the basis terms always

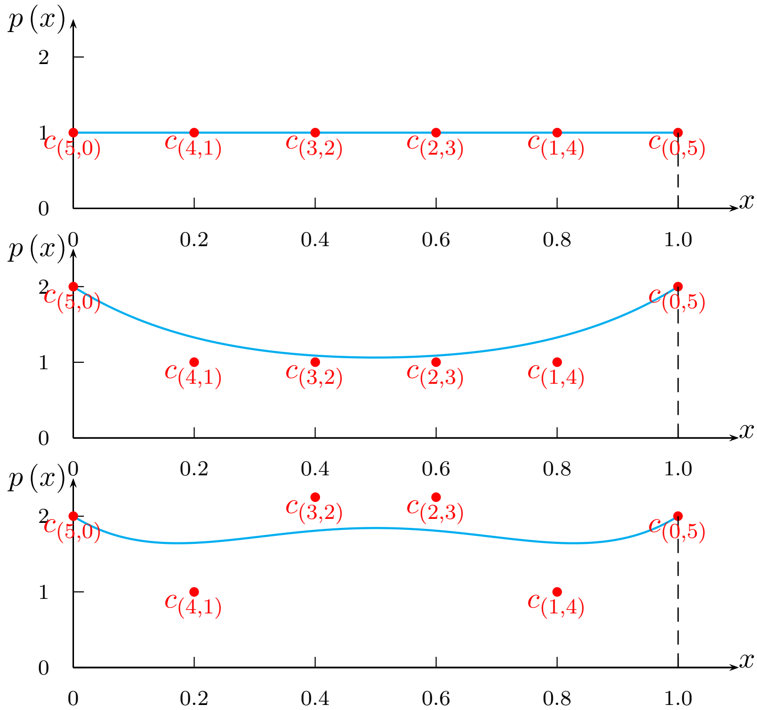


Figure B-3: Effect of the B-coefficients on the shape of the B-form polynomial function with degree $d = 5$.

sum-up to zero. Another interesting property is the spatial structure of the B-coefficients. Every B-coefficient is associated with a spatial location where it has maximum influence on

[1] The subscript Δ_j is used to emphasize that the coefficients refer a particular B-form polynomial function of the spline.

the shape of the polynomial. In terms of barycentric coordinates, the spatial locations of the B-coefficients are given by:

$$\boldsymbol{b}\left(c_{\boldsymbol{\kappa}}\right) = \frac{\boldsymbol{\kappa}}{d} \tag{B-10}$$

Figure B-3 illustrates the effect of the B-coefficients on the shape of the polynomial function for the univariate case.

B-2-3 Continuity order and inter-simplex constraints

In a simplex spline, a separate B-form polynomial is defined for every simplex of the triangulation. The polynomials are subsequently "knotted" together into one continuous function by the introduction of inter-simplex, linear equality constraints on the B-coefficients. The exact structure of the constraints depend on the *continuity order* desired for the spline functions and are discussed in (de Visser et al., 2009). For 0th order continuity, the required constraints are relatively straightforward: one simply needs to equate the B-coefficients sharing a common global position with one another. This is illustrated in Figure B-4 for two neighboring triangles with 4th degree polynomials.

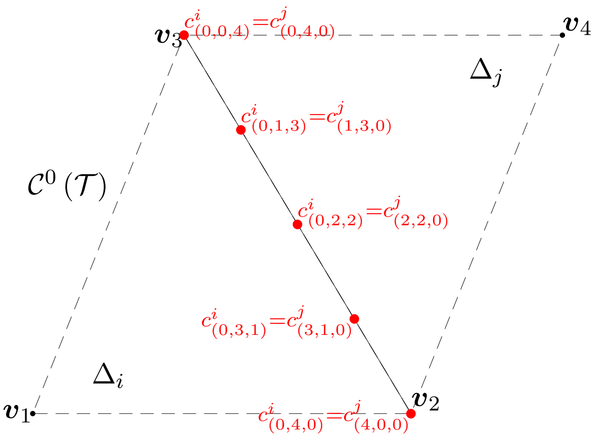


Figure B-4: Inter-simplex constraints on the B-coefficients for 0-th order continuity.

Regardless of the continuity order, the constraints can be summarized by s linearly independent relations denoted by the expression:

$$\mathrm{H}\boldsymbol{c} = 0 \tag{B-11}$$

where $H \in \mathbb{R}^{s \times J\hat{d}}$ and $\boldsymbol{c} \in \mathbb{R}^{J\hat{d}}$.

B-2-4 Computation of spline derivatives

The gradient of a spline function is relatively straightforward to determine as one simply evaluates the derivative of the respective B-form polynomials at the location of interest. The derivatives with respect to the Cartesian coordinates are found by direct application of the chain rule:

$$\frac{\partial p\left(\boldsymbol{b}\left(\boldsymbol{x}\right)\right)}{\partial \boldsymbol{x}} = \frac{\partial p\left(\boldsymbol{b}\right)}{\partial \boldsymbol{b}}\frac{\partial \boldsymbol{b}\left(\boldsymbol{x}\right)}{\partial \boldsymbol{x}} \tag{B-12}$$

In eq(B-12), the gradient with respect to the Barycentric coordinates may be expressed by:

$$\frac{\partial p\left(\boldsymbol{b}\right)}{\partial \boldsymbol{b}} = \sum_{\kappa_1+\kappa_2+\ldots+\kappa_n=d} c_{\boldsymbol{\kappa}} \frac{\partial B^d_{\boldsymbol{\kappa}}}{\partial \boldsymbol{b}}\left(\boldsymbol{b}\right) = \boldsymbol{c}^T_{\Delta_j} \frac{\partial \boldsymbol{F}\left(\boldsymbol{b}\right)}{\partial \boldsymbol{b}} \tag{B-13}$$

where $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{b}} \in \mathbb{R}^{\hat{d}\times(n+1)}$ is given by:

$$\frac{\partial \boldsymbol{F}\left(\boldsymbol{b}\right)}{\partial \boldsymbol{b}} = \begin{bmatrix} \frac{\partial B^d_{(d,0,0,\ldots,0)}(\boldsymbol{b})}{\partial b_0} & \frac{B^d_{(d,0,0,\ldots,0)}(\boldsymbol{b})}{\partial b_1} & \cdots & \frac{B^d_{(d,0,0,\ldots,0)}(\boldsymbol{b})}{\partial b_n} \\ \frac{\partial B^d_{(d-1,1,0\ldots,0)}(\boldsymbol{b})}{\partial b_0} & \frac{\partial B^d_{(d-1,1,0\ldots,0)}(\boldsymbol{b})}{\partial b_1} & \cdots & \frac{\partial B^d_{(d-1,1,0\ldots,0)}(\boldsymbol{b})}{\partial b_n} \\ \frac{\partial B^d_{(d-1,0,1,0\ldots,0)}(\boldsymbol{b})}{\partial b_0} & \frac{\partial B^d_{(d-1,1,0\ldots,0)}(\boldsymbol{b})}{\partial b_1} & \cdots & \frac{\partial B^d_{(d-1,1,0\ldots,0)}(\boldsymbol{b})}{\partial b_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial B^d_{(0,\ldots,0,1,d-1)}(\boldsymbol{b})}{\partial b_0} & \frac{\partial B^d_{(0,\ldots,0,1,d-1)}(\boldsymbol{b})}{\partial b_1} & \cdots & \frac{\partial B^d_{(0,\ldots,0,1,d-1)}(\boldsymbol{b})}{\partial b_n} \\ \frac{\partial B^d_{(0,\ldots,0,0,d)}(\boldsymbol{b})}{\partial b_0} & \frac{\partial B^d_{(0,\ldots,0,0,d)}(\boldsymbol{b})}{\partial b_1} & \cdots & \frac{\partial B^d_{(0,\ldots,0,0,d)}(\boldsymbol{b})}{\partial b_n} \end{bmatrix}$$

From eq(B-2) follows that:

$$\frac{\partial p\left(\boldsymbol{b}\left(\boldsymbol{x}\right)\right)}{\partial \boldsymbol{x}} = \boldsymbol{c}^T_{\Delta_j} \frac{\partial \boldsymbol{F}\left(\boldsymbol{b}\right)}{\partial \boldsymbol{b}} \mathcal{A}_j \tag{B-14}$$

B-3 Regression with simplex splines

Multivariate simplex splines can be used to approximate nonlinear functions of scattered data sets. In (de Visser et al., 2009), a linear regression scheme was presented to estimate the B-coefficients using standard parameter estimation techniques.

Consider a collection of data points:

$$\left(\boldsymbol{x}_i, y_i\right), \qquad i = 1, \ldots, N$$

scattered over a domain D. The objective is to find a spline function $S : \mathbb{R}^n \mapsto \mathbb{R}$ that best fits the data points. The following steps are taken.

The first step involves selecting suitable parameters for the spline model. This includes selecting a triangulation $\mathcal{T}$ for the domain D, the degree d for the polynomial functions, and, continuity order r for the spline function. The second step is to sort the data points according to their location within the triangulation, i.e.

$$\left(\boldsymbol{x}^j_i, y^j_i\right) \in \Delta_j, \qquad i = 1, \ldots, N_j$$

. and $N_1 + N_2 + \ldots + N_J = N$. The following matrices are then constructed:

$$\mathrm{X}^j = \begin{bmatrix} \boldsymbol{F}\left(\mathrm{T}_{C\mapsto B}\left(\boldsymbol{x}^j_1\right)\right)^T \\ \vdots \\ \boldsymbol{F}\left(\mathrm{T}_{C\mapsto B}\left(\boldsymbol{x}^j_{N_j}\right)\right)^T \end{bmatrix}, \qquad \boldsymbol{Y}^j = \begin{bmatrix} y^j_1 \\ \vdots \\ y^j_{N_j} \end{bmatrix}, \qquad j = 1, \ldots J$$

The final step is to solve the optimization problem:

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}\in\mathbb{R}^{J\hat{d}}} \left(\boldsymbol{Y} - \mathrm{X}\boldsymbol{c}\right)^T \left(\boldsymbol{Y} - \mathrm{X}\boldsymbol{c}\right), \quad \text{subject to the constraint: } \mathrm{H}\boldsymbol{c} = 0 \tag{B-15}$$

where:

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}^1 \\ \vdots \\ \boldsymbol{c}^J \end{bmatrix}, \mathrm{X} = \begin{bmatrix} \mathrm{X}^1 & & \\ & \ddots & \\ & & \mathrm{X}^J \end{bmatrix}, \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{Y}^1 \\ \vdots \\ \boldsymbol{Y}^J \end{bmatrix}, \text{with } \mathrm{X}^j \in \mathbb{R}^{N^j\times\hat{d}}, \boldsymbol{Y}^j \in \mathbb{R}^{N^j\times 1}$$

This optimization problem may be solved by inverting the Karun-Kuhn-Tucker (KKT) system:

$$\begin{bmatrix} Q & \mathrm{H}^T \\ \mathrm{H} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathrm{R} \\ 0 \end{bmatrix} \tag{B-16}$$

where $\mathrm{R} = \mathrm{X}^T\boldsymbol{Y}$. A unique solution exists for eq(B-15) if, and only if, the dispersion matrix $\mathrm{Q} = \mathrm{X}^T\mathrm{X}$ is positive definite on the kernel of H. Furthermore, the KKT matrix is guaranteed to be invertible when Q is non-singular and H has full row rank. (de Visser et al., 2009) showed that Q is non-singular when every simplex of the triangulation contains at least a minimum of $\hat{d}$ non-coplanar data points. The matrix H is known to have full row rank as well since it describes a set of s linearly independent relations as per eq(B-11).

Appendix C

Measurable functions

In section 3-3, the notion of a measurable signal was introduced to define the function space of admissible command signals in eq(3-4). The concept of measurable signals (or functions) is common mathematical terminology used in the field of measure theory. For basic understanding of the work in this thesis, there is no requirement to have a background in measure theory as the term "measurable" is used more as formality in the text. One can simply interpret a measurable signal as any piecewise continuous function, which includes also the class of bang-bang signals. This formulation was used also in (van Oort, 2011).

Regardless, what follows next is a concise description of measurable functions. To describe them will require the introduction of so-called σ-algebras and measurable spaces. The following definitions adapt the notation in (Hunter, unknown).

Definition C.1: ***σ-algebra***
A σ-algebra on a set X is a collection $\mathcal{A}$ of subsets of X such that:

1. $\emptyset, X \in \mathcal{A}$
2. *if* $A \in \mathcal{A}$, *then* $A^c \in \mathcal{A}$
3. *if* $A_i \in \mathcal{A}$ *for* $i \in 1, 2, 3, \ldots$ *then*

$$\bigcup_{i=1}^{\infty} A_i \in \mathcal{A}, \bigcap_{i=1}^{\infty} A_i \in \mathcal{A}$$

Definition C.2: ***Measurable space***
A measurable space $(X, \mathcal{A})$ is a non-empty set X equipped with a σ-algebra $\mathcal{A}$ on X.

Given a measurable space $(X, \mathcal{A})$, a measure $\mu : \mathcal{A} \mapsto [0, \infty]$ can be defined for that space. A measure is interpreted as a generalization of the concept of size, and satisfies the following properties.

Definition C.3: ***Measure***
A measure μ on a measurable space $(X, \mathcal{A})$ is a function: $\mathcal{A} \mapsto [0, \infty]$ such that:

1. $\mu(\emptyset) = 0$
2. *if $\{A_i \in \mathcal{A} : i \in 1, 2, 3, \ldots\}$ is a countable disjoint collection of sets in $\mathcal{A}$, then*

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \bigcup_{i=1}^{\infty} \mu(A_i)$$

Given two measurable spaces, a measurable function is defined as follows.

Definition C.4: ***Measurable function***
Let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be measurable spaces. A function $f : X \mapsto Y$ is measurable if the inverse image $f^{-1}(B) \in \mathcal{A}$ for every $B \in \mathcal{B}$.

A measurable function is a function that preserves the measurability of the underlying spaces in which the functions are defined. The property is independent of the type measure used, and solely depends on the σ-algebras.

Practically, almost any function on the real numbers is a measurable function. However, there exist some functions that are non-measurable. For example, the indicator function $\chi_E : X \mapsto \mathbb{R}$:

$$\chi_E(x) = \begin{cases} 1 & \text{if } x \in E \\ 0 & \text{if } x \notin E \end{cases} \tag{C-1}$$

is known to be unmeasurable if $E \subset X$ is a non-measurable set. Existence of non-measurable sets on the real numbers can be shown by the so-called Vitali sets, see also (Benedetto & Czaja, 2009) and the references therein.

Bibliography

Alwardi, H. A. H. (2010). *Efficient Algorithms for Solving Equations Hamilton-Jacobi-Bellman*. PhD thesis, University of Western Australia.

Aubin, J. P. (1991). *Viability Theory*. Birkhauser.

Awanou, G., & Lai, M. J. (2004). Trivariate spline approximations of 3d navier-stokes equations. *Mathematics of Computation*, *74*(250), 585–601.

Awanou, G., Lai, M. J., & Wenston, P. (2005). The Multivariate Spline Method for Scattered Data Fitting and Numerical Solutions of Partial Differential Equations. In G. Chen & M. J. Lai (Eds.), *Wavelets and splines* (pp. 24–75). Nashburo Press.

Bardi, M., & Capuzzo-Dolcetta, I. (2008). *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Boston: Birkhauser.

Barlow, J., Stepanyan, V., & Krishnakumar, K. (2011). Estimating Loss-of-Control: a Data-Based Predictive Control Approach. In *AIAA Guidance, Navigation, and Control Conference*. Portland, Oregon.

Bayen, A. M., Mitchell, I. M., Oishi, M., & Tomlin, C. J. (2007). Aircraft Autolander Safety Analysis Through Optimal Control-Based Reach Set Computation. *Journal of Guidance, Control, and Dynamics*, *30*(1), 68–77.

Belcastro, C. M., & Foster, J. V. (2010). Aircraft Loss-of-Control Accident Analysis. In *AIAA Guidance, Navigation, and Control Conference*.

Belcastro, C. M., & Jacobson, S. (2010). Future Integrated Systems Concept for Preventing Aircraft Loss-of-Control Accidents. In *AIAA Guidance Navigation, and Control Conference*.

Benedetto, J., & Czaja, W. (2009). *Integration and Modern Analysis* (First ed.). Boston: Birkhauser.

Benson, D. (2005). *A Gauss Pseudospectral Transcription for Optimal Control*. PhD thesis, Massachusetts Institute of Technology.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Betts, J. T. (1998). Survey of Numerical Methods for Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, *21*(2), 193–207.

Blanchini, F. (1999). Set invariance in control. *Automatica*, *35*.

Crandall, M. G., Evans, L. C., & Lions, P. L. (1984). Some Properties of Viscosity Solutions of Hamilton-Jacobi Equations. *Transactions of the American Mathematical Society*, *282*(2), 487–502.

Crandall, M. G., & Lions, P. L. (1983). Viscosity Solutions of Hamilton-Jacobi Equations. *Transactions of the American Mathematical Society*, *277*, 1– 42.

de Visser, C. (2011). *Global Nonlinear Model Identification with Multivariate Splines.* PhD thesis, Delft University of Technology.

de Visser, C., Chu, Q., & Mulder, J. (2009). A new approach to linear regression with multivariate splines. *Automatica*, *45*(12), 2903–2909.

Djeridane, B., & Lygeros, J. (2006). Neural approximation of PDE solutions: An application to reachability computations. In *Proceedings of the 45th ieee conference on decision and control* (pp. 3034–3039). Ieee.

Evans, L. C. (unknown). *An Introduction to Mathematical Optimal Control Theory.* Lecture notes. Department of Mathematics, University of California, Berkeley.

Falkena, W., Borst, C., Chu, Q., & Mulder, J. (2011). Investigation of Practical Flight Envelope Protection Systems for Small Aircraft. *Journal of Guidance, Control, and Dynamics*, *34*(4).

Fleming, W. H., & Soner, H. M. (2006). *Controlled Markov Processes and Viscosity Solutions* (Second ed.). Springer.

Goman, M. G., Khramtsovsky, A. V., & Kolesnikov, E. N. (2008). Evaluation of Aircraft Performance and Maneuverability by Computation of Attainable Equilibrium Sets. *Journal of Guidance, Control, and Dynamics*, *31*(2).

Hu, X.-l., Han, D.-f., & Lai, M. J. (2007). Bivariate splines of various degrees for numerical solution of partial differential equations. *SIAM J. Sci. Comput..*, *29*(3), 1338–1354.

Huang, C.-S., Wang, S., Chen, C., & Li, Z.-C. (2006). A radial basis collocation method for Hamilton Jacobi Bellman equations. *Automatica*, *42*(12), 2201–2207.

Hunter, J. K. (unknown). *Measure Theory.* Lecture notes. Department of Mathematics, University of California at Davis.

Jacobson, S. R. (2010). Aircraft Loss of Control Causal Factors and Mitigation Challenges. In *AIAA Guidance Navigation, and Control Conference* (pp. 1–18). Toronto.

Jordan, T., Langford, W., Belcastro, C., Foster, J., Shah, G., Howland, G., et al. (2004). Development of a Dynamically Scaled Generic Transport Model Testbed for Flight Research Experiments. In *Auvsi unmanned systems.* Arlington, VA.

Kirk, D. E. (1970). *Optimal Control Theory - An Introduction.* New York: Dover Publications, Inc.

Kitsios, I., & Lygeros, J. (2005). Final Glide-back Envelope Computation for Reusable Launch Vehicle Using Reachability. In *Proceedings of the 44th IEEE Conference on Decision and Control* (pp. 4059–4064). IEEE.

Krishnakumar, K., Stepanyan, V., & Barlow, J. (2011). Piloting on the Edge : Approaches to Flight Control Solutions. In *Space Mission Challenges for Information Technology SMC-IT.* Palo Alto, CA.

Kwatny, H. G., Dongmo, J. T., Bajpai, G., Yasar, M., Chang, B., & Belcastro, C. (2009). Aircraft Accident Prevention : Loss-of-Control Analysis. In *AIAA Guidance, Navigation, and Control Conference.* Chicago: AIAA.

Lai, M. J., & Schumaker, L. L. (2007). *Spline Functions on Triangulations.* Cambridge University Press.

Lygeros, J. (2004). On reachability and minimum cost optimal control. *Automatica*, *40*, 917 – 927.

Mitchell, I. (2002). *Application of Level set methods to control and reachability problems in continuous and hybrid systems.* PhD thesis, Stanford university.

Mitchell, I. (2007). *A Toolbox of Level Set Methods (Version 1.1) UBC CS TR-2007-11.*

Mitchell, I., Bayen, A., & Tomlin, C. (2005). A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, *50*(7), 947–957.

Osher, S., & Fedkiw, R. (2003). *Level Set Methods and Dynamic Implicit Surfaces* (Vol. 153; S. S. Antman, J. E. Marsden, & L. Sirovich, Eds.). New York: Springer-Verlag.

Rao, A. V. (2009). A survey of numerical methods for optimal control. *AAS*.

Sethian, J. A. (1999). *Level set methods and fast marching methods.* Cambridge University Press.

Seube, N. (2002). Viability analysis of an aircraft flight domain for take-off in a windshear. *Mathematical and Computer Modeling*, *36*(6), 633–641.

Stepanyan, V., Barlow, J., Krishnakumar, K., & Bijl, H. (2011). *Adaptive Estimation Based Loss of Control Detection and Mitigation.*

Stepanyan, V., & Krishnakumar, K. (2011). M-MRAC for Nonlinear Systems with Bounded Disturbances. In *50th IEEE Conference on Decision and Control* (pp. 5419–5424).

Tang, L., Roemer, M., Ge, J., Crassidis, A., Prasad, J. V. R., & Belcastro, C. (2009). Methodologies for Adaptive Flight Envelope Estimation and Protection. In *AIAA, Guidance, Navigation and Control Conference.*

Tang, Y., Atkins, E. M., & Sanner, R. M. (2007). Emergency Flight Planning for a Generalized Transport Aircraft with Left Wing Damage. In *AIAA Guidance, Navigation and Control Conference and Exhibit.*

Unnikrishnan, S. (2006). *Adaptive Envelope Protection Methods for Aircraft.* PhD thesis, Georgia Institute of Technology.

Vamvoudakis, K. G. (2011). *Online Learning Algorithms for Differential Dynamic Games and Optimal Control.* PhD thesis, The University of Texas at Arlington.

van Oort, E. R. (2011). *Adaptive Backstepping Control and Safety Analysis for Modern Fighter Aircraft.* PhD thesis, Delft University of Technology.

Wang, F.-y., Zhang, H., & Liu, D. (2009). Adaptive Dynamic Programming : An Introduction. *IEEE Computational Intelligence Magazine*, 39–47.

Wilborn, J. E., & Foster, J. V. (2004). Defining Commercial Transport Loss-of-Control : A Quantitative Approach. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit.* Providence: AIAA.