

Developments in Stochastic Fuel Efficient Cruise Control and Constrained Control with Applications to Aircraft

by

Kevin K. McDonough

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2015

Doctoral Committee:

Professor Ilya V. Kolmanovsky, Chair
Associate Professor Ella M. Atkins
Associate Professor Amy E.M. Cohn
Professor Demosthenis Teneketzis

”Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning.”

Winston Churchill, November 10, 1942

”Thousands of candles can be lighted from a single candle, and the life of the single candle will not be shortened. Happiness never decreases by being shared.”

Buddha

”Nothing has such power to broaden the mind as the ability to investigate systematically and truly all that comes under thy observation in life.”

Marcus Aurelius

©Kevin K. McDonough

2015

This is dedicated to my first engineering teacher and grandfather, Ralph Vogel. Thank you for all of the years of hands-on learning and for teaching me to always measure twice and cut once.

Acknowledgments

First and foremost, I would like to thank Professor Ilya Kolmanovsky for his guidance and instruction over the years. He is a great mentor, researcher, and friend. My thanks also goes to the Ford Motor Company and University of Michigan-Ford Alliance Project. This relationship afforded me the opportunity to work on real-world optimization problems and conduct vehicle testing. Next, I would like to thank the National Aeronautics and Space Administration under cooperative agreement NNX12AM54A and the National Science Foundation under award number 1130160 for their support of my research in constrained control.

I would like to acknowledge and thank Professor Ella Atkins of the University of Michigan. Her class on information systems profoundly changed the way I approach engineering problems. Thank you for continually challenging the way I view the world.

Many people have helped and guided me along the way. This list includes most prominently Professor Dennis Bernstein, Sweewarman Balachandran, Ray Yu, and Pedro Di Donato of the University of Michigan and Dr. Dimitar Filev, Dr. Anthony D'Amato, and Steve Swabowski of the Ford Motor Company. Thank you all for your help, guidance, and support.

I would like to specifically thank Professor Anouck Girard for encouraging and guiding me into the area of flight dynamics and controls in my undergraduate education.

I would also like to thank Professor Amy Cohn and Professor Demos Teneketzis for agreeing to participate in my dissertation committee. Your viewpoints and insights have been greatly welcomed, thank you very much.

Last, but not least, I would like to thank my friends and family, particularly my wife Leslie, for their support through this whole endeavor.

TABLE OF CONTENTS

Dedication	ii
Acknowledgments	iii
List of Figures	vii
List of Tables	xi
List of Appendices	xii
List of Abbreviations	xiii
Abstract	xiv
Chapter	
1 Introduction	1
1.1 Stochastic fuel-efficient cruise control	1
1.1.1 Vehicle speed control and optimization	1
1.1.2 Stochastic modeling of road grade and traffic speed	5
1.2 Constrained control with applications to aircraft	6
1.2.1 Safe, recoverable, and integral safe sets	7
1.2.2 The controller state governor and the controller state and refer- ence governor	9
1.3 Major contributions and relevant publications	10
1.3.1 Major contributions	10
1.3.2 Relevant publications	11
1.4 Dissertation outline	13
2 Modeling, Optimization, and Results for Fuel-Efficient Cruise Control	15
2.1 Longitudinal vehicle dynamics	16
2.2 Stochastic models of traffic speed and road grade	17
2.3 Cost Function and its ingredients	19
2.3.1 Fuel consumption	19
2.3.2 Travel time	21
2.3.3 Distance constraint violation penalty	21
2.4 Stochastic dynamic programming	22
2.5 Simulation case studies	23

2.5.1	In-traffic driving	24
2.5.2	Optimal vehicle-following	28
2.6	Vehicle experiments	31
2.7	Sensitivity analysis of the SDP policy results	33
2.7.1	Sensitivity to cost function parameters	33
2.7.2	Sensitivity to Markov chain model grid selection	35
3	A Practical Implementation Strategy of Time-Varying Fuel-Efficient Cruise Patterns	39
3.1	Rule based cruise control problem formulation	40
3.2	Rule based cruise control	41
3.2.1	Normal state - \mathcal{N}	42
3.2.2	Glide state - \mathcal{G}	43
3.2.3	Bleed state - \mathcal{B}	44
3.2.4	Recover state - \mathcal{R}	45
3.3	Optimization	46
3.4	Simulations and vehicle testing results	47
3.4.1	Simulation results	48
3.4.2	Test vehicle results	49
4	Advances in the Stochastic Modeling of Road Grade and Vehicle Speed for Fuel-Efficient Cruise Control	53
4.1	Assessing the similarity of Markov chain models using a divergence technique	53
4.2	Stochastic jump process for road grade prediction	56
4.2.1	Stochastic jump model	60
4.2.2	Stochastic jump model identification	61
4.2.3	Prediction using the stochastic jump process	62
4.3	Grade prediction comparison	63
4.4	Online learning of traffic patterns and road grade	64
5	Safe, Recoverable, and Integral Safe Sets with Aircraft Applications	72
5.1	Safe set	73
5.2	Recoverable set	74
5.3	Integral safe set	77
5.4	Comparison of safe, recoverable, and integral safe sets	78
5.5	Aircraft maneuver planning using safe, recoverable, and integral safe sets	83
5.6	Further examples of safe and recoverable sets for aircraft dynamics	87
5.6.1	Recoverable set examples for aircraft dynamics	89
5.6.2	Recoverable set dependence on flight condition	91
5.6.3	Set-bounded disturbance effects on safe sets	92
6	Scaling Between Safe and Recoverable Sets at Different Flight Conditions	96
6.1	Scalar scaling	97
6.2	Vector scaling	98
6.3	Numerical examples	98

6.3.1	Scalar scaling over a grid of trim conditions	98
6.3.2	Aircraft icing example	100
6.4	Sensitivity technique to predict changes in safe and recoverable sets . . .	105
7	Controller State and Reference Governor with Aircraft Applications	107
7.1	Controller state governor	108
7.2	Controller state and reference governor	111
7.3	Remarks	114
7.3.1	Linear quadratic proportional plus integral controller	114
7.3.2	Nonlinear systems	116
7.4	Constrained aircraft flight control	118
7.5	Constrained gas turbine engine control	119
8	Summary, Conclusions, and Future Work	126
8.1	Summary and conclusions	126
8.2	Future work	128
8.2.1	Stochastic fuel-efficient cruise control	128
8.2.2	Constrained control with aircraft applications	129
	Appendices	131
	Bibliography	140

LIST OF FIGURES

2.1	A schematic of the fuel-efficient speed control architecture.	16
2.2	A surface plot of road grade (left) and traffic vehicle speed (right) transition probabilities.	18
2.3	Example of the training and testing data for the neural network model of the fuel flow based on CarSim data. Note only a small sample of the 7,000 data points was selected in each plot for clarity.	20
2.4	Schematic of the Simulink control architecture.	25
2.5	Left: The percent fuel economy increase for the in-traffic driving case study in the non-porous traffic case. Right: The percent difference in average speed.	26
2.6	The time histories of the traffic analog vehicle (dotted) and the host vehicle (solid) speeds for the simulation 6 in the non-porous traffic case.	27
2.7	Left: The percent fuel economy increase for the in-traffic driving case study in the porous traffic case. Right: The percent difference in average speed.	27
2.8	Fuel economy improvement with SDP policies in the vehicle-following scenarios. The blue (first) set of bars are for the non-zero grade policies and the yellow (second) set of bars are for zero grade.	30
2.9	Left: Time history of traffic analog vehicle speed (dashed) and host vehicle (solid) for a part of the drive with non-zero grade. Right: Fuel consumption versus distance traveled of traffic analog vehicle speed (dashed) and host vehicle (solid).	30
2.10	Left: Time history of traffic analog vehicle speed (dashed) and host vehicle (solid) for a part of the drive with zero grade. Right: Fuel consumption versus distance traveled of traffic analog vehicle speed (dashed) and host vehicle (solid).	31
2.11	A similar vehicle setup what was used in the 2007 Ford Edge experimental vehicle.	32
2.12	Left: Time history of vehicle speed from vehicle testing. Right: Cumulative flow of the host vehicle (solid) versus traffic vehicle (dashed).	33
2.13	PFEI of in-traffic driving scenario for drive number 5 over a spread of different λ weights.	35
2.14	Adjusted PFEI of in-traffic driving scenario for drive number 5 over a spread of different λ . These results are the percent difference from the mean PFEI from Figure 2.13.	36
2.15	Left: The percent difference in fuel economy improvement statistics over the sixteen test drives with varying λ . Right: The percent difference in change in average vehicle speed statistics over the sixteen test drives with varying λ	36

2.16	Left: Percent fuel economy improvement for varying κ with different λ . Right: Percent difference in average speed for varying κ and with different λ	37
2.17	PFEI for the vehicle-following scenario with respect to different Markov chain models, see Table 2.2.	38
2.18	PDAS for the vehicle-following scenario with respect to different Markov chain models, see Table 2.2.	38
3.1	Examples of the vehicle speed profiles in the four RBCC states.	40
3.2	Allowed state transitions in the finite state machine for Rule Based Cruise Control (RBCC). Not shown are the transitions from a given state to itself which is a possibility in all states.	42
3.3	Simulation speed profile and RBCC states for case study 2.	49
3.4	Simulation speed profile and RBCC states for case study 3.	50
3.5	A section of the road grade along M-39 overlaid with the thresholds from case study 2.	50
3.6	A section of the speed profile generated under the RBCC strategy overlaid with the thresholds from case study 2.	51
4.1	KL divergence rate between road grade Markov chains and aggregate road grade Markov chains for highway and city driving.	57
4.2	KL divergence rate between traffic speed Markov chains of individual drives and highway aggregate traffic speed Markov chains for off-peak and rush hour traffic times.	58
4.3	KL divergence rate between traffic speed Markov chains of individual drives and city aggregate traffic speed Markov chains for off-peak and rush hour traffic times.	58
4.4	KL divergence rate between traffic speed Markov chains of individual drives and highway aggregate Markov chains for active and passive driving styles.	59
4.5	KL divergence rate between traffic speed Markov chains of individual drives and city aggregate Markov chains for active and passive driving styles.	59
4.6	Example of the stochastic jump process identification procedure.	62
4.7	An example of the data generated jump process for a positive step in grade with an initial grade $\theta = 0\%$	63
4.8	An example of the data generated jump process for a negative step in grade with an initial grade $\theta = 0\%$	64
4.9	The grade data used to generate the stochastic grade prediction example	65
4.10	The real road grade data overlaid with the Markov and jump predictions. Note that neither prediction produces gross deviations from the real data	65
4.11	An example of the data generated stochastic jump process for a down step in grade with an initial grade $\theta = 0\%$	66
4.12	Surface plot of an online learned Markov chain model. Note the similarities between this and Figure 2.2	70
4.13	An online learned stochastic jump process for a positive step change from 0% road grade. Note the similarities between this and Figure 4.7.	70

4.14	The trend of the KL divergence for both the Markov chain and stochastic jump processes throughout the learning cycle. As expected, the KL divergence tends to decrease and approaches a small value near zero—this is indicative of model convergence.	71
4.15	The trend of the mean absolute error between the predicted and actual road grade. Note how the mean absolute error decreases over the learning cycle. . .	71
5.1	Projections of O_∞ , R_∞ , and R_∞^I onto the Δp - $\Delta\phi$ plane for the lateral linearized model. Note that R_∞^I is substantially larger than R_∞ and takes less time to compute.	82
5.2	Projection of O_∞ , R_∞ , and R_∞^I onto the Δu - $\Delta\theta$ plane for the longitudinal linearized model. Note that R_∞^I and R_∞ are similar in size (though R_∞ is slightly larger) while R_∞^I takes less time to compute.	82
5.3	Example of the connectivity condition such that a transition from X_{eq}^n to X_{eq}^{n+1} is allowed.	83
5.4	The time histories of aircraft states as the aircraft transitions through the selected trim points. The transitions between selected trim points occur at time instants designated by vertical dashed lines.	85
5.5	The elevator and thrust control inputs, flight path angle γ , and integrator state trajectories as the aircraft transitions through the selected trim points. Note the vertical jumps in the integrator state (bottom right) plot indicate the integrator state reset. The transitions between selected trim points occur at time instants designated by vertical dashed lines.	86
5.6	Projections of O_∞ and R_∞ onto the $\Delta\beta$ - Δr plane. Note that as H increases, so does R_∞	90
5.7	Projections of O_∞ and R_∞ onto the Δu - $\Delta\theta$ plane. Note that as H increases, so does R_∞	90
5.8	The difference between using a shift register (S R) and Laguerre’s sequence generator (L S) for $H = 1$ and $H = 2$ recoverable sets with the longitudinal dynamics projected onto the Δu - $\Delta\theta$ plane. Note the Laguerre’s sequence generator produces larger set in general than the shift register.	91
5.9	The difference between using a shift register (S R) and Laguerre’s sequence generator (L S) for $H = 1$ and $H = 2$ recoverable sets with the lateral dynamics projected onto the $\Delta\beta$ - $\Delta\phi$ plane. Note the Laguerre’s sequence generator produces larger set in general than the shift register.	92
5.10	Projections of longitudinal recoverable sets with $H = 1$ at different flight conditions.	93
5.11	Projections of lateral recoverable sets for $H = 1$ at different flight conditions.	93
5.12	The effects of additive set-bounded disturbances on $H = 1$ recoverable sets. Note that as the disturbance increases in size, the recoverable set decreases. . .	95
5.13	The effects of additive set-bounded disturbances on $H = 1$ recoverable sets. Note that as the disturbance increases in size, the recoverable set decreases. . .	95

6.1	λ for scalar scaling transformations from the longitudinal trim condition of $U_0 = 800 \frac{\text{ft}}{\text{s}}$, $h_0 = 30000 \text{ ft}$, and $\alpha_0 = 2 \text{ deg}$ to all others in the trim condition grid.	99
6.2	Example of longitudinal $H = 1$, O_∞^{ext} sets scaled from an original $H = 1$, O_∞^{ext} set to an objective $H = 1$, O_∞^{ext} set. Note that scalar scaling in this case is conservative where as the vector scaling is significantly better.	101
6.3	Example of lateral $H = 1$, O_∞^{ext} sets scaled from an original $H = 1$, O_∞^{ext} set to an objective $H = 1$, O_∞^{ext} set. Note that both scalar and vector scaling produce decent approximations in this case. Also note that the vector scaling comes pretty close to replicating the objective set in some examples.	102
6.4	Examples of how the icing affects the $H = 1$, O_∞^{ext} sets in the $\Delta u - \Delta \theta$ plane (left) and the $\Delta u - \Delta \alpha$ plane (right). Note the icing is more predominately seen in the $\Delta u - \Delta \theta$ plane than in the $\Delta u - \Delta \alpha$ plane.	102
6.5	Scaling of the icing scenario $H = 1$, O_∞^{ext} sets	104
6.6	Examples of the accuracy of the sensitivity technique. Top left 2% change, Top right 10%, bottom 50% change. Note for large parameter changes, accuracy is poor.	106
7.1	Controller state and reference governor schematic.	108
7.2	Projection of the lateral O_∞^I , O_∞^I with $x_I(0) = 0$, and O_∞^r sets in the $\Delta \beta - \Delta \phi$ plane with sample state trajectory.	119
7.3	Lateral state trajectories of the sample trajectory in Figure 7.2.	120
7.4	Lateral reference and control time histories of the sample trajectory in Figure 7.2.	120
7.5	Projection of the longitudinal O_∞^I , O_∞^I with $x_I(0) = 0$, and O_∞^r sets in the $\Delta u - \Delta \theta$ plane with sample state trajectory.	121
7.6	Longitudinal state trajectories of the sample trajectory in Figure 7.5.	121
7.7	Longitudinal reference and control time histories of the sample trajectory in Figure 7.5.	122
7.8	The projection of the set O_∞^r on $\Delta N_f - \Delta N_c$ plane with x_I resettable (larger set) and with $x_I = 0$ (smaller set). The variables ΔN_f , ΔN_c denote the deviations from the equilibrium.	123
7.9	The time histories of the set-point \bar{r} for N_f (dash-dotted), governed set-point r and actual N_f	123
7.10	The time history of the state of the integrator.	124
7.11	The time history of the LPC surge margin (solid) and constraint (dashed).	124
7.12	The time history of the HPC surge margin (solid) and constraint (dashed).	125
7.13	The time history of the engine fueling rate (solid) and constraint (dashed).	125

LIST OF TABLES

2.1	Fuel economy improvement broken down between traffic conditions and traffic type.	28
2.2	The five Markov chain models used for model sensitivity analysis	37
3.1	The case studies and thresholds used.	48
3.2	DFCI Results from Vehicle Testing	52
4.1	Time of Day, Type of Driving, and Location	60
5.1	Table of flight conditions used in Figures 5.10 and 5.11.	94
6.1	Examples of optimal scaling parameters λ^*	103
6.2	Examples of optimal scaling parameters $\hat{\lambda}^*$	103

LIST OF APPENDICES

A Auto-Regressive Grade Prediction Technique	131
B Computation of Invariant Sets	133
C Scaled NASA Generic Transport Model	138

LIST OF ABBREVIATIONS

ACC	Adaptive Cruise Control
CACC	Cooperative Adaptive Cruise Control
CSG	Controller State Governor
CSRG	Controller State and Reference Governor
DFCI	Dynamic Fuel Consumption Improvement
FSM	Finite State Machine
KL	Kullback-Liebler
LOC	Loss of Control
PDAS	Percent Difference in Average Speed
PFEI	Percent Fuel Economy Improvement
PnG	Pulse and Glide
RBCC	Rule Based Cruise Control
SDP	Stochastic Dynamic Programming

ABSTRACT

Developments in Stochastic Fuel Efficient Cruise Control and Constrained Control with Applications to Aircraft

by

Kevin K. McDonough

Chair: Ilya V. Kolmanovsky

The dissertation presents contributions to fuel-efficient control of vehicle speed and constrained control with applications to aircraft.

In the first part of this dissertation a stochastic approach to fuel-efficient vehicle speed control is developed. This approach encompasses stochastic modeling of road grade and traffic speed, modeling of fuel consumption through the use of a neural network, and the application of stochastic dynamic programming to generate vehicle speed control policies that are optimized for the trade-off between fuel consumption and travel time. The fuel economy improvements with the proposed policies are quantified through simulations and vehicle experiments. It is shown that the policies lead to the emergence of time-varying vehicle speed patterns that are referred to as time-varying cruise. Through simulations and experiments it is confirmed that these time-varying vehicle speed profiles are more fuel-efficient than driving at a comparable constant speed. Motivated by these results, a simpler implementation strategy that is more appealing for practical implementation is also developed. This strategy relies on a finite state machine and state transition threshold

optimization, and its benefits are quantified through model-based simulations and vehicle experiments. Several additional contributions are made to approaches for stochastic modeling of road grade and vehicle speed that include the use of Kullback-Liebler divergence and divergence rate and a stochastic jump-like model for the behavior of the road grade.

In the second part of the dissertation, contributions to constrained control with applications to aircraft are described. Recoverable sets and integral safe sets of initial states of constrained closed-loop systems are introduced first and computational procedures of such sets based on linear discrete-time models are given. The use of linear discrete-time models is emphasized as they lead to fast computational procedures. Examples of these sets for aircraft longitudinal and lateral aircraft dynamics are reported, and it is shown that these sets can be larger in size

compared to the more commonly used safe sets. An approach to constrained maneuver planning based on chaining recoverable sets or integral safe sets is described and illustrated with a simulation example. To facilitate the application of this maneuver planning approach in aircraft loss of control (LOC) situations when the model is only identified at the current trim condition but when these sets need to be predicted at other flight conditions, the dependence trends of the safe and recoverable sets on aircraft flight conditions are characterized. The scaling procedure to estimate subsets of safe and recoverable sets at one trim condition based on their knowledge at another trim condition is defined. Finally, two control schemes that exploit integral safe sets are proposed. The first scheme, referred to as the controller state governor (CSG), resets the controller state (typically an integrator) to enforce the constraints and enlarge the set of plant states that can be recovered without constraint violation. The second scheme, referred to as the controller state and reference governor (CSRG), combines the controller state governor with the reference governor control architecture and provides the capability of simultaneously modifying the reference command and the controller state to enforce the constraints. Theoretical results that characterize the response properties of both schemes are presented. Examples are re-

ported that illustrate the operation of these schemes on aircraft flight dynamics models and gas turbine engine dynamic models.

CHAPTER 1

Introduction

This dissertation addresses two different topics. The first topic, discussed in Chapters 2, 3, and 4, is stochastic fuel-efficient control of vehicle speed. The second topic, discussed in Chapters 5, 6, and 7, is constrained control with aircraft applications.

1.1 Stochastic fuel-efficient cruise control

In the United States, corporate average fuel economy (CAFE) standards have been mandated to rise substantially by the year 2025 [1]. Because of both this mandate and the world-wide consumer desire to have fuel-efficient vehicles, much research is being pursued in the area automotive fuel-efficiency improvements. The pathways to better fuel economy include, but are not limited to, the use of light-weight materials, new engine architectures [2], more-efficient combustion technologies [3,4], and vehicle speed optimization [5–9]. It is this final pathway, through vehicle speed optimization, which is of main interest in this dissertation. Note that since vehicles are tested on prescribed federal drive cycles, the drive cycle fuel economy may not be improved by changes in vehicle speed profiles. However, according to [10], customer fuel economy improvements, that are possible through vehicle speed optimization, can be credited to an automotive manufacturer depending on the fraction of vehicles utilizing a specific fuel-efficient technology.

1.1.1 Vehicle speed control and optimization

Much of the past work in the area of vehicle speed control has been done on conventional cruise control, adaptive cruise control (ACC), cooperative adaptive cruise control (CACC), platooning, and string stability. As a sample of related literature, in [11], a method of collision avoidance in a platoon of vehicles is presented. References [12–14] examine the problem of string stability in a platoon of vehicles while [14] specifically develops a

decentralized approach for ensuring string stability in a platoon of vehicles. Reference [15] examines the stability and impact of human driven vehicles on ACC. Reference [16] presents an overview of some of these different developments.

More recently, interest has been growing in the optimization of ACC for fuel efficiency. Much of the past research in vehicle speed optimization for fuel efficiency has been done within a deterministic framework. To this end, it is assumed that all necessary information about present and future driving conditions (e.g., traffic conditions, road geometry, driver behavior) is accessible on board the vehicle. References [17–20] implement methods of deterministic dynamic programming in order to produce fuel-efficient vehicle speed profiles. In [18, 19], dynamic programming is used in conjunction with a preview of upcoming road grade to produce fuel-efficient speed profiles for heavy trucks. These references report an average reduction in fuel consumption of 3.5% in vehicle tests while substantially reducing the gear shifts in the heavy trucks. An interesting point in these works is that they used a kinetic-energy approach and aimed at suppressing oscillations in vehicle speed. In [20], a cloud computing approach was proposed to handle the large dynamic programming problem that results when all available information (e.g., traffic data and road geometry, traffic signals) is used. The optimized vehicle speed profile and vehicle trajectory is then relayed to the driver through an advisory application. In real vehicle testing, fuel consumption reductions of 5-15% have been reported with this approach.

In references [21–27], model predictive control techniques have been considered which use previews of upcoming road geometry and traffic information in order to reduce fuel consumption. In [21], a fuel consumption reduction of 11.36% in simulation is reported through consideration of road grade only. In [22], the model predictive control approach is used to reduce fuel consumption while also minimizing tracking error in a vehicle-following scenario. In that work, the authors report a reduction in fuel consumption of 2.2% in highway driving simulations and 5.9% in city driving simulations.

References [23–26] use vehicle-to-vehicle (V2V) communication in conjunction with model predictive control to improve fuel economy. In [23], the host vehicle (i.e., the vehicle running the model predictive control scheme) follows a simulated vehicle with a specifically programmed standard speed profile (in this case, it is the FTP75 speed profile) while the simulated lead vehicle provides a preview of its speed and the road geometry to the host vehicle via V2V communication. Fuel economy improvements in the range of 16-20% for the host vehicle are reported over the fuel economy of the preceding vehicle in both simulation and engine dynamometer testing (but have yet to be confirmed through vehicle testing). In [24–26], V2V communication relays speed information to the host vehicle running the model predictive control scheme; however, the host vehicle is given the

option of altering its following distance. Depending on the allowable distance margin and the length of the preview, the authors of [24–26] report fuel economy increases between 10-20% in simulation and dynamometer bench testing.

Another deterministic approach to vehicle speed optimization exploits a periodically varying vehicle speed pattern often referred to as *pulse and glide*— (PnG) or periodic cruise. In 1976, Elmer Gilbert, by applying results in periodic optimal control theory to a vehicle model, suggested that such periodic cruise profiles would be fuel-efficient, see [28]. In his work, Gilbert develops a rationale for periodic cruise based on decomposing the problem into four sub-problems and demonstrating that under the appropriate assumptions the periodic cruise speed profile is more efficient than a constant speed profile.

More recent works [29–31] also discuss the fuel-saving benefits of periodic vehicle speed profiles, where such profiles are determined through physics based approaches. The engine map and dynamics of the vehicle are considered to find suitable operating conditions and establish a periodic speed profile between them, while supervisory logic is used to maintain and handle disturbances to the periodic profile. Based on the simulations, the projected fuel-economy benefits of periodic speed profiles range from 33% to 77% as compared to driving at a constant speed equal to the average speed of the periodic speed profile. PnG speed profiles have also been discussed in the hyper-mileage community [32].

Stochastic approaches to the optimization of vehicle speed have also been proposed in [8, 9, 27]. In [27], a stochastic model predictive control scheme is considered in which the driver inputs are modeled by a Markov chain and a receding horizon optimal control problem is solved online based on model approximation and quadratic programming, which is a computationally expensive approach. In [8], stochastic drift counteractive optimal control (SDCOC) is applied to adaptive cruise control and hybrid vehicle control. In [9], the road grade and traffic speed are treated stochastically and a speed profile is stochastically optimized to reduce fuel consumption. This preliminary work provided the motivation for the developments in this dissertation. Note also that much work has been done on stochastic dynamic programming for hybrid electric vehicle energy management and gear shift optimization [33–36]. In these applications, wheel power demand is treated as stochastic and modeled by a Markov chain model; however, vehicle speed optimization is not considered.

In this dissertation, road grade and traffic speeds are modeled stochastically, as in [8, 9, 37–39] (see [40, 41] and references therein for stochastic modeling techniques and applications). The treatment of traffic speed as a stochastic variable is natural because local traffic behavior near a vehicle’s location cannot be deterministically predicted (e.g., a vehicle may be stuck behind a slow moving truck). The treatment of road grade as a stochastic variable deserves an explanation. If a route is known, road grade can be predicted by pre-

view systems such as electronic horizon or V2V communication. However, implementation of these preview systems entails additional cost and reliability issues to which the automotive industry is especially sensitive. Our approach to modeling road grade stochastically avoids the need for preview systems and allows the use of stochastic dynamic programming to generate control policies that do not require on-board optimization (and hence are computationally easier to implement and to certify). Another motivation for the stochastic approach is that the control policies can be generated specifically for an individual driver's typical commute.

Once the stochastic models are generated for road grade and traffic speed, in the form of Markov chain models, a cost function that accounts for expected fuel consumption, expected travel time, and relative distance to traffic is considered and used within the stochastic dynamic programming (SDP) framework [42] to produce fuel-efficient cruise control policies.

Through the development and subsequent testing of these policies, an interesting behavior emerged. When a simulated host vehicle implementing an SDP policy followed another vehicle traveling at a relatively constant speed, the policy would direct the host vehicle to oscillate its speed around the lead vehicle's speed. This oscillating behavior is similar to the PnG behaviors discussed in [29–31]. An important distinction between our work and that described in [29–31] is that in our case, PnG type behaviors emerged as a result of applying SDP policies and were not initially assumed to be fuel-efficient.

Simulation data suggest that PnG type speed profiles were more efficient than driving at a constant speed with our results indicating a potential improvement up to 18% depending on the host vehicle speed [43, 44]. SDP control policies were then developed for a test vehicle and experimentally tested. The results of the vehicle tests indicated that the time-varying PnG strategy is more efficient than driving at a constant speed with an average increase in fuel economy of 4.51% (a maximum of 11.58%) while observing an average reduction in average speed within the 2 mph range, [45]. This translates to an average fuel economy improvement of 2.74% if the fuel economy increase due to speed reduction is accounted for.

A sensitivity analysis of the fuel-economy benefits of the SDP policies to changes in cost function parameters and Markov chain model grid selection has been completed more recently and its results are also included in this dissertation. The results suggest that the policies are not highly sensitive to changes in either the cost function parameters or Markov chain model grid selection; however, better choices of cost function parameters may exist than those which were originally selected. A more exhaustive sensitivity analysis is left to future work when more data can be collected.

Because the generation of SDP policies can be time consuming and the tuning of the policies can be unintuitive for a typical calibrator in the automotive industry, an alternative control methodology, referred to as rule based cruise control (RBCC), was developed for the practical implementation of the PnG behavior, [46]. The RBCC scheme utilizes a finite state machine that controls transitions between different phases of the PnG speed profile (i.e., normal, glide, recover, and bleed) through the use of thresholds and logic applied to vehicle speed and average predicted road grade. A similar structure is developed and tested within a simulation environment in [31]. However, in [31] the algorithm is different; a grade preview is required and only simulation results are presented. In our work, we optimize the transition thresholds through Monte-Carlo simulations and present both simulation and vehicle testing results.

1.1.2 Stochastic modeling of road grade and traffic speed

The first part of this dissertation also describes developments made in the stochastic modeling of road grade and traffic speed. These include the similarity assessment between Markov chain models of road grade and traffic speed using the Kullback-Liebler (KL) divergence and divergence rate and a novel approach to modeling road grade using a stochastic jump process.

Markov chain model similarity assessments are made between models learned for different times of day, driving types, and locations. It is shown that the KL divergence rate can be used to successfully categorize Markov chain models of road grade based on the location of the collected data (i.e., city versus highway road geometries) and Markov chain models of vehicle speed based on the style of the driver (i.e., passive or active).

Similarity assessment can also be applied to the online learning of Markov chain models. Using techniques presented in [38], Markov chain models of road grade have been successfully learned in real-time on board a vehicle [47], while it was demonstrated in simulation that, with the appropriate sensors, Markov chain models for traffic speed can also be learned. To decide when a model has been sufficiently learned (i.e., it does not substantially change over a given time or distance segment), the KL divergence is used to measure the similarity of the currently learned Markov chain model to a model from the current learning cycle at a given number of steps in the past. When the KL divergence between these two models is sufficiently small, this indicates that the model learning may be terminated.

The fuel-efficient vehicle speed optimization described above is performed off-line. Ideally, a set of Markov chain models would be developed that represent a large range

of driving conditions and their corresponding SDP policies would be stored on board for vehicle use. To select the appropriate policy for a given region, the statistics of the given region must be compared to those used to generate the stored policies. Given the ability to learn Markov chain models online, the learned models can be compared to those stored in a database and the most similar Markov chain model and its corresponding policy can be selected and used.

Because transition probability matrices of Markov chain models for road grade and traffic speed are nearly diagonal [48], indicating that the prediction will likely yield the current value as the next value, a novel approach to modeling road grade is introduced in this dissertation. This novel approach models road grade as a stochastic jump process. This process predicts the distance over which a positive or negative step change in road grade is likely to occur. It is shown that for prediction horizons less than 20 meters, this stochastic jump process produces predictions that are more accurate than the Markov chain method. Online learning techniques discussed in [38, 47] are then modified and applied to the online learning of the stochastic jump process. This learning algorithm is illustrated by simulations.

1.2 Constrained control with applications to aircraft

The second part of this dissertation covers developments and advances in constrained control motivated by the need to develop control schemes which handle aircraft loss of control (LOC) situations [49, 50]. In particular, our interest is in constraint handling aspects of such problems.

The loss of control is an issue that causes many aviation accidents; however, separating LOC events from other factors can be difficult, [49, 51–53]. In a highly cited work, Wilborn [54] defines an LOC event statistically (based on a large study of aviation accidents jointly performed by NASA and the Boeing Company [55]) as an excursion from three or more critical envelopes: the adverse aerodynamics envelope, the unusual attitude envelope, the structural integrity envelope, the dynamic pitch control envelope, and the dynamic roll envelope. These envelopes are essentially a collection of aerodynamic, structural, and control constraints. As more constraints are violated, it becomes more difficult to recover the aircraft to safe operating conditions. In a separate way, albeit related, references [53, 56, 57] define LOC events from a strictly control perspective using analytical control techniques. In all cases, though, violation of aerodynamic, structural, and control constraints is agreed to be a major cause of LOC events.

The work in the second part of this dissertation will address two main topics. The first

topic centers around the use, calculation, and estimation of safe, recoverable, and integral safe sets. These sets are motivated by the aircraft LOC prevention considerations and are developed specifically with aircraft applications in mind. In particular, a constraint-admissible aircraft maneuver planning procedure that exploits these sets is presented and illustrated with an example. Scaling transformations are then developed in order to facilitate the approximation of these sets at different flight conditions when the aircraft dynamics have changed (e.g., due to aircraft wing icing or actuator malfunction).

The second topic focuses on the development, theory, and application of two novel control schemes for constrained systems: the controller state governor (CSG) and the controller state and reference governor (CSRG). The CSG was developed as an add-on to a nominal aircraft control scheme using integral action. The CSRG subsequently evolved from the CSG by combining it with the reference governor [58–62].

1.2.1 Safe, recoverable, and integral safe sets

The idea of a safe set is not unique to our work or to the definition used in our work. We define safe sets as sets of initial conditions for linear, discrete-time, closed-loop systems from which the ensuing closed-loop trajectories are constraint-admissible. That is, an initial state within a safe set will yield a closed-loop trajectory, using a given nominal controller, that satisfies constraints. The term *safe set* is also used in [53,56,57,63–70] but its meaning is different as it describes a set of reachable states without constraint violation given an initial condition of the nonlinear aircraft dynamics. References [63–65] develop methods that combine hybrid control and set theory to produce these safe sets, which are subsets of the aircraft’s complete dynamic (accelerating) and trim (steady-state) flight envelope.

In [53,56,57], the notion of a safe set as a set of constraint-admissible reachable states is approached from a rigorous control standpoint. To this end, developments are presented to handle cases such as stall [56] and actuator impairment [53,57]. In [68,70], the idea of a maneuvering envelope is introduced. This maneuvering envelope is similar to a safe set as it characterizes a set of safe conditions in which to operate. It is different, though, because it is the intersection of constraint-admissible forward and backward reachable sets over a given time horizon from an initial condition. In all cases, these sets are calculated using partial differential equations (PDEs) and require significant computational resources. Thus, this approach may require the anticipation of malfunctions for the aircraft and may not present an easily adaptable method to be implemented in real time on board an aircraft.

To tackle this problem with computational tractability, NASA Ames researchers have taken the approach of decomposing the nonlinear aircraft dynamics into fast and slow dy-

namics. They then used an aerodynamic model that is linear in identification parameters [71–73], and implemented a probability based model identification algorithm [74] to produce a tractable problem for calculating a maneuvering envelope that can be solved on board the aircraft. By decomposing the aircraft dynamics, and essentially neglecting the fast dynamics, a maneuvering envelope (like that described in [70]) over a relatively short time horizon can be quickly computed. Also, through the implementation of the model identification algorithm, the maneuvering envelope can be updated online as changes to the model occur due to changes in aerodynamics. This approach has undergone promising testing within a flight simulator with flight crews.

Our work in the area of computing safe sets for LOC prevention [75–77] has focused on the use of linearized aircraft models at given trim conditions. Besides simplifying the on-board computations, the reliance on the discrete-time linearized aircraft models is motivated by the consideration that in an LOC scenario it is unlikely that large alterations in the aircraft’s flight conditions will be attempted. Instead it is more likely that a progression of several small changes/transitions through trim state space will be executed. Furthermore, if the constraints are affine and the system model is linear discrete-time, the safe sets computed are polyhedral and are easily storable and manipulable [58, 59, 78, 79].

The safe sets are defined in this work as sets of initial states from which ensuing closed-loop trajectories of the system satisfy constraints for all future times. Many papers exist discussing the properties of these kinds of sets. See, for example, the book [80] and papers [58, 78, 81–84].

We extend the notion of safe sets to that of recoverable sets and integral safe sets. Recoverable sets are defined for the nominal closed-loop system extended with an additional auxiliary subsystem, the output of which modifies the control signal. A plant state is recoverable if there exists a state of the auxiliary subsystem such that the pair of plant and auxiliary states is constraint-admissible; i.e., the subsequent response satisfies the constraints. An integral safe set is a set of all plant states for which there exists a resettable nominal feedback controller state such that the pair is constraint-admissible; i.e., the subsequent closed-loop response satisfies the constraints. The primary advantage of this approach is that the computation of recoverable and integral safe sets reduces to that of safe sets in a larger dimensional space, moreover, the computational complexity can be adjusted by varying the order of the auxiliary subsystem or retaining only certain controller states for reset. Thus, the computations of these sets may be feasible on board an aircraft in LOC situations based on identified linear models.

After defining safe, recoverable, and integral safe sets, we demonstrate how these sets can be used to plan constraint-admissible trajectories through the trim envelope. For in-

stance, if the current aircraft state is in a safe set of the next desired trim condition, then a constraint-admissible trajectory exists for the aircraft between these two trim states. This approach is similar to [60, 85], which considers spacecraft applications. The recoverable and integral safe sets can be used similarly to safe sets for maneuver planning as we will subsequently demonstrate.

We also develop methods by which safe and recoverable sets can be scaled in order to approximate safe and recoverable sets at unexplored trim conditions. This is useful if aerodynamic changes are occurring (e.g., wing icing), thus causing the linearized models to change. By implementing a model identification technique, (we implemented retrospective cost model refinement (RCMR) in [76]) the local linear model can be identified. Then, through scaling transformations calculated off-line for the nominal system, scaled subsets of safe or recoverable sets at unexplored trim conditions of the altered aircraft dynamics can be estimated [75].

A related area to LOC research is fault detection and fault tolerant control, see [86]. References [87–90] present recent developments in this area. In [89], approaches are presented that take advantage of decomposed fast and slow dynamics to produce new control schemes online that are tolerant to various faults. References [87, 88, 90] all use different methods to predict changes to the aircraft flight envelope given information on changes in the system. In [76], we explore a fault signature detection technique in which pertinent information about aircraft stability derivatives is gleaned from the identified linear model and used to estimate changes in global aerodynamic parameters.

While much of the work described in the second part of this dissertation is strongly motivated by the aircraft LOC problem, it is applicable to constrained systems more generally.

1.2.2 The controller state governor and the controller state and reference governor

Two novel control schemes for systems with constraints, the controller state governor (CSG) and the controller state and reference governor (CSRG), are introduced, and relevant analyses of stability and convergence are presented in Chapter 7 of this dissertation. These control schemes take advantage of resettable controller states to generate constraint-admissible trajectories, reduce closed-loop system energy, and improve performance.

The CSG and CSRG are motivated by reference governors. The reference governor is an add-on control scheme for enforcing pointwise-in-time state and control constraints in well-designed linear and nonlinear closed-loop systems, see [58–62] and references therein. Reference governors accomplish this by modifying the reference commands to the nominal

closed-loop system whenever the danger of constraint violation occurs. The closed-loop system states and the modified reference are maintained in the maximum output admissible set (i.e., in the safe set) or its subset. The projection of this maximum output admissible set gives the constraint-admissible domain of attraction, that is the set of all closed-loop system states that are recoverable without constraint violation.

In systems that employ dynamic controllers, an additional opportunity exist which is to reset the controller states. Dynamic compensators are broadly used, in particular, to provide integral control action [91]. Resetting dynamic controller states has been previously exploited to improve performance in nonlinear control, see [92–94], though resetting dynamic controller states does not appear to have been broadly used for enforcing constraints. Notable exceptions are [58] where the state of the auxiliary reference generating subsystem (rather than nominal controller) is reset and [95] where the state is decomposed for the purpose of improving actuator coordination. The main benefits of resetting the controller states are enlarging the constraint-admissible domain of attraction and improving performance. The CSG and CSRG were first introduced in our publications [77,96].

1.3 Major contributions and relevant publications

1.3.1 Major contributions

The major contributions of this dissertation are summarized as follows.

Fuel-efficient cruise control

1. The development and testing, both in simulation and in vehicle, of stochastic dynamic programming (SDP) policies for vehicle speed control that accounts for statistical patterns in road grade and traffic speed. While SDP has been applied to other automotive problems in the past, the use of SDP to generate vehicle speed control policies for cruise control and quantifying their benefits in realistic simulations and experimental testing is novel. As these policies do not require preview nor on-board optimization, they are attractive to practitioners. In vehicle-following scenarios, they also lead to time-varying vehicle speed oscillations, that we refer to as time-varying cruise or pulse and glide (PnG). PnG strategies are not in themselves novel, however, demonstrating the conclusion that they can emerge from the application of SDP policies in novel.
2. The demonstration both in simulation and in vehicle that time-varying speed profiles, when adequately controlled, can be more fuel-efficient than driving at a constant

speed with the same average value as that of the time-varying speed profile.

3. The development of novel categorizers of Markov chain models for road grade and traffic speed based on the KL divergence rate. Also, the development of a convergence metric for the online learning of Markov chain models for road grade and traffic speed based on the KL divergence.
4. The development of a stochastic jump process prediction method for step changes in road grade.

Constrained control with aircraft applications

1. The development of recoverable and integral safe sets based on discrete-time linear models that can be computed quickly and used in constraint-admissible aircraft maneuver planning.
2. The scaling procedure of safe, recoverable, and integral safe sets between different flight conditions to facilitate constraint-admissible aircraft maneuver planning in LOC situations.
3. The development of the CSG and CSRG schemes for constrained control, including stability and finite-time convergence proofs.

1.3.2 Relevant publications

The following is a list of relevant publications on the different topics addressed in this dissertation.

Fuel-efficient cruise control

1. McDonough, K., D'Amato, A., Mullen, J., Petersen, C., Kolmanovsky, I., & Filev, D. Transition Threshold Optimization for a Rule Based Automotive Cruise Control. *In Decision and Control (CDC), Conference on*, 2014.
2. McDonough, K., Kolmanovsky, I., Filev, D., Szwabowski, S., Yanakiev, D., & Micheli, J. Stochastic Fuel Efficient Optimal Control of Vehicle Speed. *Optimization and Optimal Control in Automotive Systems* (pp. 147-162). Springer International Publishing, 2014.
3. McDonough, K., Kolmanovsky, I., Filev, D., Yanakiev, D., Szwabowski, S., & Micheli, J. Stochastic dynamic programming control policies for fuel efficient vehicle following. *In American Control Conference (ACC)*, 2013 (pp. 1350-1355).

4. McDonough, K., Kolmanovsky, I., Filev, D., Yanakiev, D., Szwabowski, S., & Micheli-
lini, J. Stochastic dynamic programming control policies for fuel efficient in-traffic
driving. *In American Control Conference (ACC)*, 2012 (pp. 3986-3991).
5. Kolmanovsky, I., McDonough, K., & Gusikhin, O. Estimation of fuel flow for telematics-
enabled adaptive fuel and time efficient vehicle routing. *In ITS Telecommunications
(ITST), 2011 11th International Conference on*, 2011 (pp. 139-144).

United States Patent

1. Filev, D. P., Kolmanovsky, I., McDonough, K., Szwabowski, S. J., Michelini, J. O.,
Yanakiev, D., & Abou-Nasr, M. "On-board real-time speed control setpoint variation
using stochastic optimization." U.S. Patent 8,930,116, issued January 6, 2015.

Stochastic modeling of road grade and traffic speed

1. Hoekstra, A., Filev, D., Szwabowski, S., McDonough, K., & Kolmanovsky, I. Evolv-
ing Markov chain models of driving conditions using onboard learning. *In Cybernet-
ics (CYBCONF), 2013 IEEE International Conference on* (pp. 1-6).
2. McDonough, K., Kolmanovsky, I., Filev, D., Yanakiev, D., Szwabowski, S., Miche-
lini, J., & Abou-Nasr, M. Modeling of vehicle driving conditions using transition
probability models. *In Control Applications (CCA), 2011 IEEE International Con-
ference on* (pp. 544-549).

Constrained control with aircraft applications

1. McDonough, K., & Kolmanovsky, I. Controller state and reference governors for
discrete-time linear systems with pointwise-in-time state and control constraints.
American Control Conference (ACC), 2015. (accepted).
2. McDonough, K. & Kolmanovsky, I. Integrator resetting for enforcing constraints in
aircraft flight control systems. *AIAA Sci-Tech Conference, AIAA 2015-1995*, 2015.
3. Poloni, T., Kalabic, U., McDonough, K., & Kolmanovsky, I. Disturbance Canceling
Control based on Simple Input Observers with Constraint Enforcement for Aerospace
Applications. *In Control Applications (CCA), 2014 IEEE Conference on* (pp. 158-
165).
4. Yu, M. J., McDonough, K., Bernstein, D. S., & Kolmanovsky, I. Retrospective cost
model refinement for aircraft fault signature detection. *In American Control Confer-
ence (ACC)*, 2014 (pp. 2486-2491).

5. McDonough, K, Kolmanovsky, I., & Atkins, E. Recoverable sets of initial conditions and their use for aircraft flight planning after a loss of control event. *AIAA Sci-Tech Conference*, AIAA 2014-0786, 2014.

1.4 Dissertation outline

This dissertation is divided into two parts. The first part encompasses Chapters 2, 3, and 4, and develops advancements in the area of automotive fuel-efficient cruise control. The second part encompasses Chapters 5, 6, and 7, and develops advances in the area of constrained control motivated by aircraft applications.

Chapter 2 develops an SDP approach to the optimization of vehicle speed and includes results from both simulations and in-vehicle testing. The longitudinal vehicle dynamics model is presented in Section 2.1 with an explanation that the control input in our version of fuel-efficient cruise control alters the set-point of the vehicle's nominal cruise control. Section 2.2 develops the Markov chain modeling techniques for the stochastic treatment of the operating environment (i.e., the road grade and traffic speeds). Section 2.3 develops the cost-function used in the stochastic optimization and defines each of its ingredients. The SDP methodology used to generate the on-average optimal control policies is presented in Section 2.4. Simulation and vehicle testing results are presented in Sections 2.5 and 2.6, respectively. Finally, this chapter ends with a control policy sensitivity analysis with respect to cost-function weights and Markov chain model grid selection in Section 2.7.

Chapter 3 presents the rule based cruise control (RBCC) architecture, the threshold optimization method, and simulation and vehicle testing results. Section 3.1 provides the problem formulation on which the RBCC scheme is based. Section 3.2 develops the structure and defines each of the four states used in the RBCC finite state machine. Section 3.3 develops the Monte-Carlo simulation based optimization procedure used to determine RBCC transition thresholds. Finally, this chapter ends with a presentation of the simulated and vehicle testing results in Section 3.4.

Chapter 4 describes the developments in stochastic modeling and stochastic model categorization. Section 4.1 develops a method of using the Kullback-Liebler (KL) divergence and divergence rate to categorize stochastic models based on important attributes such as location of model data (i.e., city or highway), driving time (i.e., rush-hour or off peak), and driving style (i.e., passive or active). A stochastic jump process approach to modeling the road grade is presented in Section 4.2. A comparison of the Markov chain model and stochastic jump model grade prediction methods is presented in Section 4.3. This chapter ends with a presentation of online learning techniques developed for Markov chain models

and stochastic jump models in Section 4.4.

The second part of this dissertation is organized as follows. Chapter 5 introduces the safe, recoverable, and integral safe sets in Sections 5.1, 5.2, and 5.3, respectively. This is followed by a comparison of these sets in Section 5.4 based on an aircraft model. This chapter also includes a description of a maneuver planning technique that uses safe, recoverable, or integral safe sets in Section 5.5 and provides a simple example presented that utilizes integral safe sets to that end. This chapter ends with Section 5.6, which discusses safe and recoverable set dependence on flight condition, auxiliary system selection for the generation of recovery sequences, recovery sequence horizon, and the treatment of additive set-bounded disturbances.

Chapter 6 introduces the scaling transformations used to estimate safe and recoverable sets at unexplored trim conditions. Two scaling transformations, scalar and vector, are considered in Sections 6.1 and 6.2 with numerical results presented in Section 6.3 with two examples: scalar scaling transformations over a grid of trim conditions and an aircraft icing example. Finally, this Chapter ends with the presentation of another safe set estimation technique based on set sensitivity to changes in aerodynamic parameters in Section 6.4.

Chapter 7 introduces, defines, and derives theoretical properties of the CSG and CSRG. Two examples are considered: an aircraft flight control example and a gas turbine engine control example. The CSG and analysis of stability under dynamic controller state reset is presented in Section 7.1. The CSRG and an analysis of finite-time convergence is presented in Section 7.2. Remarks discussing relevant connections between the CSRG and LQ-I control as well as a brief introduction into potential extensions of the CSRG to nonlinear systems are found in Section 7.3. Finally, Chapter 7 ends with a constrained aircraft flight example in Section 7.4 and a constrained gas turbine engine example in Section 7.5.

Concluding remarks and future work discussion end this dissertation in Chapter 8.

CHAPTER 2

Modeling, Optimization, and Results for Fuel-Efficient Cruise Control

This chapter addresses the development of stochastic dynamic programming (SDP) based vehicle speed control policies. The models used for SDP control policy generation include the longitudinal vehicle dynamics model, the stochastic models of road grade and traffic speed, and the vehicle fuel consumption model.

The models are designed to predict the vehicle states at the next spatial step given current vehicle states. Thus, the models are discrete models, but in a spatial sense and not in a temporal sense. The reason for spatially modeling the states as opposed to temporally modeling the states is that we choose to treat traffic conditions and the driving environment as being spatial and not temporal. For instance, the road grade changes spatially over a drive. Also, traffic speeds may be treated as spatially distributed, for instance, certain areas (such as those near highway exit and entrance ramps) consistently move slower than other areas. Thus, given a distance segment of length Δs , and values of road grade, traffic (reference) speed, host vehicle speed, and relative distance between vehicles at the beginning of the current segment, the models that we employ produce the values of these variables at the start of the next distance segment.

The desired control policies provide the next set-point for the vehicle's nominal speed controller based on the current state of the system, given by current vehicle speed, current traffic speed, current road grade, and relative distance to the preceding traffic vehicle, see Figure 2.1. Note that no deterministic preview of future grade or driving conditions is required.

This Chapter is organized as follows. After the, longitudinal vehicle dynamics model is addressed in Section 2.1 and the stochastic modeling of road grade and traffic speed is addressed in Section 2.2, the cost function, and its ingredients, used for the optimization is presented in Section 2.3. Next, the SDP methodology is discussed in Section 2.4.

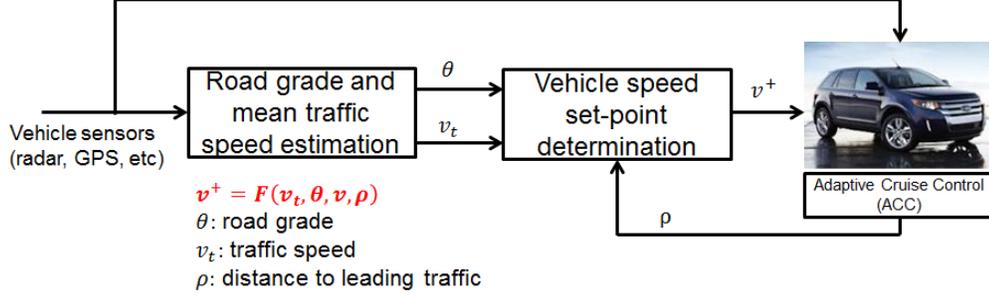


Figure 2.1: A schematic of the fuel-efficient speed control architecture.

Section 2.5 discusses the simulation results. These results are broken into two testing scenarios: *in-traffic* in which traffic speeds change and *vehicle-following* where the traffic speed is assumed to be constant. Next follows the vehicle experimentation results for the vehicle-following case in Section 2.6. Finally a sensitivity analysis of the control policies is presented in Section 2.7.

2.1 Longitudinal vehicle dynamics

A simplified model for longitudinal vehicle dynamics has been adopted for control policy generation. The controlled vehicle is referred to as the host vehicle, and its velocity is denoted by v . The reference speed, or traffic speed, denoted as v_t , represents either the local traffic speed or the speed of the preceding (lead) vehicle depending on the scenario being considered. Finally, the relative distance between the vehicles is denoted by ρ . The traffic speed, host vehicle speed, and relative distance values one distance segment ahead are denoted by v_t^+ , v^+ , and ρ^+ .

The host vehicle speed is updated according to one of the following dynamic equations,

$$v^+ = v_t + u, \quad (2.1)$$

or

$$v^+ = v + u, \quad (2.2)$$

where u is a control input that is an offset from either the current traffic speed, v_t , or the current host vehicle speed, v . In the initial simulations, (2.1) was used. However, for the vehicle experiments, (2.1) was replaced with (2.2) as it improved drivability and passenger comfort.

An approximate model for relative distance dynamics is defined by

$$\rho^+ = \rho + ((v^+ - v) + (v_t^+ - v_t))\Delta T, \quad (2.3)$$

where

$$\Delta T = \frac{2\Delta s}{v_t^+ + v_t}, \quad (2.4)$$

is (approximately) the time to travel the road segment of length Δs . A distance segment length of $\Delta s = 30$ m was used to generate control policies in the simulation case studies. To provide faster control updates, Δs was reduced to 15 m when generating control policies for the vehicle experiments.

In both the experimental vehicle testing and the computer simulations, a set-point speed, v^+ , was provided to existing speed controllers. These existing controllers were relied upon to execute the necessary changes in speed.

2.2 Stochastic models of traffic speed and road grade

Transitions in traffic speed and road grade are modeled using Markov chains. Stochastic modeling of driving conditions has been previously employed in hybrid electric powertrain management, [35], and for our preliminary work in [9]. Transition probabilities are defined for changes in traffic speed, v_t , and road grade, θ , over a road segment of length Δs ,

$$\mathbb{P}(v_t^+ | v_t), \quad \mathbb{P}(\theta^+ | \theta), \quad (2.5)$$

with values of v_t and θ quantized to values on a discrete grid, $v_t \in \{0, 1, \dots, 36\}$ m/s, $\theta \in \{-6, -5, \dots, 5, 6\}$ %.

The identification of the transition probabilities has been performed from sixteen data sets collected on board the same vehicle used for the vehicle experiments. The vehicle has been driven along a route on and around M-39 (mixed highway and city driving), in northbound and southbound directions, during different times of day (in rush hour traffic and off-peak hours traffic) and with the driver emulating two different driving styles (passive and active). Transition probability models have been identified using the relation,

$$\mathbb{P}(x_j | x_i) \approx \frac{N_{x_i, x_j}}{M_{x_i}}, \quad (2.6)$$

where (2.6) is the approximate transition probability of a variable $x \in \{v_t, \theta\}$ from a discrete state x_i to state x_j , N_{x_i, x_j} is the number of transitions from state x_i to state x_j observed

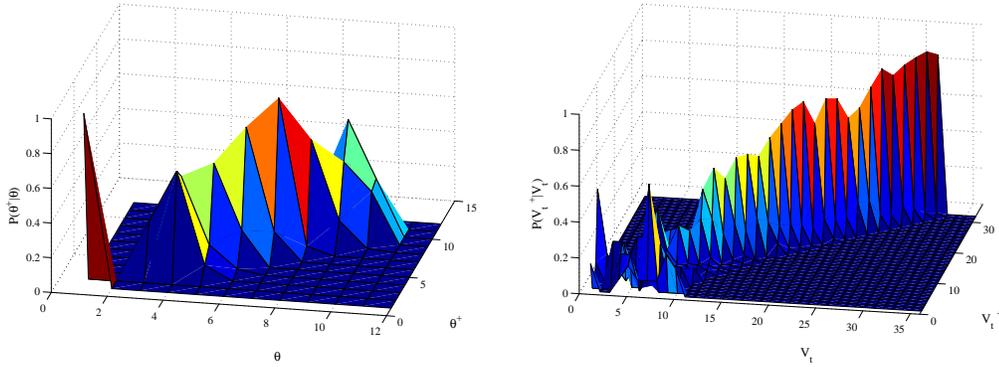


Figure 2.2: A surface plot of road grade (left) and traffic vehicle speed (right) transition probabilities.

in the data, and M_{x_i} is the total number of transitions out of state x_i . Figure 2.2 illustrates the transition probability matrices learned based on one of the data sets. Note the near diagonal character of the model suggesting that continuing with the same speed and grade, i.e., $v_t^+ = v_t$ and $\theta^+ = \theta$, is most likely. On the right side of Figure 2.2, one can observe more off-diagonal entries at lower traffic speeds as it takes a longer time for a vehicle to travel the distance segment and thus more time is available to accelerate to different speeds.

Assuming that θ and v_t are independent, it follows that

$$\mathbb{P}(\theta^+, v_t^+ | \theta, v_t) = \mathbb{P}(v_t^+ | v_t) \mathbb{P}(\theta^+ | \theta). \quad (2.7)$$

Remark: For the modeling of traffic speed as a stochastic variable, a multi-model approach where several transition probability models (TPMs) in the form of Markov chains and SDP policies are developed for different traffic types (i.e., rush hour traffic versus off-peak hour traffic) and road types may be used. The Kullback-Liebler (KL) divergence can be used to establish similarity between currently observed transition probabilities and transition probability models that have been stored and for which the corresponding SDP policy is available. The SDP policy for the best matching TPM is then switched to for the on-board use. References [47, 48], as well as Chapter 4, discuss an evolving models approach where multiple models are generated to cover the drives of interest.

In the development of the SDP control policies in this chapter, the Markov chain models are exclusively used. However, Chapter 4 discusses another approach to stochastically modeling road grade.

For the in-traffic simulation results presented in Section 2.5.1, five Markov chain models were used. One model was used for the prediction of road grade, θ , and this model was

generated using concatenated data from all of the road grade data from the M-39 drives. Four different vehicle speed Markov chain models were developed for the prediction of traffic speed. Traffic speed data was collected during different times of the day (rush hour and off-peak time) and using two different driving styles (passive and active) during both times of the day, this information is presented in Table 4.1. Thus, Markov chain models were generated from concatenated data for passive driving during rush hour, passive driving during off-peak time, active driving during rush hour, and active driving during off-peak time. An SDP policy for each of these four combinations of road grade (one model) and traffic speed (four models) Markov chain models was generated. When the traffic analog speed profile was selected, the appropriate traffic speed Markov chain model was also selected, see Table 4.1. For instance, if the traffic analog were traveling according to drive 6, the active driving during rush hour Markov chain model was used.

2.3 Cost Function and its ingredients

The SDP problem formulation uses an incremental cost function of the general form,

$$R = \bar{W}_f + \lambda \bar{T}_t + \phi(\rho), \quad (2.8)$$

where the ingredients are the expected fuel consumption, \bar{W}_f , over the road segment of length Δs , the expected segment travel time, \bar{T}_t , and the relative distance constraint violation penalty, $\phi(\rho)$.

The ingredients of the cost function are now discussed.

2.3.1 Fuel consumption

The general model for the fuel consumption is of the form,

$$W_f = W_f(v, v^+, \theta, \theta^+), \quad (2.9)$$

where v and θ are the vehicle speed and the road grade, respectively, at the beginning of the current road segment of length Δs , while v^+ and θ^+ are the vehicle speed and road grade, respectively, at the beginning of the next road segment.

For the simulation case studies, a model of (2.9) was developed based on a single hidden layer neural network and has the form,

$$W_f = \sigma_2(w_2 \sigma_1(w_1 u_{nn} + b_1) + b_2), \quad (2.10)$$

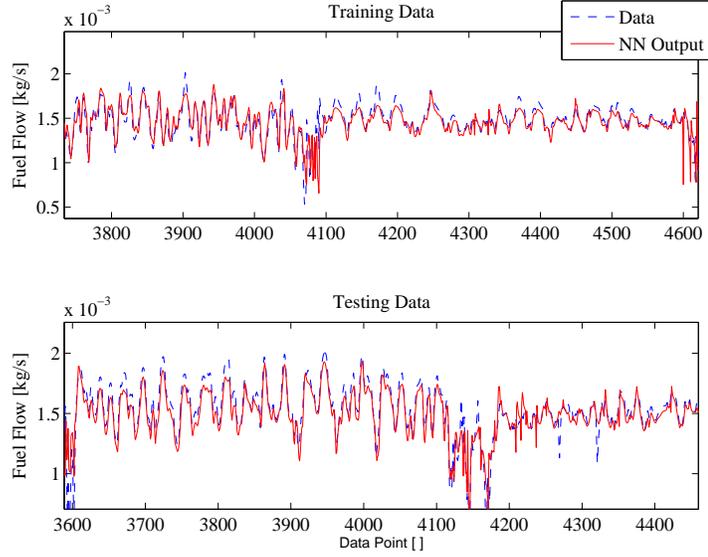


Figure 2.3: Example of the training and testing data for the neural network model of the fuel flow based on CarSim data. Note only a small sample of the 7,000 data points was selected in each plot for clarity.

where σ_1 and σ_2 are the hyperbolic and linear activation functions, respectively, w_1 and w_2 are the corresponding vectors of weights, b_1 and b_2 are the corresponding vectors of biases, and $u_{nn} = (v, v^+, v^+ - v, \theta, \theta^+)^T$ is the model input vector. The neural network was trained using Bayesian regularization back propagation (on the MATLAB neural network toolbox, [97]) applied to data from the sixteen vehicle drives replicated in the CarSim simulation environment, [98]. Roughly 14,000 data points were used to generate the neural network (7,000 to train and 7,000 to test). See Figure 2.3 which demonstrates suitable accuracy is achieved through this neural network approach.

For the experimental vehicle testing, a physics-based fuel consumption model representative of the 2007 Ford Edge experimental vehicle was provided by the Ford Motor Company. Note that both models account for gear shifts and torque converter lock/unlock events.

The expected value of fuel consumption used in the incremental cost function (2.8) has the form

$$\bar{W}_f = \mathbb{E}_{v^+, \theta^+} [W_f(v, v^+, \theta, \theta^+)]. \quad (2.11)$$

2.3.2 Travel time

The travel time of a road segment of length Δs is approximated as

$$T_t(v, v^+) = \frac{2\Delta s}{v + v^+}. \quad (2.12)$$

The expected value of travel time used in the incremental cost function (2.8) has the form,

$$\bar{T}_t = \mathbb{E}_{v^+}[T_t(v, v^+)]. \quad (2.13)$$

2.3.3 Distance constraint violation penalty

The penalty function for the distance constraint violation in (2.8) has one of the following forms. The first form,

$$\phi(\rho) = \begin{cases} \kappa & \text{if } \rho > \rho_{max}, \\ \kappa & \text{if } \rho < \rho_{min}, \\ 0 & \text{else,} \end{cases} \quad (2.14)$$

was used for the simulations while the second form,

$$\phi(\rho) = \begin{cases} (\kappa e^{\rho - \rho_{max}} - \kappa) & \text{if } \rho > \rho_{max}, \\ (\kappa e^{\rho_{min} - \rho} - \kappa) & \text{if } \rho < \rho_{min}, \\ 0 & \text{else,} \end{cases} \quad (2.15)$$

was used for the vehicle experiments. In the above, $[\rho_{min}, \rho_{max}]$ is the desired interval in which the distance is to be maintained and $\kappa > 0$ is a parameter. The value of $\kappa = 10$ was used to generate SDP policies for fuel-efficient vehicle-following using (2.14) in simulations and $\kappa = 0.005$ was used to generate the SDP policies for the vehicle-following vehicle experiments using (2.15). These values of κ were selected after some initial tuning. Values of $\rho_{min} = 3$ m and $\rho_{max} = 10$ m were used for the simulations with (2.14) while values of $\rho_{min} = 5$ m and $\rho_{max} = 15$ m were used for the vehicle experiments using (2.15). A switch was made from (2.14) to (2.15) in order to eliminate discontinuities within the cost function.

Note that the minimum distance constraint is imposed to prevent the host vehicle from colliding with a preceding vehicle, while the maximum distance constraint is imposed to improve driving comfort and reduce the impact of the host vehicle actions on the flow of traffic around it. While it is true that imposing the maximum distance constraint does not guarantee that there will be no impact on following traffic, we assert that the smaller

the maximum distance is the less impact there will be on following traffic. However, as the distance constraint becomes more constrictive, the potential fuel economy gains may be reduced. Similar arguments and constraints have been posed in [23, 24, 26] to reduce impact on the following traffic.

No attempt has been made to take advantage of drag reduction in close following, though in principle such a possibility can be realized by prescribing a small value for ρ_{max} . Also, no attempt has been made to use asymmetric penalty factors in (2.14) or (2.15) and equally weight both relative distance extremes as both extremes are important for the safety of the host vehicle and surrounding traffic. These developments are left to future work.

2.4 Stochastic dynamic programming

SDP was used to generate best-on-average control policies, [42]. For the incremental cost (2.8), the stochastic optimal control problem that SDP solves is of the form

$$J = \mathbb{E} \left[\sum_{k=0}^{\infty} q^k R(v(k), \theta(k), \rho(k)) \right] \rightarrow \min_{u \in U}, \quad (2.16)$$

subject to the models in Section 2.3 where $0 \leq q < 1$ is a discount factor introduced to guarantee that the cost is finite and the set U denotes the set of feasible control values. In (2.16), the value of $q = 0.96$ was used in order to approximate the average cost.

We use the value iteration approach [40, 42] to solve the SDP problem. Under the standard assumptions, the following iterations converge as $n \rightarrow \infty$ to the value function $V^*(v_t, v, \theta, \rho)$,

$$V_{n+1}(v_t, v, \theta, \rho) = \min_{u \in U} Q_n(v_t, v, \theta, \rho, u), \quad (2.17)$$

$$Q_n(v_t, v, \theta, \rho, u) = \quad (2.18)$$

$$R(v_t, v, \theta) + \sum_{\theta^+, v_t^+} q V_n(v_t^+, v^+, \theta^+, \rho^+) \mathbb{P}(v_t^+, \theta^+ | v_t, \theta),$$

$$V_0(v_t, v, \theta, \rho) = 0. \quad (2.19)$$

The value iterations are performed numerically using standard gridding techniques. The value iteration was terminated once

$$\max || V_{n+1} - V_n || \leq C_{th}$$

was satisfied. For all presented results, a value of $C_{th} = 10^{-4}$ was used.

Once the optimization was terminated, the optimal control policy, U^* , is selected as the minimizer of (2.18)

$$U^*(v_t, v, \theta, \rho) \in \arg \min_{u \in U} Q(v_t, v, \theta, \rho, u).$$

The selected control policy, U^* , is then implemented directly without interpolation through addition according to either (2.1) or (2.2).

In the in-traffic driving scenario, there were a total of 3,198 state and control combinations (with 7 possible control values, $u \in \{-3, -2, \dots, 3\}$ m/s). For this scenario, it is assumed that $\phi(\rho) = 0$, thus eliminating ρ as a state in the optimization. This assumption is made due to the curse of dimensionality that plagues, in general, the SDP approach, [42]. If, for instance, ρ were included in the optimization, this raises the total number of states to around 67,000. Running this full optimization, though, is presently being considered for future work.

However, ρ is included as a state for the vehicle-following simulations and the vehicle tests. In these cases, the traffic speed is assumed to be constant which reduces the total state space, allowing for the addition of relative distance as a state. For the vehicle-following simulations, there are 1,911 state and control combinations with 7 possible control values.

For vehicle testing, the optimization was similar to that done for the vehicle-following scenario. This optimization included 819 state and control combinations with 3 feasible control values ($u \in \{-2, 0, 2\}$ mph)

2.5 Simulation case studies

To quantify the fuel economy benefits of the SDP policies, simulation case studies were implemented in a simulation environment based on CarSim and MATLAB/Simulink. The CarSim software package provides vehicle models and the capability of programming the road geometry over which the simulations will occur. The modification of desired vehicle parameters is then performed in Simulink, see Figure 2.4, which can be linked with CarSim. For the following simulations, two vehicles are simulated using the same CarSim models over a virtual M-39 constructed from the collected vehicle data. One of the simulated vehicles, the traffic analog vehicle, is programmed to follow a prescribed vehicle trajectory of any one of the sixteen data collection runs (the in-traffic driving scenario) or a set constant speed (the vehicle-following scenario). The second simulated vehicle, referred to as the host vehicle, is programmed to respond to the traffic analog vehicle according to the SDP policy for the respective scenario. The SDP policy is executed in Simulink. The

SDP policies were generated to prescribe the offset relative to traffic speed in (2.1),

$$u \in \{-3, -2, -1, 0, 1, 2, 3\} \text{ m/s} .$$

The metrics used to evaluate the results are the percent fuel economy improvement (PFEI) and percent difference in average vehicle speed (PDAS). They are defined by

$$PFEI = \frac{mpg_h - mpg_t}{mpg_t} \times 100, \quad PDAS = \frac{\bar{v}_h - \bar{v}_t}{\bar{v}_t} \times 100, \quad (2.20)$$

where mpg_t , \bar{v}_t are, respectively, the fuel economy and the average speed of the traffic analog vehicle, while mpg_h , \bar{v}_h are, respectively, the fuel economy and the average speed of the host vehicle. Positive values of $PFEI$ indicate the fuel economy improvements of the host vehicle over the traffic analog vehicle and positive values of $PDAS$ indicate faster average speed of the host vehicle as compared to the traffic analog vehicle.

2.5.1 In-traffic driving

For the in-traffic driving scenario, the traffic analog is programmed to follow the sixteen data collection drives. In the problem we are solving, we are minimizing the cost (2.16) with

$$R = \bar{W}_f + \lambda \bar{T}_t, \quad (2.21)$$

where \bar{W}_f is given by (2.11) and \bar{T}_t is given by (2.13). The minimization is performed with respect to the control input $u \in \{-3, -2, -1, 0, 1, 2, 3\}$ m/s where the progression of traffic speed, v_t , and road grade, θ , described by Markov chain models (2.5) and (2.7) and vehicle dynamics described by (2.1).

The value of the weight $\lambda = 0.002$ in (2.8) was chosen after some tuning. We assumed $\phi(\rho) \equiv 0$ in (2.8) at the stage of generation of SDP policies. As a result, the SDP policy is independent of the relative distance, ρ , which simplified its off-line computation and online implementation. When evaluating this SDP policy, two case studies of non-porous traffic and porous traffic were considered.

2.5.1.1 Non-porous traffic

In the non-porous traffic case, the host vehicle was not able to pass the traffic analog vehicle and the brakes would be activated in the simulation to ensure no collision occurred. The non-porous traffic represented the worst case for the evaluation of SDP policy. The results based on our sixteen drives are summarized in Figure 2.5. For the non-porous traffic case,

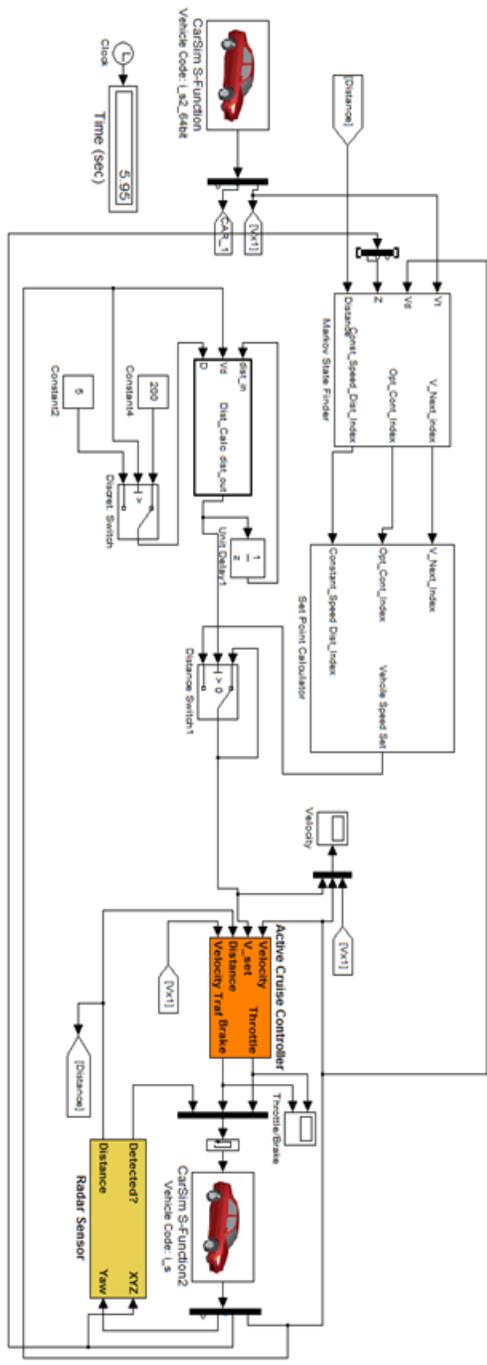


Figure 2.4: Schematic of the Simulink control architecture.

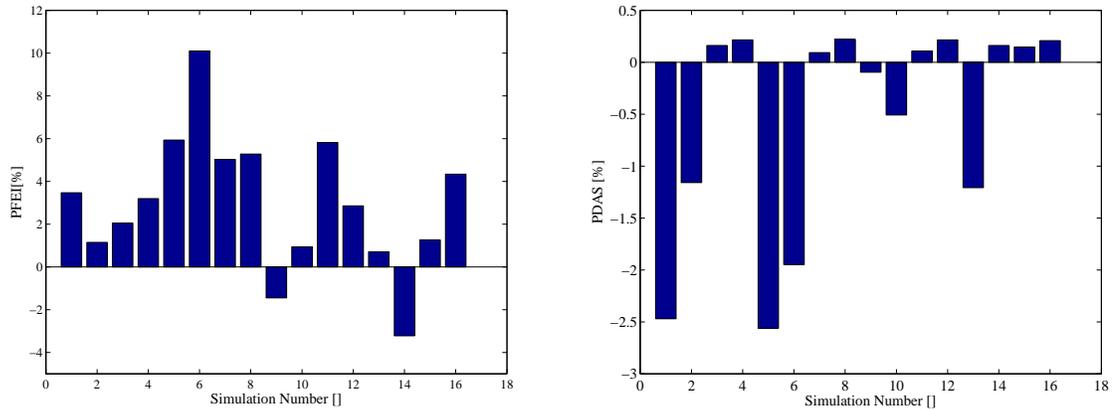


Figure 2.5: Left: The percent fuel economy increase for the in-traffic driving case study in the non-porous traffic case. Right: The percent difference in average speed.

the average improvement in mpg is 2.97% with an average drop in average speed of 0.78%. Slight positive values of the average speed difference on the right of Figure 2.5 for some of the drives indicate that the host and traffic analog vehicles ended up closer together at the end of the simulation than when the simulation was started.

Figure 2.6 compares the vehicle speed time histories of the traffic analog vehicle and host vehicle for drive number 6 where the largest fuel economy improvement has been observed. The speed of the host vehicle with the SDP policy has fewer and smaller abrupt changes and an overall “smoothed-out” character, for large accelerations and decelerations. This conclusion is also confirmed by examining the vehicle speed traces for other drives.

2.5.1.2 Porous traffic

In the porous traffic case, the host vehicle is able to follow the SDP policy freely passing or being passed by the traffic analog vehicle if needed. This case is consistent with the cost function used for SDP policy generation and represents the best case scenario when v_t corresponds to the average traffic speed but no actual vehicle hinders the host vehicle’s progress. The results based on the sixteen drives are shown in Figure 2.7. For the porous traffic case, the SDP policy achieved simultaneous improvement in mpg (average improvement is 5.67%) and an increase in average speed (average increase is 5.38%).

Note that in the non-porous traffic case, the average speed for many of the simulations is lower than the traffic analog vehicle. Though one can argue that some of the fuel economy improvement can, in principle, be due to this reduction, it is clear from the porous traffic case results that simultaneous increases in average speed and fuel economy are also

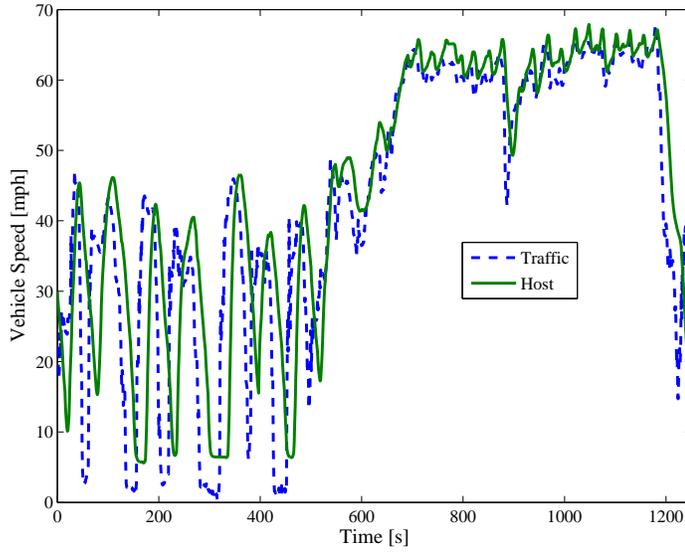


Figure 2.6: The time histories of the traffic analog vehicle (dotted) and the host vehicle (solid) speeds for the simulation 6 in the non-porous traffic case.

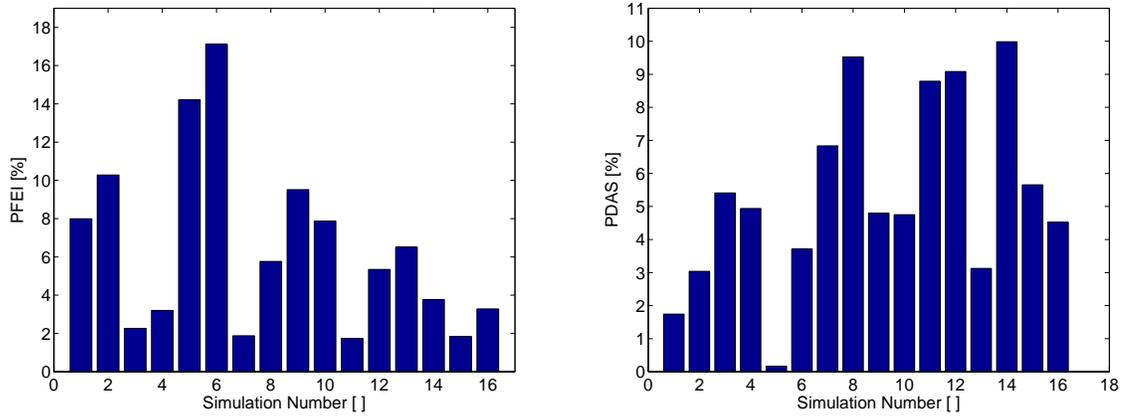


Figure 2.7: Left: The percent fuel economy increase for the in-traffic driving case study in the porous traffic case. Right: The percent difference in average speed.

Table 2.1: Fuel economy improvement broken down between traffic conditions and traffic type.

Traffic Condition	Traffic Type	Percent Improvement in mpg
Rush Hour	Porous	8.98
Rush Hour	Non-Porous	2.20
Off-Peak	Porous	2.39
Off-Peak	Non-Porous	3.73

feasible.

2.5.1.3 Rush hour versus off-peak traffic conditions

Table 2.1 compares the average results over rush hour simulations (drives 1, 2, 5, 6, 9, 10, 13, 14) and off-peak traffic conditions (drives 3, 4, 7, 8, 11, 12, 15, 16). In both the porous and non-porous cases, some fuel savings, as compared to the traffic analog vehicle, come from smoothing the accelerations and a reduction in the amount of vehicle braking. This observation is consistent with the results that in the non-porous case the host vehicle performs better than traffic during off-peak times while in the porous case the host vehicle does better during rush-hour times. However, in the porous case when the traffic is traveling at a relatively constant speed, time-varying cruise patterns begin to emerge from the host vehicle around the traffic analog vehicle’s constant speed. This phenomenon is discussed in the next section.

2.5.2 Optimal vehicle-following

The vehicle-following scenario is now considered. In the problem we are solving, we are minimizing the cost (2.16) with

$$R = \bar{W}_f + \lambda \bar{T}_t + \phi(\rho), \quad (2.22)$$

where \bar{W}_f is given by (2.11), \bar{T}_t is given by (2.13), and $\phi(\rho)$ is given by (2.14). The minimization is performed with respect to the control input $u \in \{-3, -2, -1, 0, 1, 2, 3\}$ m/s where traffic speed, $v_t \in \{45, 50, 55, 60\}$ mph, is held constant, road grade, θ , described by the Markov chain model in (2.5) and vehicle dynamics described by (2.1).

For this case, separate SDP policies with $\phi(\rho)$ defined by (2.14), $\kappa = 10$, and $\lambda = 0.012$ in (2.8) were generated for each of four values of v_t . Furthermore, the cases of uniformly

zero grade along the route and non-zero grade modeled stochastically, as in the in-traffic scenario, were considered separately. The SDP policies are functions of ρ , v , and θ (in the non-zero grade case) and prescribe the offset u in (2.1). Note that the zero grade policies result from deterministic dynamic programming. The zero grade policies were simulated using a flat road profile with the same length as the M-39 grade profile while the non-zero grade policies were simulated with the M-39 grade profile.

As is seen in Figure 2.8, the vehicle-following SDP policies improve the fuel economy at all four of the tested speeds. This improvement is attributed to a time-varying oscillation of the host vehicle speed induced by the SDP policies, see Figures 2.9-2.10. The time-varying behavior has been observed in both non-zero grade (Figure 2.9) and zero grade (Figure 2.10) simulated drives. The oscillating patterns in non-zero grade simulation become more regular in the zero grade case. The fuel flow during sharp acceleration exceeds the fuel flow of the lead vehicle, but is less during the deceleration phase. This results in a lower overall fuel consumption, seen in the right side of both Figures 2.9 and 2.10 of the host vehicle compared to the lead vehicle. Due to the distance constraint, average vehicle speeds of the host and lead vehicles are nearly the same.

The fuel improvement mechanism of the time-varying oscillations differs from that of acceleration smoothing. The acceleration smoothing limits the magnitude of the accelerations based on prediction of the lead vehicle velocity. It keeps the host vehicle from braking as much as traffic, potentially also avoiding unnecessary down shifts and torque converter unlocks. With the time-varying oscillations, the lead vehicle is assumed to be at a constant and known speed, and the control policy takes advantage of differences in efficiency between operating points in the engine map, and of efficiently executed transients, to reduce fuel consumption.

It should be noted that the time-varying oscillations were observed in a diminished form in the in-traffic scenario when the traffic analog vehicle would travel at a relatively constant speed for an extended period of time. However, in the vehicle-following case, the oscillations in vehicle speed were quite distinct. We also note that fuel efficiency improvements with time-varying vehicle speed profiles were demonstrated in [29–31]. In these cases, these time-varying profiles were assumed to be fuel efficient and, as such, were sought out. However, in our case the time-varying patterns emerged as a result of implementing SDP policies and were not initially sought out.

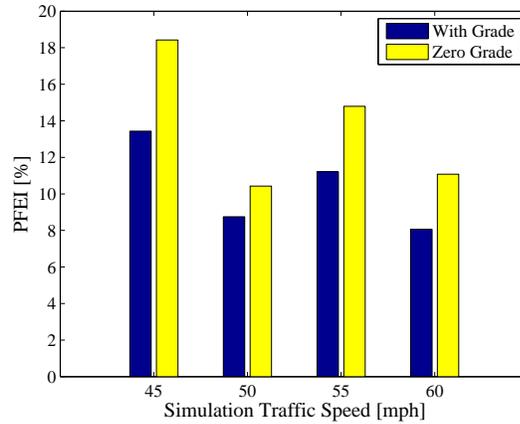


Figure 2.8: Fuel economy improvement with SDP policies in the vehicle-following scenarios. The blue (first) set of bars are for the non-zero grade policies and the yellow (second) set of bars are for zero grade.

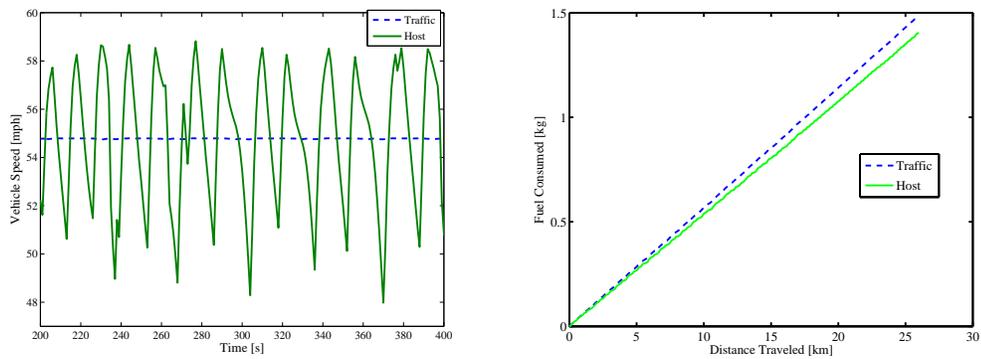


Figure 2.9: Left: Time history of traffic analog vehicle speed (dashed) and host vehicle speed (solid) for a part of the drive with non-zero grade. Right: Fuel consumption versus distance traveled of traffic analog vehicle speed (dashed) and host vehicle (solid).

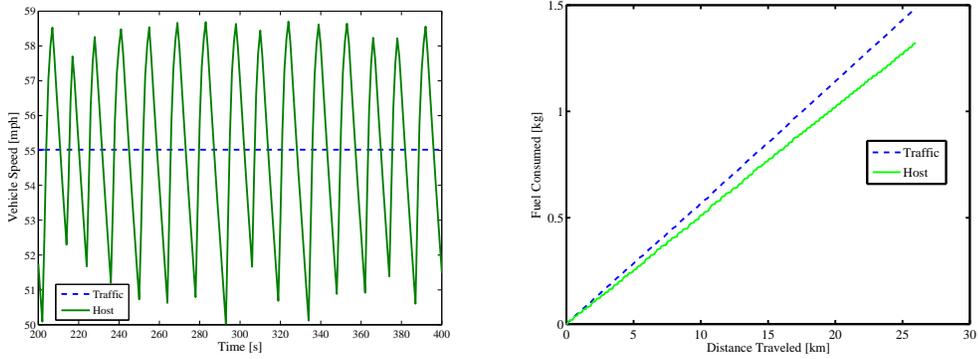


Figure 2.10: Left: Time history of traffic analog vehicle speed (dashed) and host vehicle (solid) for a part of the drive with zero grade. Right: Fuel consumption versus distance traveled of traffic analog vehicle speed (dashed) and host vehicle (solid).

2.6 Vehicle experiments

The experimental testing of the fuel-efficient vehicle-following scenario was performed in a 2007 Ford Edge. The tests were performed along the M-39 highway portion of the same route used for data collection. For the implementation, the vehicle speed set-point fed to the vehicle cruise controller was bypassed with the output of SDP policy running on a dSPACE RTI 1005 board, see Figure 2.11 for a similar test vehicle setup. The testing was restricted to vehicle-following scenario, where the traffic vehicle driving at a constant speed was implemented as a virtual vehicle (in software) while the host vehicle was the experimental vehicle. This implementation permitted testing with the single available vehicle which did not have a radar nor adaptive cruise control. The vehicle was driven over the same route at least twice whenever testing was performed, once with the regular cruise control and the others with the SDP policy. This was done in order to reduce errors due to day-to-day fluctuations in fuel economy and to make accurate assessments of fuel economy benefits or detriments.

After preliminary testing, to improve driver/passenger comfort, the SDP policies were regenerated to prescribe the offset to the current host vehicle speed, v , (see (2.2)) rather than to the traffic vehicle speed, v_t , and restricted the offset values to a small range $u \in \{-2, 0, 2\}$ mph to prevent large speed change requests. We additionally constrained u by adding to U the constraint $50 \leq v + u \leq 60$. A value of $\lambda = 0.012$ was used with the distance constraint penalty (2.15) with $\kappa = 0.005$ in the cost (2.8).

To summarize the formulation, the optimization problem we are solving is to minimize



Figure 2.11: A similar vehicle setup what was used in the 2007 Ford Edge experimental vehicle.

the cost

$$R = \bar{W}_f + \lambda \bar{T}_t + \phi(\rho), \quad (2.23)$$

where \bar{W}_f is given by (2.11), \bar{T}_t is given by (2.13), and $\phi(\rho)$ is given by (2.15). The minimization is performed with respect to the control input $u \in \{-2, 0, 2\}$ mph where traffic speed, $v_t = 54$ mph, is held constant, road grade, θ , described by the Markov chain model in (2.5) and vehicle dynamics described by (2.2).

Over 12 drives on M-39 (half in northbound and half in southbound direction) with $v_t = 54$ mph, the average fuel economy improvement of 4.51% with a maximum of 11.58% and a minimum of -3.28% was recorded. The maximum and minimum were observed on the same day during which there was a strong wind that may have skewed the results. Without these outliers, the average fuel economy improvement was 5.38%.

Figure 2.12 shows the results of vehicle experiments for one of the drives. An oscillating pattern, a similar pattern to that in Subsection 2.5.2, clearly emerges in the experimental vehicle speed. The difference in the character of the response is attributed to changes in SDP problem formulation that were made to improve vehicle drivability and to the slow and asymmetric (between acceleration and deceleration) response of the nominal cruise controller to the set-point provided by the SDP policy. Also, due to the nominal cruise controller response (unmodeled at the stage of generating SDP control policies), the average speed of the host vehicle was below the lead vehicle average speed. The distance ρ computed by a limited integrator in our implementation was thus saturated at the upper bound, ρ_{max} , during the tests.

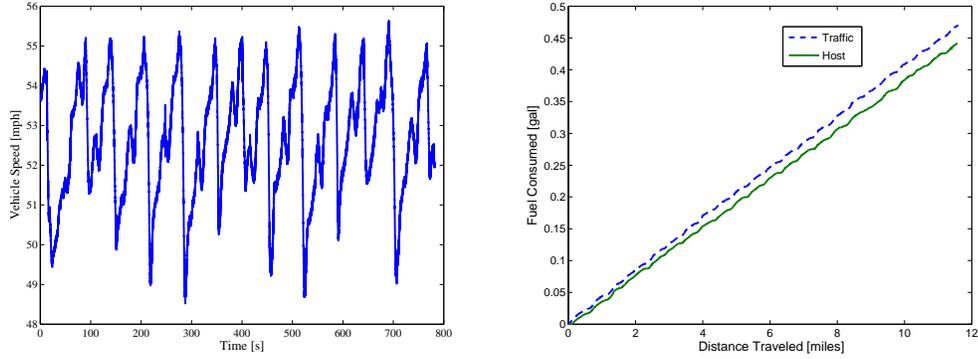


Figure 2.12: Left: Time history of vehicle speed from vehicle testing. Right: Cumulative flow of the host vehicle (solid) versus traffic vehicle (dashed).

The average speed difference over all 12 drives is -1.95 mph. To account for the increased fuel economy due to a lower average speed, test drives were performed using the standard cruise control set to 50, 52, and 54 mph in order to build a relationship between a reduction in average speed and increase in fuel economy. The resulting relationship is, on average, a 0.49% reduction in fuel consumption for every percent drop in average speed. This results in a corrected fuel consumption improvement of 2.74% when considering all of the tests and an improvement of 3.61% when not considering the outliers.

2.7 Sensitivity analysis of the SDP policy results

Two sensitivity analyses are presented in this Section. The first is SDP policy sensitivity to changing weights in the cost function (2.8). The second is SDP policy sensitivity to changes in the Markov chain model grid selection.

2.7.1 Sensitivity to cost function parameters

A sensitivity analysis was performed over a range of λ for the in-traffic simulation scenario and over a range of λ and κ for the vehicle-following simulation scenario.

For the in-traffic scenario, different policies were generated for different values of λ with $\lambda \in \{0, 0.002, 0.004, 0.006, 0.008, 0.01, 0.012, 0.014, 0.016, 0.018, 0.02\}$. Each policy was tested over the sixteen simulation test runs, under the porous traffic assumption, and the results were statistically analyzed and summarized in Figure 2.15. The policies were generated using Markov chain models for road grade and vehicle speed that concatenated all of the available data. This was done intentionally to test the effects of changing

the cost weights with no other changing factors between the simulations.

The results of the sensitivity analysis are in Figure 2.15. In this figure, positive values indicate better-on-average fuel economy. To generate the values in Figure 2.15, each of the sixteen test runs were simulated with each of the different λ policies, producing a set of *PFEIs* and *PDASs* (produced 176 values for both *PFEI* and *PDAS*), see Figure 2.13 for the specific results for test run 5.

Then, for the i^{th} test run, the average percent fuel economy improvement, $PFEI_i^*$, and average percent difference in average speed, $PDAS_i^*$, over the different λ policies was calculated excluding the $\lambda = 0$ policy results. This produces a mean fuel economy improvement for the i^{th} test run over all the different values of λ used to generate control policies.

Then, for the i^{th} test run, the percent difference in *PFEI* and *PDAS* from $PFEI_i^*$ and $PDAS_i^*$ for each λ policy result is calculated so that the results are normalized for the given test run, see Figure 2.14 for the specific results for test run 5 while noting that the values now represent a percent difference from the mean of the values in Figure 2.13.

This process produces a set of values for each test run that demonstrates how much the change in λ affects change in *PFEI* and *PDAS* relative to the average result for that test run. The average and standard deviation of these percent differences is then calculated for each λ policy and reported as μ in Figure 2.15 with standard deviation σ .

Percent differences from the mean were calculated for each test run because the raw *PFEIs* between the different test runs varied widely (though in every case a fuel economy improvement was observed similar to Figure 2.7). To simplify the presentation of the results, and to understand the effects of changes in λ on average for all of the drives, these variations were eliminated. Also, the $\lambda = 0$ policy was excluded when averaging the results because, in each case, it was a large outlier due to the fact the policies are intrinsically different when $\lambda = 0$ in that they do not consider travel time in the cost.

As is seen in Figure 2.15, the calculated μ and $\mu \pm \sigma$ for the *PFEI* (on the left) are within a $\pm 10\%$ bound, with the cost being dominated by travel time for $\lambda \geq 0.01$, with the exception of the results for $\lambda = 0$ and $\mu - \sigma$ for $\lambda = 0.002$. However, when $\lambda = 0$ and travel time is not considered within the cost function, the results can vary greatly. This is primarily due to the difference in speed profiles (results seen on the right side of Figure 2.15). Also seen in both plots in Figure 2.15 is less variability in results as λ increases and the cost function becomes dominated by the consideration of travel time.

Figure 2.15 also demonstrates that the selection of $\lambda = 0.002$ was not the best choice. This is because at $\lambda = 0.002$, the average percent difference from mean *PFEI* is less than zero. This means that, for the i^{th} test run with $\lambda = 0.002$, it is likely the *PFEI* was

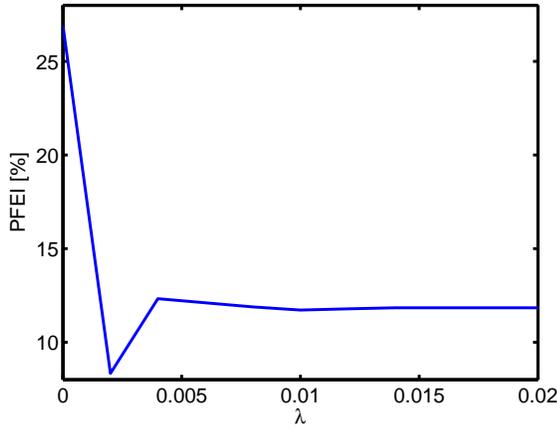


Figure 2.13: PFEI of in-traffic driving scenario for drive number 5 over a spread of different λ weights.

less than the mean, $PFEI_i^*$, for that test run. Thus, a different choice in weighting value (perhaps $\lambda = 0.004$ judging from Figure 2.15) would have produced, on average, higher $PFEI$ across the sixteen test runs.

For the vehicle-following scenario, different policies were generated for different values of λ and κ with $\lambda \in \{0.002, 0.006, 0.01\}$ and $\kappa \in \{10^{-6}, 5 \times 10^{-6}, 10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}, 0.1, 0.5, 1, 5, 10, 50, 100\}$ using the smooth relative distance constraint penalty (2.15). The results presented are the actual PFEI and PDAS specifically for the case when $v_t = 55$ mph. Three lines are plotted over the span of κ , one for each of the tested values of λ .

In Figure 2.16, it is seen that over a large range of κ the results do not greatly vary for both $PFEI$ and $PDAS$. The greatest deviations occur for all λ between $\kappa = 10^{-4}$ and $\kappa = 5 \times 10^{-4}$. For $\kappa > 0.1$, the cost function becomes dominated by the relative distance constraint penalty. It should be noted that the $PDAS$ plot on the right side of Figure 2.16 indicates that there is no significant change in the average vehicle speeds between the traffic analog vehicle and the host vehicle over all values of λ and κ that were tested.

2.7.2 Sensitivity to Markov chain model grid selection

A sensitivity analysis was performed over five different grid selections for the road grade Markov chain model in the vehicle-following driving scenario. These grid selections are based on changes in road grade of 0.25%, 0.5%, 1%, 1.5%, and 2%. This results in five different Markov chain models with properties listed in Table 2.2.

Figures 2.17 and 2.18 show the results of this analysis for vehicle-following policies

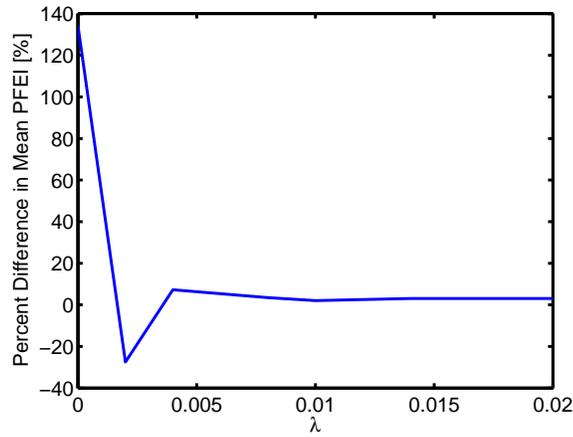


Figure 2.14: Adjusted PFEI of in-traffic driving scenario for drive number 5 over a spread of different λ . These results are the percent difference from the mean PFEI from Figure 2.13.

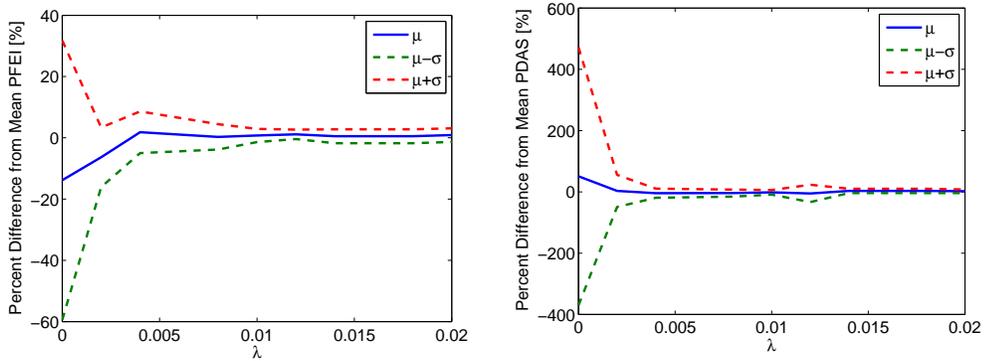


Figure 2.15: Left: The percent difference in fuel economy improvement statistics over the sixteen test drives with varying λ . Right: The percent difference in change in average vehicle speed statistics over the sixteen test drives with varying λ .

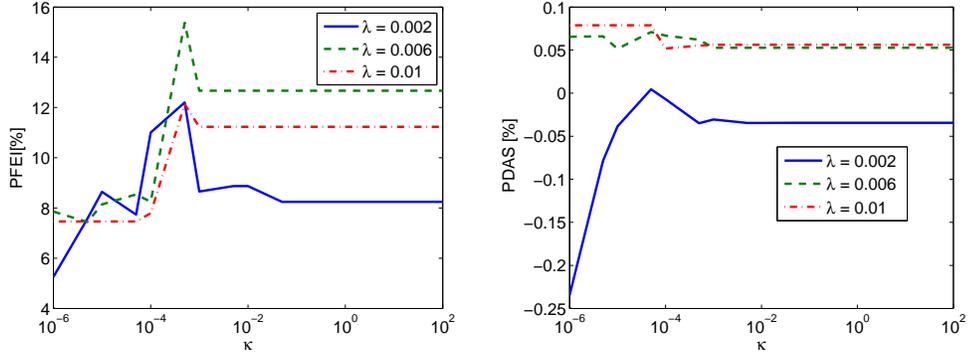


Figure 2.16: Left: Percent fuel economy improvement for varying κ with different λ . Right: Percent difference in average speed for varying κ and with different λ .

Table 2.2: The five Markov chain models used for model sensitivity analysis

Model	Grid Selection	States	Optimization States
1	0.25	$\theta \in \{-6, -5.75, -5.5, \dots, 5.75, 6\}\%$	7203
2	0.50	$\theta \in \{-6, -5.5, -5, \dots, 5.5, 6\}\%$	3675
3	1	$\theta \in \{-6, -5, -4, \dots, 5, 6\}\%$	1911
4	1.5	$\theta \in \{-6, -4.5, -3, \dots, 4.5, 6\}\%$	1323
5	2	$\theta \in \{-6, -4, -2, \dots, 4, 6\}\%$	1029

calculated with $\lambda = 0.006$ and $\kappa = 5 \times 10^{-4}$ using distance constraint penalty (2.15) with $v_t = 55$ mph. Figure 2.17 presents the PFEI for each of the different models. As is seen, the resultant PFEIs are all between 15-16%. Figure 2.18 presents the PDAS for each of the different models. Note that these results are in percent and thus are very similar.

This analysis was performed on the vehicle-following scenario only due to its initial lower dimensional state space. Future work will include a thorough sensitivity analysis of policy dependence on Markov chain model for the in-traffic driving scenario.

A different model sensitivity experiment can also be considered, that is, sensitivity to changes in modeling data. However, an analysis such as this would have to wait until a wide range of data can be collected (e.g., mountain driving, winter weather driving, etc).

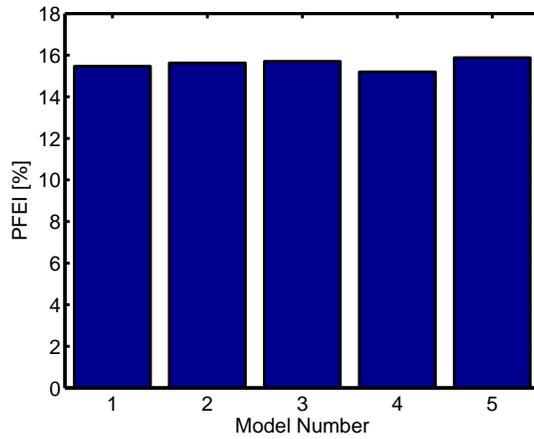


Figure 2.17: PFEI for the vehicle-following scenario with respect to different Markov chain models, see Table 2.2.

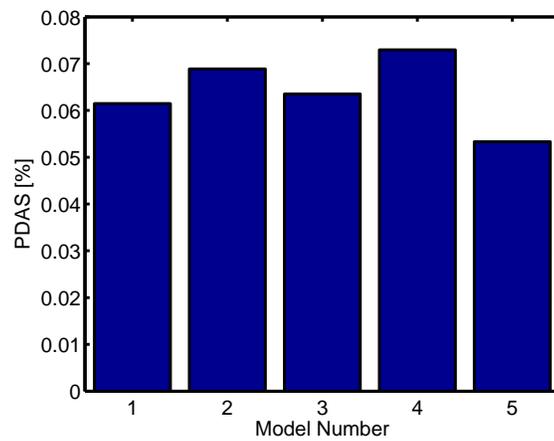


Figure 2.18: PDAS for the vehicle-following scenario with respect to different Markov chain models, see Table 2.2.

CHAPTER 3

A Practical Implementation Strategy of Time-Varying Fuel-Efficient Cruise Patterns

In Chapter 2, a stochastic dynamic programming (SDP) methodology for improving fuel economy through the control of vehicle speed is developed. SDP policies induced time-varying vehicle speed profiles that improved fuel economy. We refer to this behavior as time-varying cruise. The time-varying cruise pattern was observed in the porous-traffic case of the in-traffic driving scenario discussed in Subsection 2.5.1 and in the vehicle-following scenario discussed in Subsection 2.5.2. This pattern, seen in Figure 2.12, emerges when the traffic speed is relatively constant. In these cases, the SDP policy guides the vehicle in a series of accelerations and decelerations—a "pulse and glide" (PnG) pattern.

In this chapter, the time-varying cruise pattern is modeled and realized using a finite state machine (FSM). This is done in an effort to simplify the in-vehicle implementation of the strategy that induces time-varying cruise patterns and make it more appealing to calibration engineers in the automotive industry. With a FSM, each state represents a phase of the time-varying cruise pattern and provides a practical structure from which implementation can be straight forward. This strategy is referred to as rule based cruise control (RBCC).

The RBCC has four states. These states are normal, glide, recover, and bleed. In the normal state, the nominal cruise control is used to maintain a constant speed. In the glide state, the vehicle speed set-point is lowered to a minimum allowed value and the vehicle coasts to this speed while the vehicle transmission is in neutral. The recover state is the acceleration from the lowest allowed speed to the highest allowed speed. The bleed state occurs when it is not efficient to glide (particularly when there is a large positive grade) and instead a constant torque level is held in order to reduce the speed loss and stave off the recover state. Figure 3.1 provides examples of speed profiles in different RBCC states.

This chapter is organized as follows. The RBCC design problem is discussed in Section 3.1. The FSM structure will be presented and described, along with the thresholds

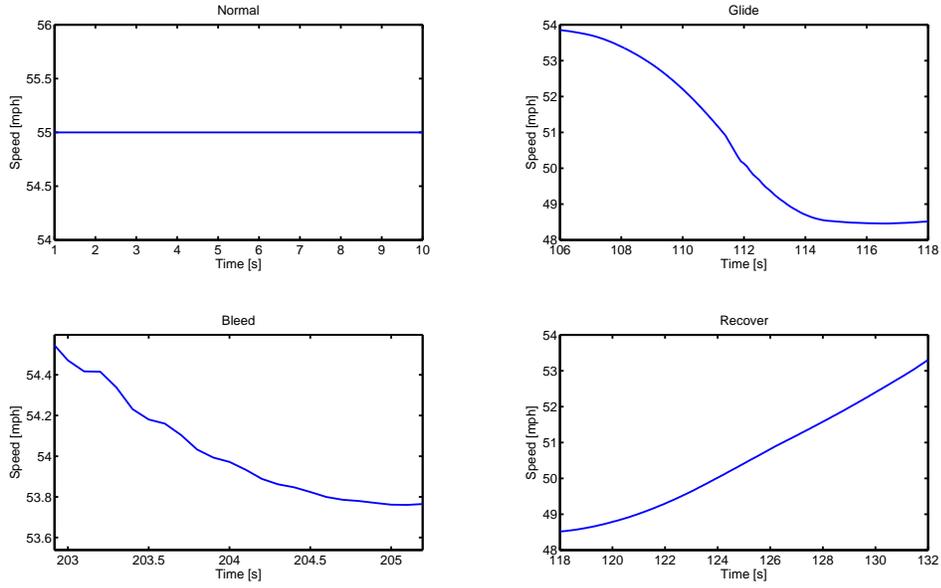


Figure 3.1: Examples of the vehicle speed profiles in the four RBCC states.

that govern the transitions between FSM states in Section 3.2. The method by which the transitions between the FSM states are optimized is described in Section 3.3. Results from the simulations and vehicle experiments with the RBCC are described in Section 3.4. Unlike the SDP policies in the previous chapter, the set-point set by the driver through the cruise control input is considered to be the maximum speed, i.e., the host vehicle will never exceed the driver set speed.

3.1 Rule based cruise control problem formulation

Consider a fuel consumption model of the form,

$$f_c(k) = G(v(k), v_t, u(k), \theta(k)), \quad (3.1)$$

where $f_c(k)$ is the instantaneous fuel consumption at time instant k , $v(k)$ is the instantaneous vehicle speed, v_t is the driver selected cruise control set-point, $u(k)$ is the modification to driver selected cruise control set-point with the commanded speed, consistent with (2.1), as $v_t + u(k)$, and $\theta(k)$ is the instantaneous road grade. The constraint $u(k) \leq 0$ is imposed so that the commanded speed does not exceed v_t . Note that the continuous dynamics are determined by the vehicle and the nominal cruise controller and are treated as

being discretely sampled.

Two baseline fuel consumption calculations are referenced throughout this chapter. The first is the result of $u(k) \equiv 0$, that is, $v(k) \equiv v_t$. The resulting fuel consumption over a route driven at a constant speed of v_t is $f_{c,t}$. The second is a result of $v(k) \equiv v_{\text{ave}}$ where v_{ave} is the average speed of a vehicle speed profile induced by the RBCC strategy, with a resulting fuel consumption over a route of $f_{c,\text{ave}}$. The objective is to develop a control strategy that maximizes the cost function,

$$f_{c,\text{dyn}} = \frac{f_{c,\text{ave}} - \sum_{k=0}^n f_c(k)}{f_{c,t}} \times 100. \quad (3.2)$$

The cost function (3.2) quantifies the dynamic fuel consumption improvement (DFCI). Specifically, (3.2) is a percent improvement over the baseline $f_{c,t}$ along a route with horizon n , but discounts fuel economy improvement due to a change in the average vehicle speed and indirectly penalizes large deviations of vehicle speed from the cruise control set-point.

Note that if $f_{c,\text{dyn}} > 0$, the RBCC strategy is more fuel-efficient than traveling at $v(k) = v_{\text{ave}}$. If, in addition, $f_{c,t} \geq f_{c,\text{ave}}$, then the RBCC strategy is more fuel-efficient than traveling at $v(k) = v_t$. Also note that the distinction is made between v_{ave} and v_t , where v_t is the maximum allowed speed set by the driver while v_{ave} is the average speed the vehicle travels using the RBCC strategy.

3.2 Rule based cruise control

In this section, the structure and state transitions of the FSM are described. Within each state, two outputs are commanded, they are the speed offset command, $u(k)$, and neutral command, $n(k) \in \{0, 1\}$. For the neutral command, $n(k) = 0$ indicates the vehicle is in gear at a time instant k and $n(k) = 1$ indicates the vehicle is in neutral. The FSM is shown graphically in Figure 3.2 with its four states. Let $x(k)$ be the state of the FSM with outputs $u(k)$ and $n(k)$ described as

$$\begin{aligned} x(k+1) &= f(A_{x(k)}, x(k)), \\ u(k) &= h_1(x(k)), \\ n(k) &= h_2(x(k)), \end{aligned}$$

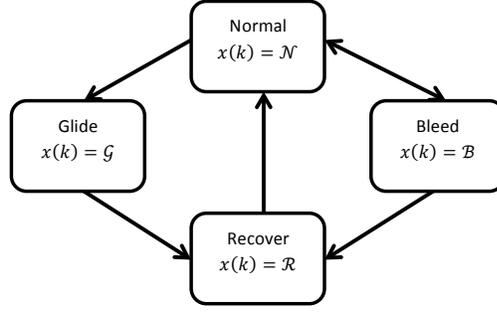


Figure 3.2: Allowed state transitions in the finite state machine for Rule Based Cruise Control (RBCC). Not shown are the transitions from a given state to itself which is a possibility in all states.

where $x(k) \in \{\mathcal{N}, \mathcal{G}, \mathcal{B}, \mathcal{R}\}$ and

$$A_{x(k)} \subset A = \{v_d, v_r, \bar{\theta}_{\mathcal{N}\mathcal{G}}, \bar{\theta}_{\mathcal{N}\mathcal{B}}, \bar{\theta}_{\mathcal{B}\mathcal{R}}\},$$

are the set of parameters defining thresholds that dictate the state transitions. Here v_d is the maximum deviation from the cruise control set-point v_t , v_r is the recovery speed, $\bar{\theta}_{\mathcal{N}\mathcal{G}}$ is the normal to glide grade transition threshold, $\bar{\theta}_{\mathcal{N}\mathcal{B}}$ is the neutral to bleed grade transition threshold, and $\bar{\theta}_{\mathcal{B}\mathcal{R}}$ is the bleed to recovery grade transition threshold. Furthermore, assume that $(v_t - v_r) < v_d$.

The mappings $f(A_{x(k)}, x(k))$, $h_1(x(k))$, and $h_2(x(k))$, are defined for each state as follows.

3.2.1 Normal state - \mathcal{N}

The normal cruise state of the RBCC holds a speed set-point determined by the driver. In the normal state, allowable state transitions are

$$\begin{aligned} x(k+1) &= f(A_{\mathcal{N}}, \mathcal{N}) \\ &= \begin{cases} \mathcal{G}, & \text{if } d_s d_{pl} = 1 \\ \mathcal{B}, & \text{if } d_s d_{ph} = 1 \\ \mathcal{N}, & \text{if } d_s(1 - (1 - d_{pl})(1 - d_{ph})) = 0 \end{cases}, \end{aligned}$$

where $A_{\mathcal{N}} = \{v_r, \bar{\theta}_{\mathcal{NG}}, \bar{\theta}_{\mathcal{NB}}\}$,

$$d_{\text{pl}} = \begin{cases} 1, & \text{if } \bar{\theta}(k) \leq \bar{\theta}_{\mathcal{NG}} \\ 0, & \text{if } \bar{\theta}(k) > \bar{\theta}_{\mathcal{NG}} \end{cases},$$

$$d_{\text{ph}} = \begin{cases} 1, & \text{if } \bar{\theta}(k) \geq \bar{\theta}_{\mathcal{NB}} \\ 0, & \text{if } \bar{\theta}(k) < \bar{\theta}_{\mathcal{NB}} \end{cases},$$

$$d_{\text{s}} = \begin{cases} 1, & \text{if } v_t - v(k) \leq v_r \\ 0, & \text{if } v_t - v(k) > v_r \end{cases}, \quad (3.3)$$

and where $\bar{\theta}(k)$ is the average predicted grade over the horizon $k \dots, k+r$. An autoregressive grade prediction method, developed by Ford Motor Company (see appendix A or [46]), was used to generate $\bar{\theta}(k)$ for the results presented in this chapter.

The alteration to the cruise control set-point is

$$u(k) = h_1(\mathcal{N}) = \Delta v(k-1),$$

and the neutral command is

$$n(k) = h_2(\mathcal{N}) = 0.$$

3.2.2 Glide state - \mathcal{G}

The glide state allows the vehicle to naturally decelerate while in neutral. In the glide state, allowable state transitions are

$$x(k+1) = f(A_{\mathcal{G}}, \mathcal{G}) = \begin{cases} \mathcal{R}, & \text{if } d_{\text{s}} = 1 \\ \mathcal{G}, & \text{if } d_{\text{s}} = 0 \end{cases},$$

where $A_{\mathcal{G}} = \{v_d\}$, and

$$d_{\text{s}} = \begin{cases} 1, & \text{if } v_t - v(k) \geq v_d \\ 0, & \text{if } v_t - v(k) < v_d \end{cases}.$$

The alteration to the cruise control set-point is

$$u(k) = h_1(\mathcal{G}) = v(k) - v_t,$$

and the neutral command is

$$n(k) = h_2(\mathcal{G}) = 1.$$

Here, $u(k)$ is designed such that $v(k) = v(k)$ so that the vehicle naturally decelerates.

3.2.3 Bleed state - \mathcal{B}

The bleed state holds an engine torque level to reduce vehicle speed loss, particularly on large grades. In the bleed state, allowable state transitions are

$$\begin{aligned} x(k+1) &= f(A_{\mathcal{B}}, \mathcal{B}) \\ &= \begin{cases} \mathcal{N}, & \text{if } d_{\text{sh}} = 1 \\ \mathcal{R}, & \text{if } (1 - d_{\text{sl}})(1 - d_{\text{p}}) = 0 \\ \mathcal{B}, & \text{if } (1 - d_{\text{sh}})(1 - d_{\text{sl}})(1 - d_{\text{p}}) = 1 \end{cases}, \end{aligned}$$

where $A_{\mathcal{B}} = \{v_d, v_r, \bar{\theta}_{\mathcal{BR}}\}$,

$$d_{\text{sh}} = \begin{cases} 1, & \text{if } v_t - v(k) < v_r \\ 0, & \text{if } v_t - v(k) \geq v_r \end{cases},$$

$$d_{\text{sl}} = \begin{cases} 1, & \text{if } v_t - v(k) \geq v_d \\ 0, & \text{if } v_t - v(k) < v_d \end{cases},$$

and

$$d_{\text{p}} = \begin{cases} 1, & \text{if } \bar{\theta} \leq \bar{\theta}_{\mathcal{BR}} \\ 0, & \text{if } \bar{\theta} > \bar{\theta}_{\mathcal{BR}} \end{cases}.$$

The alteration to the cruise control set-point is

$$\begin{aligned} u(k) = h_1(\mathcal{B}) &= F_1(F_{\text{PI}}(q)(\tau_{\mathcal{B}} - \tau(k)) \\ &\quad - F_2v(k)) + v(k) - v_t, \end{aligned}$$

where $\tau_{\mathcal{B}}$ is the desired bleed torque, $\tau(k)$ is an estimate of engine torque, and F_1 and F_2 are known cruise control parameters used to linearize $u(k)$ to the $\tau(k)$ control path. $F_{\text{PI}}(q)$ is a proportional plus integral (PI) controller used to track the desired torque $\tau_{\mathcal{B}}$.

The neutral command is

$$n(k) = h_2(\mathcal{B}) = 0.$$

3.2.4 Recover state - \mathcal{R}

The recover state accelerates the vehicle to v_r , the recover speed. The nominal cruise control has an allowed variance in the control of the speed. That is, the nominal cruise control system will keep the vehicle speed within $v_t - \Delta v \leq v(k) \leq v_t + \Delta v$. The value of v_r is slightly less than v_t and recovering to this speed is intended to put the vehicle speed within the allowed variance of the nominal speed controller. In the recover state, the allowable state transitions are

$$x(k+1) = f(A_{\mathcal{R}}, \mathcal{R}) = \begin{cases} \mathcal{N}, & \text{if } d_s = 1 \\ \mathcal{R}, & \text{if } d_s = 0, \end{cases},$$

where $A_{\mathcal{R}} = \{v_r\}$,

$$d_s = \begin{cases} 1, & v_t - v(k) \leq v_r \\ 0, & v_t - v(k) > v_r \end{cases}.$$

The alteration to the cruise control set-point is

$$u(k) = h_1(\mathcal{R}) = F_1(F_{\text{PI}}(q)(\tau_{\mathcal{R}}(k) - \tau(k)) - F_2v(k)) + v(k) - v_t,$$

$$\tau_{\mathcal{R}}(k) = \begin{cases} s(\bar{\theta}(k)), & x(k) \neq \mathcal{R}, \\ \tau_{\mathcal{R}}(k-1), & x(k) = \mathcal{R}, \end{cases}$$

$$s(\bar{\theta}) = \begin{cases} \tau_{\min}, & m_{\mathcal{R}}\bar{\theta} + b_{\mathcal{R}} < \tau_{\min}, \\ m_{\mathcal{R}}\bar{\theta} + b_{\mathcal{R}}, & \tau_{\min} \leq m_{\mathcal{R}}\bar{\theta} + b_{\mathcal{R}} \leq \tau_{\max}, \\ \tau_{\max}, & m_{\mathcal{R}}\bar{\theta} + b_{\mathcal{R}} > \tau_{\max}, \end{cases}$$

where τ_{\min} and τ_{\max} are the minimum and maximum desired engine torque values, respectively. The parameters $m_{\mathcal{R}}$ and $b_{\mathcal{R}}$ define the optimal recovery torque, which is a trade off

between time to recover speed and fuel consumption during recovery. These values are vehicle specific and known. The neutral command in the recovery state is

$$n(k) = h_2(\mathcal{R}) = 0.$$

3.3 Optimization

The state transitions in the finite state machine in Section 3.2 are dependent on six threshold parameters. These parameters are tuned to maximize the cost (3.2) subject to driver comfort considerations. The tuning guidelines and the off-line, model-based optimization procedure are now described.

The first parameter is the normal cruise set-point speed, v_t . The RBCC varies vehicle speed in a neighborhood below this value. As in conventional adaptive cruise control systems, v_t limits the maximum speed the vehicle will reach and is set according to posted speed limits or driver comfort.

The second parameter is the speed at which the vehicle speed is considered to be recovered, v_r . The value of v_r is set manually depending on the performance of the nominal cruise controller observed in vehicle experiments.

The third parameter is the maximum allowed drift speed from the set-point speed, v_d . Its value is determined as a result of the optimization but its range is restricted by considerations of the driver comfort. If the vehicle slows down too much with respect to the driver set-point speed, the driver may become uncomfortable or the vehicle may start impeding the traffic flow.

The fourth, fifth, and sixth parameters are the road grade related parameters $\bar{\theta}_{NG}$, $\bar{\theta}_{NB}$, and $\bar{\theta}_{BR}$. These three parameters and v_d are determined through the optimization.

The optimization problem is to determine the threshold parameter vector,

$$p = \left[v_d \bar{\theta}_{NG} \bar{\theta}_{NB} \bar{\theta}_{BR} \right]^T, \quad (3.4)$$

such that, for given values of v_r and v_t , the cost function (3.2) is maximized. Since the route, grade, and initial state of the RBCC machine are not known a priori, several speed and grade profiles are considered, and the average value of the cost (3.2) is maximized.

The approach to computing the average cost from the different speed and grade profiles consists of three steps. In the first step, grade profiles over sample routes are used to generate a statistical model of the grade in the form of a Markov chain. In this work a sample route based on M-39 highway in Michigan, which has a notable periodic grade,

was used. Techniques exemplified in the previous work [43–45, 47] and Chapter 2 were exploited to infer the transition probability matrix (TPM) of the Markov chain.

In the second step, a series of $N_\theta \times N_x \times N_{MC}$ Monte Carlo simulations are performed that are organized as follows. For each initial grade, $\theta(0)$, and for each initial state of the RBCC machine, $x(0)$, N_{MC} grade profiles are generated at random using the statistical model of the grade from the first step. Each simulation is run until a full cycle of state transitions of the RBCC machine is executed.

In the third step, the resulting values of the cost (3.2), denoted by $\bar{f}_{c,\text{dyn},i}^{x(0),\theta(0)}(p)$, $i = 1, \dots, N_{MC}$, are first averaged to yield the expected value of the cost (3.2) conditional to chosen $\theta(0)$, $x(0)$ and p :

$$\bar{f}_{c,\text{dyn}}^{x(0),\theta(0)}(p) = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \bar{f}_{c,\text{dyn},i}^{x(0),\theta(0)}(p), \quad (3.5)$$

and then averaged with respect to $\theta(0)$ and $x(0)$. For the latter averaging, the distribution of $\theta(0)$ is inferred from the steady state grade distribution as implied by the grade statistical model, while the distribution of $x(0)$ is assumed to be uniform. This leads to expressions,

$$\bar{f}_{c,\text{dyn}}^{x(0)}(p) = \sum_{i=1}^{N_\theta} \bar{f}_{c,\text{dyn}}^{x(0),\theta_i}(p) \mathbb{P}(\theta(0) = \theta_i), \quad (3.6)$$

where $\mathbb{P}(\theta(0) = \theta_i)$ denotes the probability that the initial grade $\theta(0)$ takes one of N_θ given values θ_i , and

$$\bar{f}_{c,\text{dyn}}(p) = \frac{1}{N_x} \sum_{i=1}^{N_x} \bar{f}_{c,\text{dyn}}^{x_i}(p), \quad (3.7)$$

where x_i , $i = 1, \dots, N_x$, denote the possible values of $x(0)$.

The average cost function $\bar{f}_{c,\text{dyn}}(p)$ is maximized with respect to p subject to range constraints on p . The optimization is performed using an adaptive mesh search algorithm. As a result, look up tables of parameters v_d , $\bar{\theta}_{NG}$, $\bar{\theta}_{NB}$, and $\bar{\theta}_{BR}$ are produced as functions of the given v_t and v_r . Note that this optimization is performed off-line and the online implementation of the RBCC logic and these look-up tables is straightforward.

3.4 Simulations and vehicle testing results

Results are now presented for the simulation and vehicle testing. A model of the experimental vehicle representing vehicle and engine dynamics was used to perform the threshold optimization and assess fuel consumption benefits. This model was provided by Dr. Dimi-

Table 3.1: The case studies and thresholds used.

case study	v_d	$\bar{\theta}_{NG}$	$\bar{\theta}_{NB}$	$\bar{\theta}_{BR}$
1	6	-1.5	0.5	-1
2	3.58	1.67	-1.68	1.5
3	3.9	1.4	-1.4	2.5

tar Filev and his research group at the Ford Motor Company. The experimental vehicle has a naturally aspirated engine and uses neutral and its highest gear during testing. The results presented are for the specific case where $v_t = 55$ mph and $v_r = 54.5$ mph for the same route along M-39 that was used in Chapter 2 for vehicle testing.

Three test case studies are considered. Case study 1 is a validation test with parameter values chosen heuristically for initial testing of the RBCC structure. Case study 2 is based on the threshold optimization which was performed over the intervals $v_d \in [3, 4]$ mph, $\bar{\theta}_{NG} \in [1, 2]\%$, $\bar{\theta}_{NB} \in [-2, -1]\%$, and $\bar{\theta}_{BR} \in [1, 2]\%$. Case study 3 is based on threshold optimization over the intervals $v_d \in [3, 4]$ mph, $\bar{\theta}_{NG} \in [1, 2]\%$, $\bar{\theta}_{NB} \in [-2, -1]\%$, and, $\bar{\theta}_{BR} \in [-2, 3]$. Note that the interval for $\bar{\theta}_{BR}$ was relaxed from case study 2 to case study 3 in an effort to improve results. Table 3.1 provides the values of the thresholds obtained after optimizations and used for each case study.

In all three case studies for vehicle testing, Ford’s autoregressive grade prediction, see [46] or Appendix A, was used to generate grade predictions over a prediction horizon r . For the simulation results, a preview of the grade was fed into the algorithm as the predicted grade in order to produce a benchmark for the vehicle testing. The differences in vehicle testing and simulation results are due to model error.

3.4.1 Simulation results

The simulations were performed for case studies 2 and 3. Case study 1 was not evaluated in the simulations and was only used to validate the RBCC strategy in the vehicle. For case study 2, an average speed of 52.38 mph was observed with a DFCI of 3.30%. For case study 3, the average speed was 52.15 mph and the DFCI was 3.54%. As can be observed, changing the intervals over which the optimization is performed did not produce significantly different simulated DFCI results, though this was not the case during the vehicle testing.

Figures 3.3 and 3.4 show the speed profiles of the vehicle under the RBCC scheme for case studies 2 and 3 as well as the RBCC machine states, where 10 corresponds to the

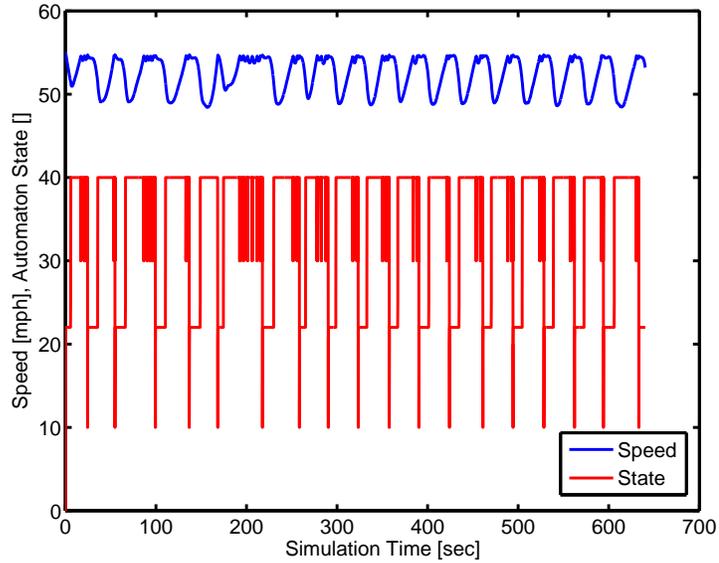


Figure 3.3: Simulation speed profile and RBCC states for case study 2.

normal cruise state, 20-25 are glide states, 30 is bleed state, and 40 is recover state. We note that for the implementation, the glide state was broken into a set of sub-states to generate a smooth transition in and out of neutral and, as such, states 20-25 correspond to the entire glide state.

Figures 3.5 and 3.6 provide examples of the grade profile and RBCC speed profile appropriately overlaid with the threshold values for case study 2. In Figure 3.6, it is seen that the vehicle speed drops below $v_t - v_d$. This is because $v_t - v_d$ is the lowest commanded speed, however, due to the dynamics of the nominal speed controller the vehicle speed may drift below the lowest commanded speed when it is passing from glide to recover. This phenomenon was also observed during the vehicle testing.

3.4.2 Test vehicle results

Vehicle tests were performed during the Fall of 2013 along the M-39 route. At the beginning of each test the vehicle was driven long enough to sufficiently warm the engine and transmission. Because M-39 has an average positive grade in the northbound direction, baseline drives were conducted in both directions at 55 mph to eliminate any effects due to the direction.

The vehicle testing results from all three case studies are summarized in Table 3.2. In Table 3.2, "NB" denotes the northbound direction and "SB" denotes the southbound

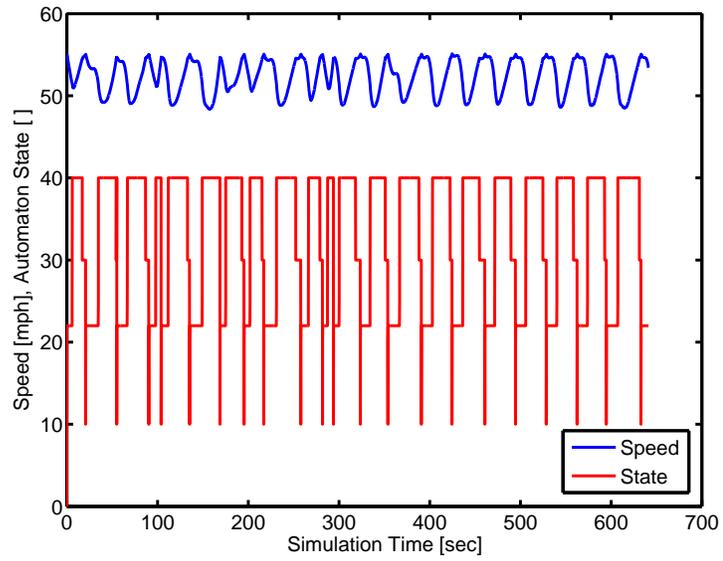


Figure 3.4: Simulation speed profile and RBCC states for case study 3.

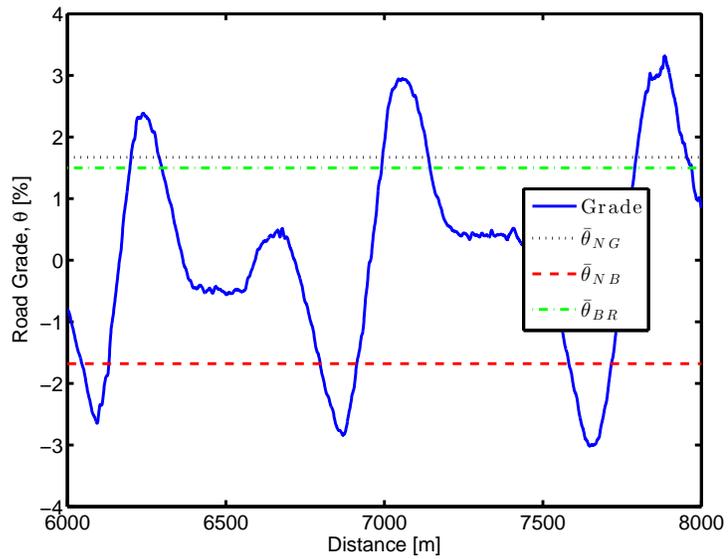


Figure 3.5: A section of the road grade along M-39 overlaid with the thresholds from case study 2.

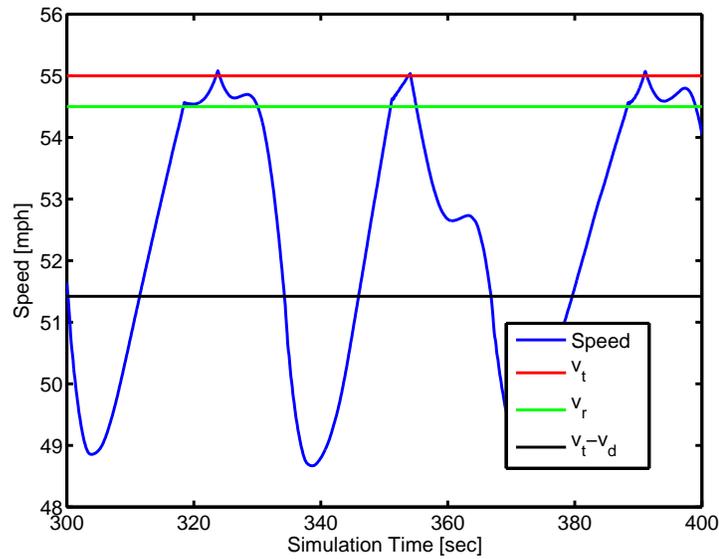


Figure 3.6: A section of the speed profile generated under the RBCC strategy overlaid with the thresholds from case study 2.

direction. As is observed, the RBCC strategy in case study 2 performs better than the RBCC strategy in case study 1, though it still maintains the primarily one-directional benefit observed with the heuristic threshold parameters. Case study 3 was then performed to eliminate this one-directional benefit. Though case study 3 eliminates the one-directional benefit, the overall benefit in the northbound direction is reduced.

In Chapter 2, stochastic dynamic programming (SDP) was used to develop a control policy that produces oscillations in the vehicle speed similar to the RBCC strategy. The SDP strategy was implemented on the same testing platform and tests were conducted on the same stretch of M-39. The SDP strategy produced an average DFCI of 2.74% for all vehicle tests. This is compared to an average of 2.99% in case study 2 and an average of 2.52% in case study 3 for RBCC (which do take advantage of neutral gear glides). The results are of similar magnitude in both cases, and improved upon in case study 2, while also benefited from a reduction in the overall complexity required to develop and implement the strategies. Note that four test drives were conducted for each case study to test RBCC and twelve test drives were conducted to produce the SDP strategy average.

Table 3.2: DFCI Results from Vehicle Testing

case study	NB $f_{c,dyn}$	NB v_{ave}	SB $f_{c,dyn}$	SB v_{ave}
1	3.43%	50 mph	-0.23%	51 mph
2	5.47%	52 mph	0.50%	52 mph
3	2.81%	51 mph	2.22%	50 mph

CHAPTER 4

Advances in the Stochastic Modeling of Road Grade and Vehicle Speed for Fuel-Efficient Cruise Control

This chapter addresses additional topics in the area of stochastic modeling of road grade and vehicle speed. Specifically, several novel developments in the use of the Kullback-Liebler (KL) divergence and divergence rate for categorizing and similarity assessment of traffic speed and road grade profiles, the use of a stochastic jump process to predict step changes in road grade, and the validation of online learning techniques of Markov chain models and stochastic jump models of road grade are reported.

This chapter is organized as follows. Section 4.1 demonstrates the use of the KL divergence rate to assess Markov chain model similarity. Section 4.2 discusses a novel stochastic jump process based approach to predicting road grade. Section 4.3 compares the stochastic jump grade prediction technique to the standard Markov model approach. Section 4.4 discusses methods of performing online learning of both the Markov chain and stochastic jump process prediction models.

4.1 Assessing the similarity of Markov chain models using a divergence technique

To practically implement a system based on stochastic control policies as in Chapter 2 or stochastically tuned finite state machines as in Chapter 3 for different road geometries and traffic patterns, a way of assessing the statistical similarity of different regions' terrain and traffic patterns is desirable. The basic motivation is that a stochastic control policy generated for one region may be used for another statistically similar region.

A symmetric version of the Kullback-Liebler (KL) divergence, referred to as the Jensen-Shannon divergence [99], and divergence rate allow for a statistical assessment of the similarity of different Markov chain models. The KL divergence between two Markov chains, [100], P and Q is

$$D_{KL}^M(P \parallel Q) = \sum \sum [P(x_j | x_i)] \log \left(\frac{P(x_j | x_i)}{Q(x_j | x_i)} \right), \quad (4.1)$$

where $P(x_j | x_i)$ and $Q(x_j | x_i)$ denote the transition probabilities. In (4.1), we assume that the entries in Q are greater than zero if the corresponding entries of P are greater than zero. If Q does not satisfy this property, we replace Q in (4.1) with

$$Q_{reg} = (1 - \epsilon)Q + \epsilon \left(\frac{1}{n} \right) E, \quad (4.2)$$

where ϵ is a parameter between 0 and 1 (we use $\epsilon = 0.0001$), n is the number of states in the Markov chain model, and E is an $n \times n$ matrix of ones.

The KL divergence has three properties:

1. The KL divergence is always non-negative
2. $D_{KL}^M(P \parallel Q) = 0$ iff $P = Q$
3. $D_{KL}^M(P \parallel Q) \neq D_{KL}^M(Q \parallel P)$ in general

A symmetric KL divergence [99] can be defined as

$$D_{KL,S}^M(P \parallel Q) = \frac{1}{2} (D_{KL}^M(P \parallel Q) + D_{KL}^M(Q \parallel P)). \quad (4.3)$$

The KL divergence rate between two Markov chains [100], P and Q is

$$R_{KL}^M(P \parallel Q) = \sum \sum [P(x_j | x_i) P^*(x_j)] \log \left(\frac{P(x_j | x_i)}{Q(x_j | x_i)} \right), \quad (4.4)$$

where P^* is the steady-state probability distribution of P which is assumed to exist and is determined as an eigenvector corresponding to the eigenvalue of 1, i.e., from

$$(P^*)^T P = (P^*)^T. \quad (4.5)$$

As in (4.1), Q is replaced in (4.4) with (4.2) under the same conditions.

The KL divergence rate, like the KL divergence, is non-negative, zero only when taken between two identical models, and is not necessarily symmetric. Similarly, a symmetric

KL divergence rate can be defined as

$$R_{KL,S}^M(P \parallel Q) = \frac{1}{2}(R_{KL}^M(P \parallel Q) + R_{KL}^M(Q \parallel P)). \quad (4.6)$$

The KL divergence rate can be shown to be bounded as the dimension of the Markov chain increases and it generalizes the KL divergence to the case when one cannot assume that the samples used to generate the models are independent, see [100].

To illustrate the use of the KL divergence rate to assess the model similarity and evaluate its classification ability, data over sixteen highway and eight city drives were collected and analyzed. The data were used to generate Markov chains using two approximations. The first approximation is that the road grade and vehicle speed are piecewise constant and change to the new values only at the end of distance intervals of length Δs . Thus, spatial discretization of the road grade profile is employed consistent with Chapter 2, i.e.,

$$\bar{x}(s) = \frac{x(s) + x(s + \Delta s)}{2}, \quad (4.7)$$

for $s_k \leq s \leq s_{k+1}$, $s_{k+1} = s_k + \Delta s$, where x is the continuous profile and \bar{x} is the piecewise constant profile.

The second approximation is that the constant values of road grade or vehicle speed within each distance segment belong to a given set of discrete values of road grade, Θ , or vehicle speed, V . The constant grade values, $\bar{\theta}$, over distance segments, Δs , are mapped to discrete values $\theta^* \in \Theta$ using

$$\theta^* \in \arg \min_{\hat{\theta} \in \Theta} |\bar{\theta} - \hat{\theta}|, \quad (4.8)$$

where θ^* is the selected discrete value in Θ that is closest $\bar{\theta}$. Similarly, the constant vehicle speed values are mapped by

$$v^* \in \arg \min_{\hat{v} \in V} |\bar{v} - \hat{v}|. \quad (4.9)$$

Based on the piecewise-constant road grade and vehicle speed profiles and using (2.6), Markov chains are generated from the data and the similarity of these chains is assessed using the symmetric KL divergence rate (4.6), see Figures 4.1 through 4.5. In these figures, Markov chains from individual drives are compared to Markov chains that are representative of a particular aggregate characteristic (i.e., highway driving, rush hour traffic, city terrain, etc). These aggregate Markov chains are generated from selected data from the appropriate data collection drives. Table 4.1 provides the description of each of the data collection drives. The results presented are for $\Delta s = 30$ m, $\Theta = \{-6, -5, \dots, 6\}$ % and $V = \{1, 2, \dots, 36\}$ m/s.

In the following, P is used to denote a Markov chain used for road grade prediction while PV is used to denote a Markov chain used for traffic speed prediction.

Figure 4.1 shows the KL divergence rate between each of the road grade Markov chains for the data collection runs and the city, P_C , and highway, P_{HW} , aggregate Markov chains. Note that the highway Markov chains are more similar to the highway aggregate and the city Markov chains are more similar to the city aggregate (indicated by lower values of the KL divergence rate). This figure demonstrates that the KL divergence rate can be used to distinguish between and categorize different grade profiles.

Figures 4.2 and 4.3 present the KL divergence rate between traffic speed Markov chains of individual drives and aggregate Markov chains for rush hour, PV_{RH} , and off peak, PV_{OP} , driving times. Figure 4.2 shows the results for highway driving and indicates that there is not a strong ability to differentiate between traffic speed patterns based on KL divergence rate for highway drives. However, looking at Figure 4.3, which presents the results for city driving, a stronger ability to differentiate between rush hour and off-peak driving based on KL divergence rate is observed.

Figures 4.4 and 4.5 show the KL divergence rate between Markov chains from individual drives and aggregate Markov chains for different driving styles of active, PV_{Ac} , and passive PV_{Pa} driving. For both city and highway driving, as can be observed from these figures, the KL divergence rate provides a good differentiation between different driving styles. This indicates that this method can be used to distinguish between different driving styles. This also indicates that it may be more useful to predict the behavior of a single vehicle based on the driver's behavior as opposed to the flow of traffic as a whole based on the time of day.

4.2 Stochastic jump process for road grade prediction

This section introduces a novel technique for predicting road grade through the use of a stochastic jump process which models stochastically the distance over which a step change in the road grade occurs. The development of a new model is motivated by the observation that the Markov chains generated for road grade prediction are nearly diagonal. This has a rate limiting effect (the next value of the road grade cannot be too far from the current value). Though the diagonal Markov chains may reflect the statistical properties of the road grade transitions, the predicted grade profile may lag the actual road geometry.

The stochastic jump model prescribes a probability that a step change in road grade with a magnitude of $\Delta\theta$ occurs within a given distance interval. To develop the model, a uniform mesh of discrete distance values, $s = \{s_1, s_2, \dots, s_m\}$, $s_{k+1} > s_k$, is first chosen.

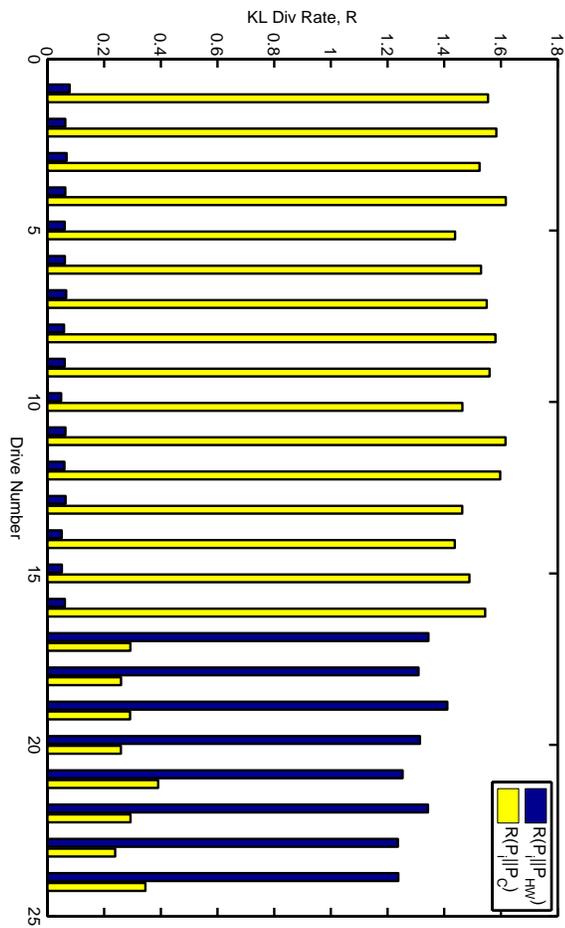


Figure 4.1: KL divergence rate between road grade Markov chains and aggregate road grade Markov chains for highway and city driving.

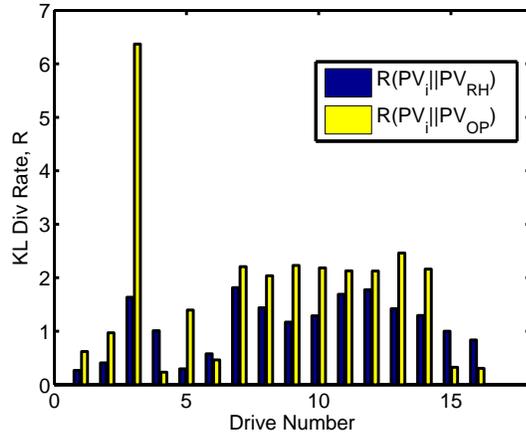


Figure 4.2: KL divergence rate between traffic speed Markov chains of individual drives and highway aggregate traffic speed Markov chains for off-peak and rush hour traffic times.

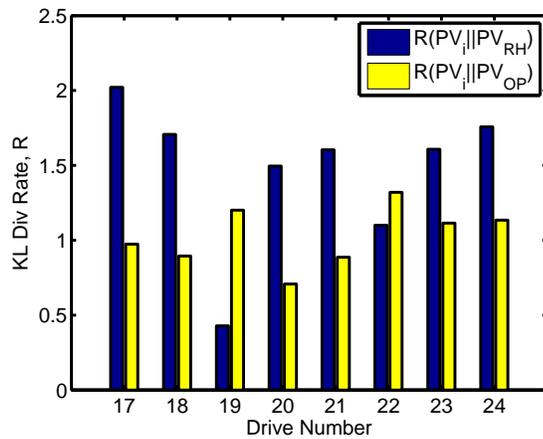


Figure 4.3: KL divergence rate between traffic speed Markov chains of individual drives and city aggregate traffic speed Markov chains for off-peak and rush hour traffic times.

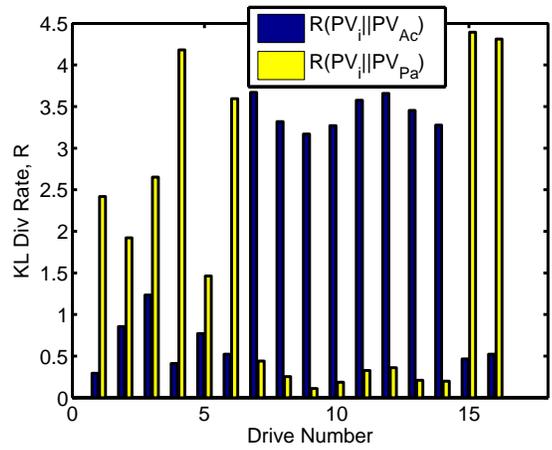


Figure 4.4: KL divergence rate between traffic speed Markov chains of individual drives and highway aggregate Markov chains for active and passive driving styles.

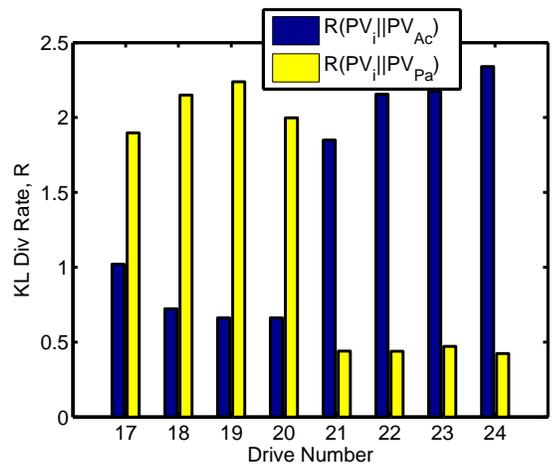


Figure 4.5: KL divergence rate between traffic speed Markov chains of individual drives and city aggregate Markov chains for active and passive driving styles.

Table 4.1: Time of Day, Type of Driving, and Location

Drive Number	Location	Time of Day	Type
1	Highway	Rush Hour	Active
2	Highway	Rush Hour	Active
3	Highway	Off Peak	Active
4	Highway	Off Peak	Active
5	Highway	Rush Hour	Active
6	Highway	Rush Hour	Active
7	Highway	Off Peak	Passive
8	Highway	Off Peak	Passive
9	Highway	Rush Hour	Passive
10	Highway	Rush Hour	Passive
11	Highway	Off Peak	Passive
12	Highway	Off Peak	Passive
13	Highway	Rush Hour	Passive
14	Highway	Rush Hour	Passive
15	Highway	Off Peak	Active
16	Highway	Off Peak	Active
17	City	Off Peak	Active
18	City	Off Peak	Active
19	City	Rush Hour	Active
20	City	Off Peak	Active
21	City	Off Peak	Passive
22	City	Rush Hour	Passive
23	City	Off Peak	Passive
24	City	Off Peak	Passive

Then the road grade data are segmented into distance segments $\rho_1, \rho_2, \dots, \rho_n$. The average grade is found within these segments and then using the quantized set of grade values Θ and (4.8), the average grade values are replaced with the closest value, $\theta^* \in \Theta$. These approximations produce a piecewise constant grade profile approximation which consists of discrete step changes in the road grade.

4.2.1 Stochastic jump model

The stochastic jump process is modeled as two jump models [101],

$$\begin{aligned} \mathbb{P}^+(s = s_j | \theta_0), \\ \mathbb{P}^-(s = s_j | \theta_0), \end{aligned} \tag{4.10}$$

where $s_j \in S = \{s_1, s_2, \dots, s_m\}$ is a discrete set of distances, $\mathbb{P}^+(s = s_j | \theta_0)$ is the probability that the first jump of $+\Delta\theta$ in road grade occurs at a distance $s = s_j$ from the position of last change, ρ , for an initial grade θ_0 , and $\mathbb{P}^-(s = s_j | \theta_0)$ is the probability that the first jump of $-\Delta\theta$ in road grade occurs at a distance $s = s_j$ from the position of last change, ρ , for an initial grade θ_0 . The position of last change, ρ , is used to indicate from where the measuring of the distances will start. It is referred to as the position of last change because it is the last position that a step change in road grade was observed.

It is important note that the road grade data used in this section and the following sections is based upon effective road grade, calculated from changes engine load, as opposed to road grade calculated from changes in altitude, which was used in the previous sections. As such, the effective road grade (seen in Figure 4.9) is lower in magnitude and thus $\Theta = \{-3, -2.5, \dots, 3\}\%$ in this and the following sections with $\Delta\theta = 0.5\%$. While this is an important change to note, it does not inherently change the value of the results.

4.2.2 Stochastic jump model identification

With the continuous grade versus distance profile mapped into the discrete states, the distances between step changes are counted and the frequencies of the transitions of $\pm\Delta\theta$ are determined with an individual model for each direction. The probability that at a given initial grade, θ_k , and a certain distance, $s_j^+ \in S$, a positive transition will occur is

$$\mathbb{P}^+(s = s_j^+ | \theta_k) = \frac{N_{k,s_j^+}^+}{M_k^+}, \quad (4.11)$$

and given a certain distance, $s_j^- \in S$, a negative transition will occur is

$$\mathbb{P}^-(s = s_j^- | \theta_k) = \frac{N_{k,s_j^-}^-}{M_k^-}, \quad (4.12)$$

where $N_{k,s_j^+}^+$ and $N_{k,s_j^-}^-$ are, respectively, the number of positive and negative step changes in the road grade observed at a specific distance $s_j^+ \in S$, $s_j^- \in S$ for a given grade value $\theta_k \in \Theta$ observed in a grade profile and M_k^+ and M_k^- are, respectively, the total number of positive and negative step changes for all values in S for a given grade value $\theta_k \in \Theta$ observed in a grade profile. See Figure 4.6 for an example. In this Figure, the black dotted line is the continuous grade profile, the solid blue line is the piecewise constant profile, and the important values are indicated with labeled dashed red lines. The value ρ is the position of last change from which the distance measurements s_j^+ and s_j^- are being made.

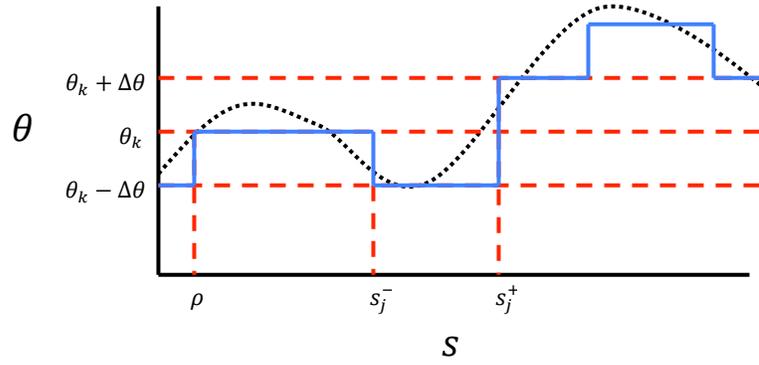


Figure 4.6: Example of the stochastic jump process identification procedure.

Because the jump process is modeled using two stochastic jump processes, two probability distributions exist for each value of road grade (an up and a down distribution). However, there are two exceptions to this. For $\theta_k = \min \Theta$ there is no transition probability distribution for a negative step change and for $\theta_k = \max \Theta$ there is not distribution for a positive step change. This is to ensure $\theta^+ \in \Theta$.

Figures 4.7 and 4.8 provide examples of the cumulative distribution functions for the positive step change and negative step change, respectively, jump processes for an initial grade of 0% with $S = \{0, 5, \dots, 50\}$ m and $\Delta\theta = 0.5\%$. Note that these cumulative distribution functions are similar to the cumulative distribution functions for Poisson processes.

4.2.3 Prediction using the stochastic jump process

The stochastic jump processes are used for predicting the road grade in the following way. Given a current position ρ with an initial road grade θ , the next position, ρ^+ , and next grade θ^+ are defined as

$$\begin{aligned} \rho^+ &= \rho + f(\theta), \\ \theta^+ &= \begin{cases} \theta + \Delta\theta & \text{if } \bar{s}^+(\theta) < \bar{s}^-(\theta), \\ \theta - \Delta\theta & \text{if } \bar{s}^-(\theta) \leq \bar{s}^+(\theta), \end{cases} \end{aligned} \quad (4.13)$$

where

$$f(\theta) = \min\{\bar{s}^+(\theta), \bar{s}^-(\theta)\}, \quad (4.14)$$

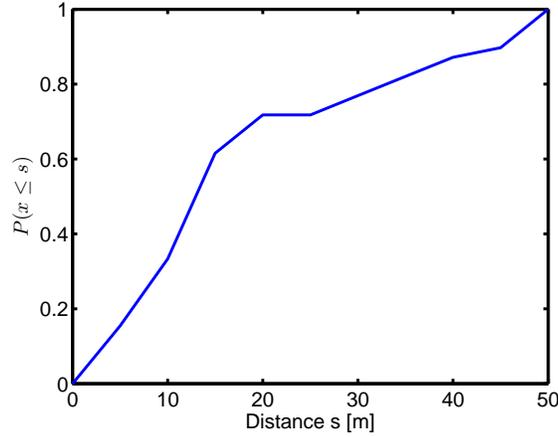


Figure 4.7: An example of the data generated jump process for a positive step in grade with an initial grade $\theta = 0\%$

and

$$\begin{aligned}\bar{s}^+(\theta) &= \mathbb{E}[s^+ | \theta], \\ \bar{s}^-(\theta) &= \mathbb{E}[s^- | \theta],\end{aligned}\tag{4.15}$$

where $\mathbb{E}[s^+ | \theta]$ and $\mathbb{E}[s^- | \theta]$ are the expected distances over which a positive or negative step change, respectively, will occur. This is repeated until a predicted grade profile exists over the desired prediction horizon, Δs .

4.3 Grade prediction comparison

In the following section, a comparison of prediction accuracies of the stochastic jump process and the Markov chain method will be presented. The example data used was collected on a 21 km stretch of northbound M-39 with the test vehicle traveling at an average speed of 88 KPH. Figure 4.9 shows this grade data for this example that was used to generate both the Markov chain and jump process distributions.

The Markov grade prediction is generated as expected value of next grade given the current grade for the Markov chain,

$$\theta^+ = \mathbb{E}[\theta^+ | \theta].\tag{4.16}$$

This value of θ^+ is then constant over the entire prediction horizon Δs .

The prediction accuracy is assessed by calculating the mean absolute error (MAE) be-

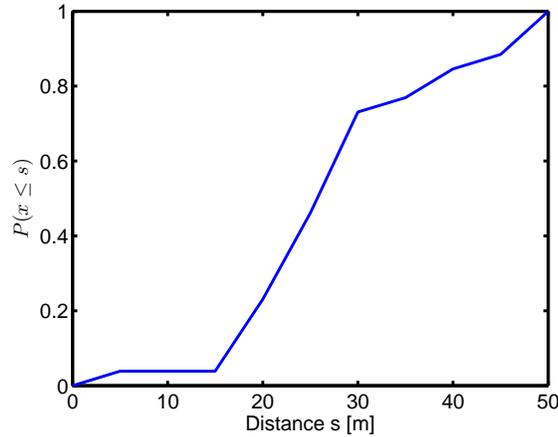


Figure 4.8: An example of the data generated jump process for a negative step in grade with an initial grade $\theta = 0\%$

tween each of the prediction methods and the real data for different values of the prediction horizon $\Delta s \in \{10, 15, \dots, 50\}$ m with $\Delta\theta = 0.5\%$ for the stochastic jump process method. Figure 4.10 provides example profiles of these prediction methods for $\Delta s = 15$ m overlaid with real data. Figure 4.11 shows the accuracies of each of the prediction methods. The jump process is more accurate up to 20 m, however, after this, the Markov chain method is more accurate.

4.4 Online learning of traffic patterns and road grade

The previous sections use off-line learning to generate the stochastic models. The online learning of Markov chain models is also of interest (see [38, 47]), as well as the capability of performing quick comparisons between Markov chain models using the KL divergence to determine if a new model needs to be learned or if the current model is sufficient, see [47, 48]. This section will briefly discuss how the online learning of Markov chain models is performed and then will introduce a similar learning method and convergence analysis to be applied to the online learning of the stochastic jump models.

During off-line learning of Markov chain models, frequencies of transitions from a given state, x_i , to all other possible states, x_j , are counted and transition probabilities are estimated. This process, based on equation (2.6), is modified to an online version using the following observations.

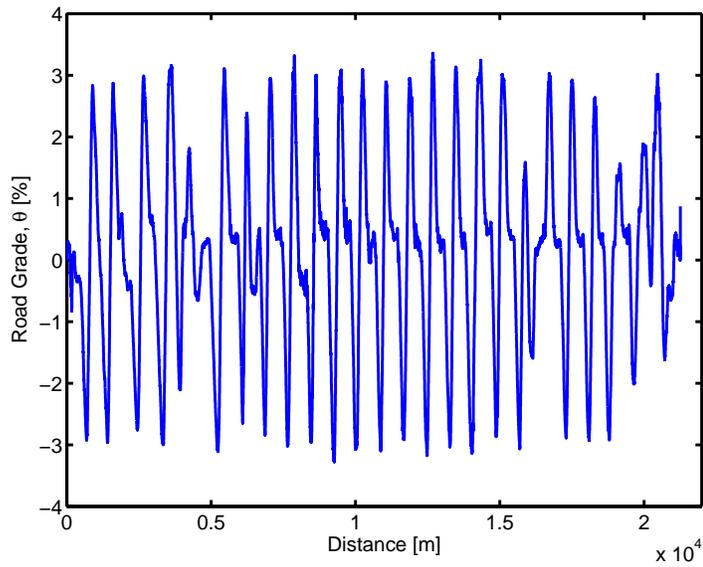


Figure 4.9: The grade data used to generate the stochastic grade prediction example

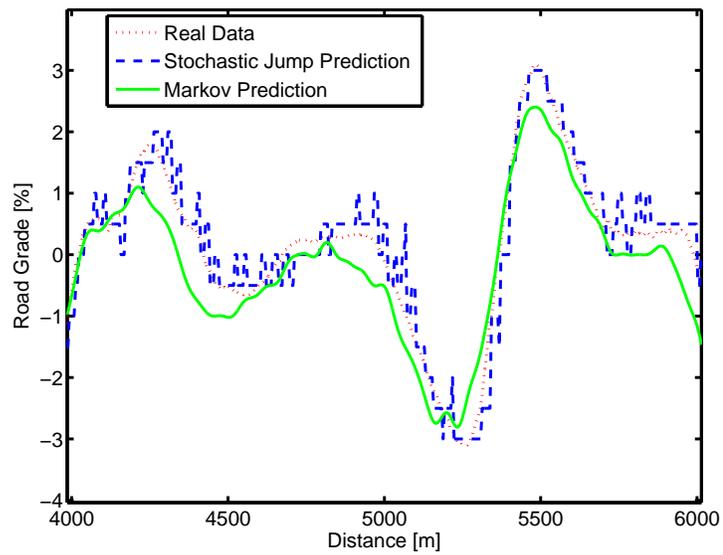


Figure 4.10: The real road grade data overlaid with the Markov and jump predictions. Note that neither prediction produces gross deviations from the real data

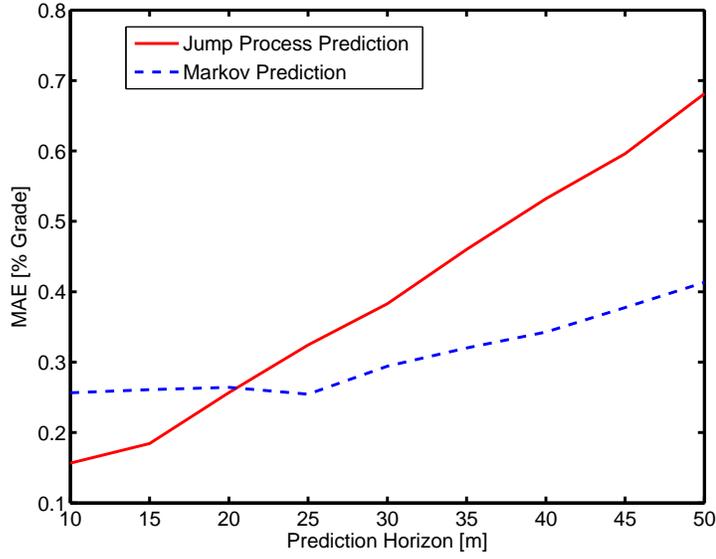


Figure 4.11: An example of the data generated stochastic jump process for a down step in grade with an initial grade $\theta = 0\%$

Equation (2.6) can be rewritten as

$$T(x_j | x_i) = \frac{N_{x_i, x_j}(k)/k}{M_{x_i}(k)/k} = \frac{F_{x_i, x_j}(k)}{F_{0, x_i}(k)}, \quad (4.17)$$

where $F_{x_i, x_j}(k)$ is the mean frequency of transition events from state x_i to state x_j at an instant k and $F_{0, x_i}(k)$ is the mean frequency of transition events out of state x_i at time k . If k is set equal to the time it takes to collect a set of data, or equivalently the number of discrete samples collected in a set of data, then (4.17) reduces to (2.6).

Define a transition event flag from state x_i to state x_j at time k to be $f_{x_i, x_j}(k)$ (i.e., $f_{x_i, x_j}(k) = 1$ if the event occurs at the time instance k and $f_{x_i, x_j}(k) = 0$ otherwise) and the transition event flag out of state x_i at time k to be $f_{0, x_i}(k)$. Then,

$$F_{x_i, x_j}(k) = \frac{1}{k} \sum_{t=1}^k f_{x_i, x_j}(t), \quad (4.18)$$

$$F_{0, x_i}(k) = \frac{1}{k} \sum_{t=1}^k f_{0, x_i}(t).$$

These equations can be rewritten in a form suitable for online learning as,

$$\begin{aligned} F_{x_i, x_j}(k) &= (1 - \phi)F_{x_i, x_j}(k - 1) + \phi(f_{x_i, x_j}(k)), \\ F_{0, x_i}(k) &= (1 - \phi)F_{0, x_i}(k - 1) + \phi(f_{0, x_i}(k)). \end{aligned} \quad (4.19)$$

In (4.19), $\phi = \frac{1}{k}$. Following [38], ϕ can be modified to a constant value, $\phi \in (0, 1]$, which represents a constant forgetting factor thereby emphasizing newer data over older data.

Using (4.17)-(4.19), a Markov chain model can be learned online as data are collected, see [38]. To test whether or not the learned Markov chain model has converged, the KL divergence as defined for Markov chains, (4.1), is used.

Reference [47] demonstrates this procedure online in vehicle tests. This work also shows that similarly, comparing current models with previous models, in conjunction with an appropriate forgetting factor ϕ , a system implementing this learning strategy can automatically distinguish between and learn separate models when appropriate.

Like the learning of Markov chains, jump processes can also be learned online. Equations (4.11) and (4.12) similarly can be rewritten recursively and applied to online learning. The recursive form of (4.11) is

$$\begin{aligned} F_{\theta(k), s_j}^+(k) &= (1 - \phi)F_{\theta(k), s_j}^+(k - 1) + \phi(f_{\theta(k), s_j}^+(k)), \\ F_{0, \theta(k)}^+(k) &= (1 - \phi)F_{0, \theta(k)}^+(k - 1) + \phi(f_{0, \theta(k)}^+(k)). \end{aligned} \quad (4.20)$$

Above, $F_{\theta(k), s_j}^+(k)$ is the mean frequency of positive step change from $\theta(k)$ at distance s_j , $f_{\theta(k), s_j}^+(k)$ is the event flag of a positive step change from $\theta(k)$ at distance s_j , $F_{0, \theta(k)}^+(k)$ is the mean frequency of positive step changes from $\theta(k)$, and $f_{0, \theta(k)}^+(k)$ is the event flag of a positive step change in grade from $\theta(k)$. As before, if a positive step change from $\theta(k)$ at a distance of s_j occurs at time (k) , then $f_{\theta(k), s_j}^+(k) = 1$ and if a positive step change from $\theta(k)$ occurs at time k , then $f_{0, \theta(k)}^+(k) = 1$. The similar expression for a negative step change in grade is

$$F_{\theta(k), s_j}^-(k) = (1 - \phi)F_{\theta(k), s_j}^-(k - 1) + \phi(f_{\theta(k), s_j}^-(k)), \quad (4.21)$$

$$F_{0, \theta(k)}^-(k) = (1 - \phi)F_{0, \theta(k)}^-(k - 1) + \phi(f_{0, \theta(k)}^-(k)). \quad (4.22)$$

As more data are used to learn the Markov chain and stochastic jump models online, the frequencies of transitions converge to their true values (i.e., the values found from off-line learning). To test this convergence, the current learned model can be compared with an older learned model from Δk steps in the past via the symmetric KL divergence (4.3). If the models are suitably similar, that is the KL divergence is below a selected threshold,

then the models have not changed substantially over the Δk horizon and can be said to have converged [47].

For Markov chain model learning, convergence is said to occur when

$$D_{KL,S}^M(P(k) \parallel P(k - \Delta k)) \leq \gamma_M, \quad (4.23)$$

for $\gamma_M \geq 0$ with $P(k)$ denoting the current learned Markov chain model at time k and $P(k - \Delta k)$ denoting the learned Markov chain model Δk steps in the past.

To calculate the KL divergence between two jump distributions, P and Q , the KL divergence for Poisson distributions is used,

$$D_{KL}^P(P \parallel Q) = \lambda_Q - \lambda_P + \lambda_P \log \left(\frac{\lambda_P}{\lambda_Q} \right), \quad (4.24)$$

where λ_P and λ_Q are the expectations of distribution P and distribution Q respectively.

The Poisson process KL divergence still possesses the same properties as the Markov chain KL divergence. For the purpose of comparing the Markov chain model learning and the stochastic jump process learning, we define a summed symmetric Poisson process KL divergence as

$$D_{KL,S}^P(k, \Delta k) = \sum_{i=1}^{2(N_\theta-1)} \frac{1}{2} (D_{KL}^P(P_i(k) \parallel P_i(k - \Delta k)) + D_{KL}^P(P_i(k - \Delta k) \parallel P_i(k))), \quad (4.25)$$

with a convergence criterion defined as

$$D_{KL,S}^P(k, \Delta k) \leq \gamma_P, \quad (4.26)$$

where $\gamma_P \geq 0$, $P_i(k)$ is the current i^{th} learned stochastic jump model, and $P_i(k - \Delta k)$ is the i^{th} learned stochastic jump model Δk steps in the past. The thresholds γ_M and γ_P are selected based on the desired sensitivity of the learning convergence [47].

Because there are $2(N_\theta - 1)$ stochastic jump distributions (N_θ is the number of elements in Θ , 2 because there exists a set for both up and down transitions, and subtract one from each of the up and down transitions because the transitions are limited such that the road grade profile is constrained to be piecewise in Θ) the summed symmetric KL divergence presented for the jump distributions is the sum of all $2(N_\theta - 1)$ divergences between each individual current distribution and its Δk -steps-in-the-past distribution. This is done in order to produce a single value that can be easily plotted. It should be noted that this value will still approach zero as all $2(N_\theta - 1)$ distributions converge.

The results summarizing the online learning capabilities of Markov chain and stochastic jump process models will now be presented. Figures 4.12 and 4.13 present examples of the probability distributions learned recursively over one data set using the above methods, Figure 4.12 is the learned Markov chain model and Figure 4.13 is the learned jump process for a positive step change from 0% road grade. These Figures present the distributions at the end of the data set with all points used in the learning. Note the similarity between Figures 4.12 and 4.13 and Figures 2.2 and 4.7.

Figure 4.14 shows how the KL divergence of the two models decreases over the course of the learning cycle with $\Delta k = 20$ steps. Note, the section before 5000 m in Figure 4.14 is where the bulk of the learning occurs, demonstrated by the many changes in the value of KL divergence for both models. Also note that though both are plotting KL divergence, the value of the KL divergence for the jump process has much larger values in the initial learning phase than the Markov models. This is not as meaningful as the fact that both values being plotted tend to zero as more data are used for learning.

Figure 4.15 shows the mean absolute error from the predictions made while learning the Markov chain and jump processes online. The predictions near the beginning of the learning cycle are poor, but as learning continues, the predictions become more accurate.

This section demonstrates that both the grade predicting Markov chains and jump processes can be learned online and used throughout the learning cycle. As to be expected, the accuracy of the models is low at the beginning of the learning cycle, but improves throughout the learning cycle. It should be noted that the online learning of Markov chain models was successfully demonstrated in an experimental vehicle in [47]. Future work will be to implement the learning of the stochastic jump process on board an experimental vehicle and to also apply the stochastic jump process modeling to the prediction of traffic speed.

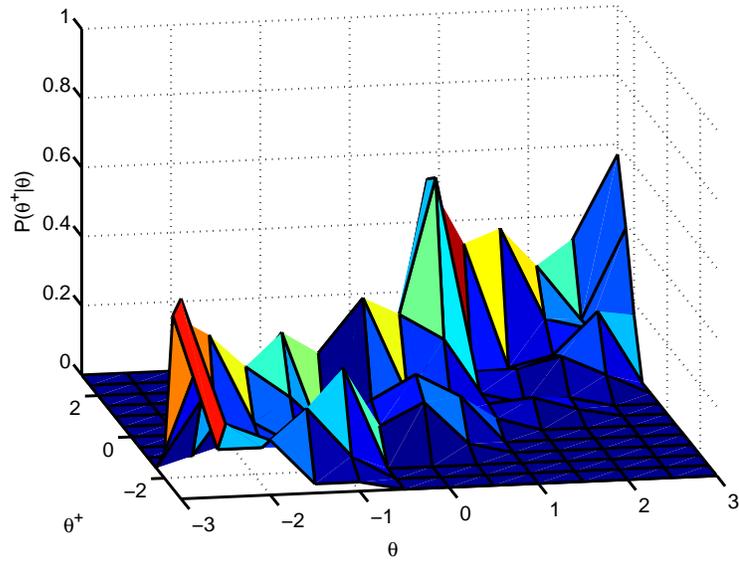


Figure 4.12: Surface plot of an online learned Markov chain model. Note the similarities between this and Figure 2.2

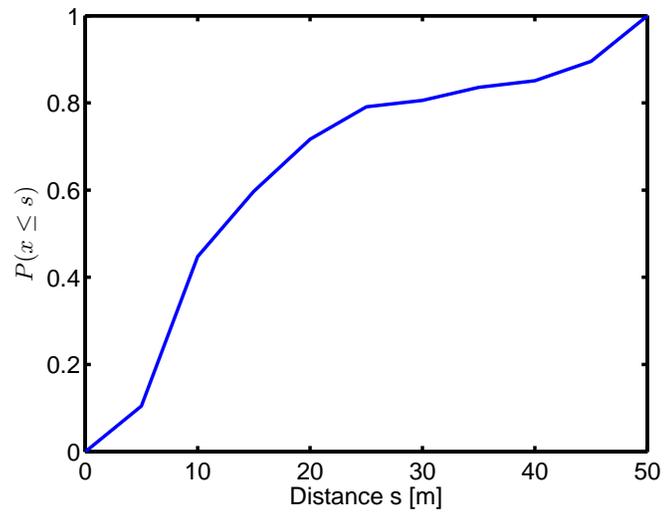


Figure 4.13: An online learned stochastic jump process for a positive step change from 0% road grade. Note the similarities between this and Figure 4.7.

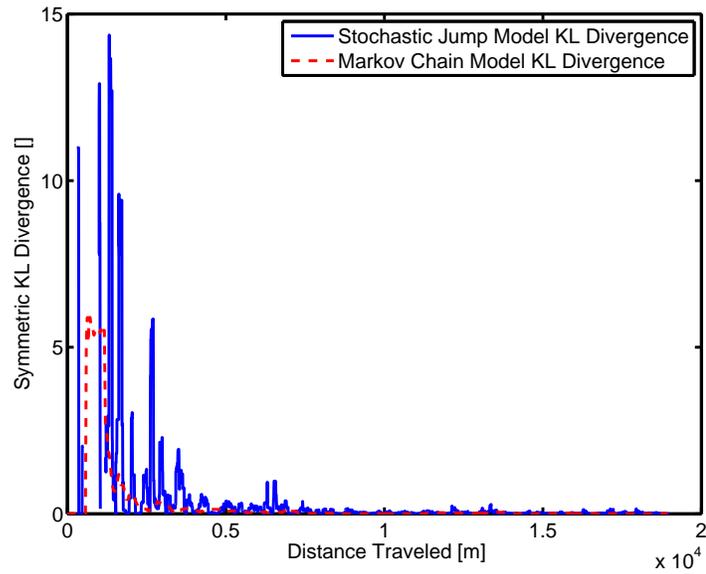


Figure 4.14: The trend of the KL divergence for both the Markov chain and stochastic jump processes throughout the learning cycle. As expected, the KL divergence tends to decrease and approaches a small value near zero—this is indicative of model convergence.

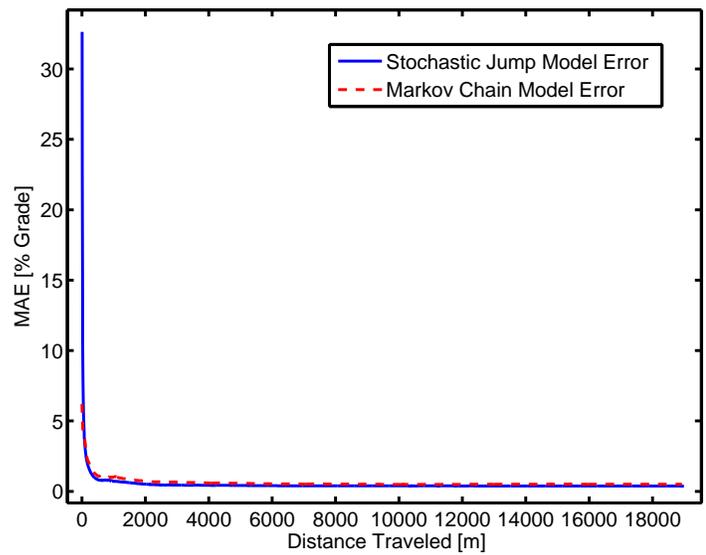


Figure 4.15: The trend of the mean absolute error between the predicted and actual road grade. Note how the mean absolute error decreases over the learning cycle.

CHAPTER 5

Safe, Recoverable, and Integral Safe Sets with Aircraft Applications

This chapter discusses safe sets and introduces recoverable and integral safe sets for linear, discrete-time systems in Sections 5.1-5.3. A comparison of these three sets for linearized aircraft lateral and longitudinal dynamics is presented in Section 5.4. The application of these sets to aircraft maneuver planning is described in Section 5.5. Further examples of recoverable sets for different choices of auxiliary subsystem, in cases of additive set-bounded disturbances, and at different flight conditions are considered in Section 5.6.

We consider the following linear discrete-time open-loop system,

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k), \\y(k) &= Cx(k) + Du(k),\end{aligned}\tag{5.1}$$

with imposed output constraints,

$$Gy(k) \leq g.\tag{5.2}$$

Define the output constraint set, Y , as

$$Y = \{y : Gy \leq g\}.\tag{5.3}$$

Note that (5.1)-(5.3) can represent a discrete-time model of open-loop linearized aircraft dynamics. The output constraints, (5.2), in that case, are generated from consideration of the flight envelope, the control constraints, and the assumptions made for the region of validity of the linearized model.

5.1 Safe set

Consider the open-loop system (5.1) with output constraints (5.2). Allow the loop to be closed with a stabilizing feedback control of the form,

$$u(k) = -Kx(k), \quad (5.4)$$

such that

$$\begin{aligned} x(k+1) &= A_{CL}x(k), \\ y(k) &= C_{CL}x(k), \end{aligned} \quad (5.5)$$

where $A_{CL} = A - BK$ and $C_{CL} = C - DK$, is now the closed-loop system. In practice, the controller can be any stabilizing feedback controller, however, in this dissertation an LQR control architecture will be exclusively used. The safe set, O_∞ , for this system [58] is then defined as the set of all initial conditions such that the ensuing closed-loop trajectories are output constraint admissible. That is,

$$O_\infty = \{x(0) \in \mathbb{R}^n : y(k) = C_{CL}A_{CL}^k x(0) \in Y, \forall k \geq 0\}. \quad (5.6)$$

If A_{CL} is Schur, (C_{CL}, A_{CL}) is observable, $0 \in \text{int } Y$, and Y is compact, then O_∞ is positively-invariant, finitely-determined, and is a polytope (bounded polyhedron) [58]. Finite determination means that there exists t^* such that $O_\infty = O_t$ for all $t \geq t^*$, where

$$O_t = \{x(0) \in \mathbb{R}^n : y(k) = C_{CL}A_{CL}^k x(0) \in Y, \text{ for } k = 0, \dots, t\}, \quad (5.7)$$

or, equivalently,

$$O_\infty = O_{t^*} = \{x(0) \in \mathbb{R}^n : GC_{CL}A_{CL}^k x(0) \leq g, \text{ for } k = 0, \dots, t^*\}. \quad (5.8)$$

In practice, t^* is usually comparable to the settling time of the closed-loop system. Note also that not all the constraints forming O_∞ in (5.8) are necessary as many are redundant and can be eliminated. The advantage of eliminating the redundant constraints is that the storage memory requirements are reduced, and subsequent computations involving O_∞ are simplified. Simpler subsets that closely approximate O_∞ can be generated by also eliminating the almost redundant constraints and applying a pull-in transformation [102]. For the generalizations of the safe sets to systems with additive set-bounded disturbances, see reference [79] and references therein and for systems with polytopic uncertainty, see

reference [83] and references therein. For nonlinear systems, see [103] and references therein.

The computational algorithm used to calculate safe sets in this dissertation is adapted from [83]. See Appendix B for a description. This algorithm has been implemented in MATLAB.

Note that O_∞ is dependent on the choice of the nominal controller (5.4).

5.2 Recoverable set

We note that $x(0) \notin O_\infty$ may be recoverable (i.e., the closed-loop response from $x(0)$ can be made to satisfy constraints) if (5.4) is modified. Hence, consider now the controller (5.4) augmented with an external signal, $v(k)$,

$$u(k) = -Kx(k) + v(k). \quad (5.9)$$

This leads to the closed-loop system of the form

$$\begin{aligned} x(k+1) &= A_{CL}x(k) + Bv(k), \\ y(k) &= C_{CL}x(k) + Dv(k). \end{aligned} \quad (5.10)$$

We refer to $v(k)$ as the recovery sequence.

Define the viability set, V_∞ [71, 104], for (5.10) with constraints (5.2) as the set of all initial conditions, $x(0)$, for which there exists a sequence $v(\cdot)$ that enforces the constraints, i.e.,

$$V_\infty = \{x(0) : \exists v(\cdot) \text{ such that } y(k) \in Y \forall k \geq 0\}. \quad (5.11)$$

Clearly, $O_\infty \subset V_\infty$, the sequence $v(\cdot)$ is specific to each initial condition, $x(0)$, and V_∞ can be much larger than O_∞ . Unfortunately computing V_∞ can be very involved which motivates the approximation we now discuss.

To develop an inner approximation to V_∞ , we assume that the recovery sequence is generated by an auxiliary system of the form,

$$\begin{aligned} v(k) &= \hat{c} + \hat{C}\hat{x}(k), \\ \hat{x}(k+1) &= \hat{A}\hat{x}(k), \end{aligned} \quad (5.12)$$

where \hat{A} is a Schur (asymptotically stable) matrix and \hat{x} is the \hat{n} -vector state of the auxiliary system. A version of the safe set for (5.10) and (5.12) with constraints (5.2) is referred to

as O_∞^{ext} and is defined as,

$$O_\infty^{ext} = \{(x(0), \hat{c}, \hat{x}(0)) \in \mathbb{R}^{n+p+\hat{n}} : y(k) \in Y, \forall k \geq 0\}, \quad (5.13)$$

where n is the dimension of x , p is the dimension of \hat{c} and u , and \hat{n} is the dimension of \hat{x} . The recoverable set, R_∞ , is defined as a projection of O_∞^{ext} on the plant coordinates,

$$R_\infty = \{x(0) \in \mathbb{R}^n : \exists \hat{c}, \hat{x}(0) \text{ such that } (x(0), \hat{c}, \hat{x}(0)) \in O_\infty^{ext}\}. \quad (5.14)$$

Note that since the constraints (5.2) are affine, both O_∞^{ext} and R_∞ are polyhedral. We note that if O_∞^{ext} fails to be finitely-determined an arbitrarily close inner approximation of O_∞^{ext} which is positively-invariant can be shown to exist [79]. We assume O_∞^{ext} is finitely-determined in the subsequent discussions noting that it can be replaced by a finitely-determined approximation if not. Note also that the recoverable set (5.14) is positively-invariant for (5.10) and (5.12).

From the computational standpoint, generating O_∞^{ext} and R_∞ can be accomplished using the conventional safe set computations and projection methods, Appendix B, for which tools and software already exist.

Given $x(0) \in R_\infty$, computing an appropriate \hat{c} and $\hat{x}(0)$ for which constraints are enforced reduces to a problem of the type,

$$\begin{aligned} \frac{1}{2}\hat{c}^T\hat{c} + \frac{1}{2}\hat{x}(0)^T P \hat{x}(0) \rightarrow \min_{\hat{c}, \hat{x}(0)}, \\ \text{subject to } (x(0), \hat{c}, \hat{x}(0)) \in O_\infty^{ext}, \end{aligned} \quad (5.15)$$

where $P = P^T > 0$. The minimization in (5.15) aims at preserving the operation with the nominal controller and avoiding the unnecessary use of recovery sequence. Computationally, (5.15) is a quadratic programming problem. By using 1 or ∞ norms in (5.15) instead of quadratic penalties, a closely related linear programming formulation can be obtained. Once $\hat{x}(0)$ and \hat{c} are determined, the execution of (5.12) yields a recovery sequence for a given $x(0)$.

Several choices exist for the selection of the auxiliary dynamics (5.12), including the shift register and the Laguerre's sequence generators. These choices are motivated by related developments in the extended command governor case [105] and in the model-predictive control case [106]. When utilizing the shift register, the recovery sequence be-

comes equal to \hat{c} after $H + 1$ steps and \hat{A} and \hat{C} have the form

$$\hat{A} = \begin{bmatrix} \bar{S}_1 & 0 & \cdots & 0 \\ 0 & \bar{S}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \bar{S}_p \end{bmatrix}, \quad \hat{C} = \begin{bmatrix} \bar{T}_1 & 0 & \cdots & 0 \\ 0 & \bar{T}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \bar{T}_p \end{bmatrix}, \quad (5.16)$$

with

$$\bar{S}_i = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 1 \\ 0 & 0 & \cdots & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{(H+1) \times (H+1)}, \quad \bar{T}_i = [1 \ 0 \ \cdots \ \cdots \ 0] \in \mathbb{R}^{1 \times (H+1)}, \quad (5.17)$$

with the number of blocks in \hat{A} and \hat{C} being equal to the number of control channels being utilized.

When using Laguerre's sequence generators, \hat{A} and \hat{C} have the form (5.16) and \bar{S}_i and \bar{T}_i have the following structure,

$$\bar{S}_i = \begin{bmatrix} \alpha_L & \beta & -\alpha_L \beta_L & \alpha_L^2 \beta_L & \cdots \\ 0 & \alpha_L & \beta_L & -\alpha_L \beta_L & \cdots \\ 0 & 0 & \alpha_L & \beta_L & \cdots \\ 0 & 0 & 0 & \alpha_L & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}, \quad \bar{T}_i = \sqrt{\beta_L} [1 \ -\alpha_L \ \alpha_L^2 \ -\alpha_L^3 \ \cdots], \quad (5.18)$$

where $\beta_L = 1 - \alpha_L^2$, $0 \leq \alpha_L \leq 1$ are selectable parameters. The choice $\alpha_L = 0$ corresponds to the shift register. Numerical examples suggest that low dimensional Laguerre's sequence generators for $\alpha_L \neq 0$ are capable of producing a rich set of recovery sequences.

To incorporate the auxiliary dynamics with the nominal closed-loop dynamics, note that

$$u(k) = -Kx(k) + v = -Kx(k) + \hat{c} + \hat{C}\bar{x}(k),$$

where K is the stabilizing feedback gain and thus,

$$u(k) = \begin{bmatrix} -K & I & \hat{C} \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{c} \\ \hat{x}(k) \end{bmatrix}. \quad (5.19)$$

Therefore, when one constructs O_∞^{ext} with the inclusion of a nominal controller, the state vector becomes

$$\begin{bmatrix} x(k) \\ \hat{c} \\ \hat{x}(k) \end{bmatrix},$$

with

$$A_{ext} = \begin{bmatrix} A & B & B\hat{C} \\ 0 & I & 0 \\ 0 & 0 & \hat{A} \end{bmatrix}, \quad C_{ext} = \begin{bmatrix} C & D & D\hat{C} \\ -K & I & \hat{C} \end{bmatrix}, \quad (5.20)$$

used in place of the nominal A and C . The second row of C_{ext} is introduced to handle the constraints imposed on the control.

5.3 Integral safe set

Practical controllers frequently employ integral action to guarantee offset-free tracking of constant reference commands and asymptotic rejection of constant input disturbances. Suppose the closed-loop system has the following form,

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ x_I(k+1) &= C_I x(k) + x_I(k), \end{aligned} \quad (5.21)$$

where

$$u(k) = -K_1 x(k) - K_2 x_I(k). \quad (5.22)$$

The controller with an integral action can be designed using a variety of techniques such as LQ-I and will be assumed to give an asymptotically stable closed-loop system of the following form,

$$\begin{aligned} \bar{x}(k+1) &= \bar{A}\bar{x}(k), \\ \bar{y}(k) &= \bar{C}\bar{x}(k), \end{aligned} \quad (5.23)$$

where

$$\bar{x}(k) = \begin{bmatrix} x(k) \\ x_I(k) \end{bmatrix}, \quad (5.24)$$

with

$$\bar{A} = \begin{bmatrix} A - BK_1 & -BK_2 \\ C_I & I \end{bmatrix} \quad (5.25)$$

and \bar{C} appropriately defined for the output.

The constraints have the form $\bar{y}(k) \in \bar{Y}$ where

$$\bar{Y} = \{\bar{y} : \bar{G}\bar{y} \leq \bar{g}\}. \quad (5.26)$$

The integral safe set, O_∞^I , is then defined as the set of initial plant and controller states, $\bar{x}(0)$, such that the ensuing closed-loop trajectories are constraint-admissible. That is,

$$O_\infty^I = \{\bar{x}(0) \in \mathbb{R}^{n+p} : \bar{C}\bar{A}^k\bar{x}(0) \in \bar{Y}, \forall k \geq 0\}. \quad (5.27)$$

If \bar{A} is Schur, (\bar{C}, \bar{A}) is observable, $0 \in \text{int } \bar{Y}$, and \bar{Y} is compact, (5.27) is positively-invariant, finitely-determined and is a polytope [58].

We define

$$R_\infty^I = \{x(0) : \exists x_I(0) \text{ such that } (x(0), x_I(0)) \in O_\infty^I\}, \quad (5.28)$$

as the projection of O_∞^I onto the plant states. Finding a suitable $x_I(0)$ for a given $x(0) \in R_\infty^I$ can be posed as a quadratic programming problem, see subsequent chapters.

5.4 Comparison of safe, recoverable, and integral safe sets

The sets O_∞ , O_∞^I , and O_∞^{ext} can be used to generate a constraint-admissible trajectory for a given $x(0) \in O_\infty$, R_∞^I , or R_∞ , respectively.

The practical use of O_∞^{ext} and O_∞^I is similar. In the former case, a polyhedral representation of O_∞^{ext} needs to be stored or computed on board and then a quadratic programming problem, of the form (5.15), is solved to determine the initial state of the auxiliary subsystem and the recovery sequence. In the latter case, a polyhedral representation of O_∞^I needs to be stored or computed on board and then a quadratic programming problem is solved with respect to the controller state to determine the value to which the controller state needs to be set so as to enforce constraints.

Clearly, $O_\infty \subseteq R_\infty$, $O_\infty \subseteq R_\infty^I$, $R_\infty \subseteq V_\infty$, and $R_\infty^I \subseteq V_\infty$, however, $R_\infty^I \not\subseteq R_\infty$, nor $R_\infty \not\subseteq R_\infty^I$ in general. We expect that as \hat{n} increases in the shift register case, R_∞ approaches V_∞ in the Hausdorff norm sense but the proof of this conjecture, under suitable assumptions, is left to future research.

To demonstrate the differences in the sets, consider the following aircraft examples. These examples are generated using the scaled NASA generic transport model (GTM) [107, 108]. The trim condition for which the following results are calculated is at an altitude of

$h_0 = 800$ ft, airspeed of $U_0 = 118.15 \frac{\text{ft}}{\text{s}}$ (70 knots), sideslip angle $\beta_0 = 0$ rad, flight path angle $\gamma_0 = 0$ rad, and yaw rate $\dot{\psi}_0 = 0 \frac{\text{rad}}{\text{s}}$.

In the following, the states of the linear models are deviations from the trim conditions at which the nonlinear model is linearized. For the lateral dynamics, the states are deviations in side-slip angle, $\Delta\beta$, roll rate, Δp , body-fixed vertical axis rotation, Δr , and roll angle, $\Delta\phi$, with angles in rad and rates in $\frac{\text{rad}}{\text{s}}$. The control inputs for the lateral dynamics are deviations in the nominal control input of the aileron, $\Delta\delta_a$, in rad and the rudder, $\Delta\delta_r$, in rad. For the longitudinal dynamics, the states are deviations in longitudinal airspeed, Δu , angle of attack, $\Delta\alpha$, pitch rate, Δq , and pitch angle, $\Delta\theta$, with airspeed in $\frac{\text{ft}}{\text{s}}$, angles in rad, and angular rates in $\frac{\text{rad}}{\text{s}}$. The control inputs for the longitudinal dynamics are deviations in the nominal control input of the elevator, $\Delta\delta_e$, in rad and the throttle, $\Delta\delta_T$, in percent throttle.

The lateral continuous-time linear model at the chosen trim condition is

$$\begin{aligned} \begin{bmatrix} \dot{\Delta\beta} \\ \dot{\Delta p} \\ \dot{\Delta r} \\ \dot{\Delta\phi} \end{bmatrix} &= \begin{bmatrix} -0.5229 & 0.0861 & -0.9852 & 0.2374 \\ -90.5885 & -6.2736 & 2.0861 & 0 \\ 29.1873 & -0.4833 & -1.4043 & 0 \\ 0 & 1 & 0.0857 & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\phi \end{bmatrix} \\ &+ \begin{bmatrix} -0.0002 & 0.0031 \\ -0.9174 & 0.2321 \\ -0.0523 & -0.4436 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \end{bmatrix}, \end{aligned} \quad (5.29)$$

and the longitudinal model is given by

$$\begin{aligned} \begin{bmatrix} \dot{\Delta u} \\ \dot{\Delta\alpha} \\ \dot{\Delta q} \\ \dot{\Delta\theta} \end{bmatrix} &= \begin{bmatrix} -0.0665 & -11.4608 & 0.1439 & -32.1740 \\ -0.0035 & -2.4714 & 0.9514 & 0 \\ -0.0090 & -43.9070 & -3.4738 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta\alpha \\ \Delta q \\ \Delta\theta \end{bmatrix} \\ &+ \begin{bmatrix} -0.0435 & 0.1424 \\ -0.0043 & -0.0001 \\ -0.7662 & 0.0192 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_e \\ \Delta\delta_T \end{bmatrix}. \end{aligned} \quad (5.30)$$

Discrete-time models are generated with a time step of $T_s = 0.1$ seconds. To generate

O_∞^I for the lateral model, C_I in (5.21) is set to

$$C_{I,\text{lat}} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.31)$$

To generate O_∞^I for the longitudinal model, the integrator is applied to the error in flight path angle, $\gamma = -\alpha + \theta$, thus C_I in (5.21) is set to

$$C_{I,\text{lon}} = \begin{bmatrix} 0 & -1 & 0 & 1 \end{bmatrix}. \quad (5.32)$$

A comparison is now made between the projections of the sets O_∞ , R_∞ , and R_∞^I onto Δp - $\Delta \phi$ plane for lateral dynamics and the Δu - $\Delta \theta$ plane for longitudinal dynamics. For this comparison, the recovery sequences used to produce R_∞ are generated using a shift register for the auxiliary dynamics (5.12) with $H = 3$. The nominal controller, (5.4), and K_1 and K_2 in (5.22) were designed using discrete-time LQ techniques and tuned for good transient response. The value of K in (5.4) and K_1 in (5.22) were made the same. The values for K_1 and K_2 for the lateral dynamics were

$$K_1 = \begin{bmatrix} 1.4888 & -0.3021 & -0.8550 & -2.1804 \\ 0.4416 & -0.2362 & -1.2210 & -2.1287 \end{bmatrix},$$

$$K_2 = \begin{bmatrix} -0.0680 \\ -0.0684 \end{bmatrix},$$

and for the longitudinal dynamics K_1 and K_2 were

$$K_1 = \begin{bmatrix} 0.7565 & 25.4919 & -2.1734 & -33.0973 \\ 0.3818 & -5.2650 & -0.0921 & 4.3783 \end{bmatrix},$$

$$K_2 = \begin{bmatrix} -0.3770 \\ -0.9132 \end{bmatrix}.$$

The constraints for the lateral linearized system in (5.2) are

$$\begin{aligned}
-0.2618 &\leq \Delta\beta \leq 0.2618, \\
-0.2618 &\leq \Delta p \leq 0.2618, \\
-0.2618 &\leq \Delta r \leq 0.2618, \\
-1.0472 &\leq \Delta\phi \leq 1.0472, \\
-0.5236 &\leq \Delta\delta_a \leq 0.5236, \\
-0.5236 &\leq \Delta\delta_r \leq 0.5236,
\end{aligned} \tag{5.33}$$

where $\Delta\beta$, $\Delta\phi$, $\Delta\delta_r$, and $\Delta\delta_a$ are in rad and Δp and Δr is in $\frac{\text{rad}}{\text{s}}$. The constraints for the longitudinal linearized system used are

$$\begin{aligned}
-20 &\leq \Delta u \leq 20, \\
-0.0987 &\leq \Delta\alpha \leq 0.1456, \\
-0.2618 &\leq \Delta q \leq 0.2618, \\
-0.6981 &\leq \Delta\theta \leq 0.6981, \\
-0.5236 &\leq \Delta\delta_e \leq 0.5236, \\
-20 &\leq \Delta\delta_T \leq 20,
\end{aligned} \tag{5.34}$$

where Δu is in $\frac{\text{ft}}{\text{s}}$, $\Delta\alpha$, $\Delta\theta$ and $\Delta\delta_e$ are in rad, Δq is in $\frac{\text{rad}}{\text{s}}$, and $\Delta\delta_T$ is in percent throttle.

Figures 5.1 and 5.2 show projections of O_∞ , R_∞ , and R_∞^I onto selected coordinates for lateral and longitudinal dynamics, respectively. From Figure 5.1, it is clear the R_∞^I is substantially larger than R_∞ (while O_∞ is a subset of both). However, in Figure 5.2, R_∞ has a larger area than R_∞^I , but not substantially larger (while O_∞ is again a subset of both).

Utilizing MATLAB running on a laptop with a 2.66 GHz Intel Core 2 Duo processor and 4 GB of RAM, O_∞ and O_∞^I can be computed in about 0.5 seconds. However, to compute R_∞ with $H = 3$ on the same laptop requires roughly 5 seconds (and subsequently the calculations for R_∞ with $H = 6$ requires about 30 seconds). All sets were calculated using the techniques described in Appendix B.

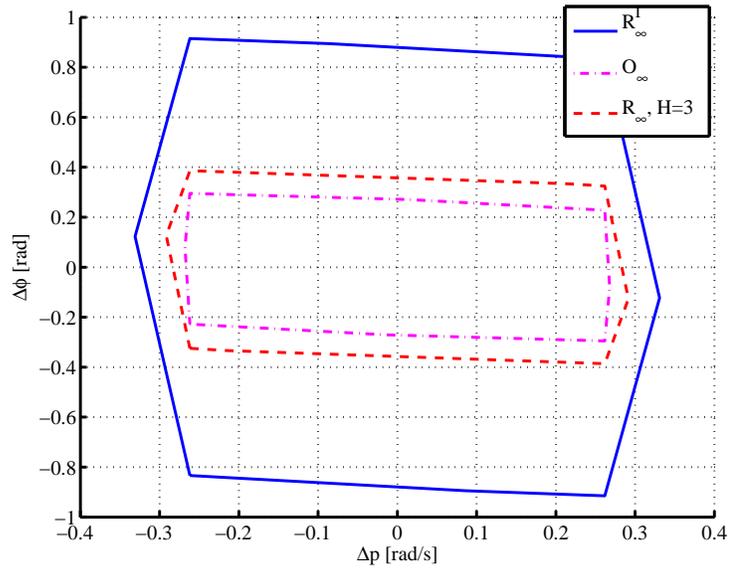


Figure 5.1: Projections of O_∞ , R_∞ , and R_∞^I onto the Δp - $\Delta\phi$ plane for the lateral linearized model. Note that R_∞^I is substantially larger than R_∞ and takes less time to compute.

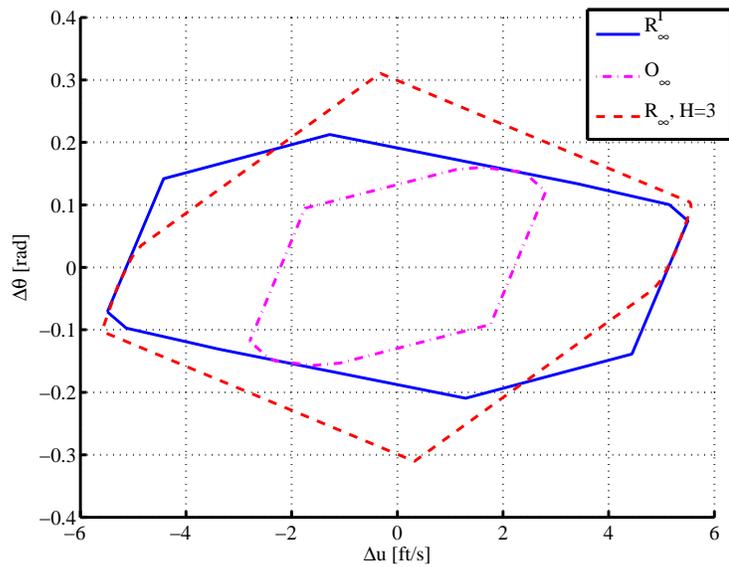


Figure 5.2: Projection of O_∞ , R_∞ , and R_∞^I onto the Δu - $\Delta\theta$ plane for the longitudinal linearized model. Note that R_∞^I and R_∞ are similar in size (though R_∞ is slightly larger) while R_∞^I takes less time to compute.

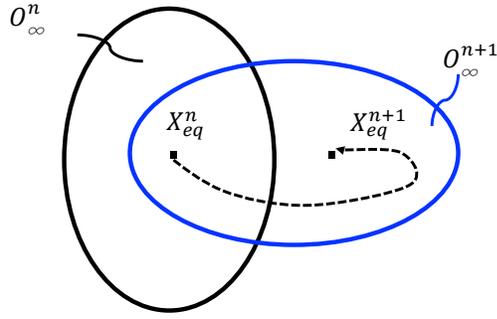


Figure 5.3: Example of the connectivity condition such that a transition from X_{eq}^n to X_{eq}^{n+1} is allowed.

5.5 Aircraft maneuver planning using safe, recoverable, and integral safe sets

The safe, recoverable, and integral safe sets can be used in aircraft maneuver planning to determine a sequence of trim points that can be safely followed by the aircraft. In the case of safe sets, the condition for connectivity and safe transition between trim points can be expressed as

$$X_{eq}^n - X_{eq}^{n+1} \in \text{int } O_{\infty}^{n+1}, \quad (5.35)$$

where X_{eq}^n , X_{eq}^{n+1} denote the n th and $(n + 1)$ th equilibrium states of the nonlinear closed-loop system, respectively, and O_{∞}^{n+1} denotes the safe set corresponding to the linearized closed-loop model at X_{eq}^{n+1} , see Figure 5.3. With the controller commands corresponding to the n th trim point, $X(t)$ converges to X_{eq}^n and (5.35) ensures that the $(n + 1)$ th trim condition can be commanded and achieved without violating the constraints.

Note also that in (5.35), $\text{int } O_{\infty}^{n+1}$ can be replaced with $\text{int } R_{\infty}^{n+1}$ or $\text{int } R_{\infty}^{I,n+1}$ with the same implications. If R_{∞}^{n+1} replaces O_{∞}^{n+1} in (5.35), i.e.,

$$X_{eq}^n - X_{eq}^{n+1} \in \text{int } R_{\infty}^{n+1}, \quad (5.36)$$

then a recovery sequence is necessary to achieve the commanded trim condition without violating constraints. The actual recovery sequence is computed by solving (5.15) with $x(0) = X(t) - X_{eq}^{n+1}$, where $X(t)$ is the current state, and $O_{\infty}^{ext} = O_{\infty}^{ext,n+1}$ corresponds to R_{∞}^{n+1} . Note that neither the knowledge of the actual recovery sequence, nor $O_{\infty}^{ext,n+1}$, is required at the maneuver planning stage to determine a safe sequence of trim points based on condition (5.36). A similar discussion applies to the case when the integral safe sets are

used for maneuver planning with (5.35) being replaced by

$$X_{eq}^n - X_{eq}^{n+1} \in \text{int } R_{\infty}^{I,n+1}. \quad (5.37)$$

Reference [60], see also [75, 85], exemplifies a similar approach based on safe sets in the case of spacecraft relative motion control while enforcing debris avoidance and thrust limit constraints. It uses a condition of the form (5.35) to compute a connectivity graph and then perform a graph search to determine the path between the origin and destination. Here, we generalize the approach of [60] to the one based on recoverable and integral safe sets which allows significantly more trim conditions to be connected.

Note that O_{∞} , R_{∞} , and R_{∞}^I can be replaced by their subsets in (5.35), (5.36), (5.37), respectively. Thus, in principle, the knowledge of the sets O_{∞} , R_{∞} , and R_{∞}^I is not required and the knowledge of their subsets is sufficient.

We will now present an example of constraint-admissible aircraft maneuver planning via R_{∞}^I . Towards this end, we consider a longitudinal aircraft dynamics model. Note that a longitudinal trim condition is uniquely defined by prescribing an airspeed, U_0 , flight path angle, γ_0 , altitude, h_0 , and $\dot{\psi} = \text{constant}$ (in this case, zero).

The constrained aircraft maneuver planning problem reduces to a graph search to identify a sequence of connected, i.e., satisfying (5.37), trim points leading from the initial trim state to the final trim state.

The execution of the flight path proceeds as follows. Let $i(k) \in \{1, \dots, N\}$ designate the index of the currently commanded trim point. Suppose the operation is currently with the trim point index, $i(k) = i^-$, i.e., the commanded trim point is $X_{eq}^{i^-}$. At a time instant, k , we check if the condition (5.37) is satisfied, or equivalently if the condition

$$\begin{bmatrix} X(k) - X_{eq}^{i^+} \\ x_I(k) \end{bmatrix} \in O_{\infty}^{I,i^+}, \quad (5.38)$$

where $i^+ = \min\{i^- + 1, N\}$, can be satisfied by a selection of the integrator state, $x_I(k) \in \mathbb{R}^p$. If a feasible solution exists, then the switch $i(k) = i^+$ is made, resulting in an update to the current trim condition and feedback control gains, while the integrator state is appropriately reset to the feasible value thereby found. We note that O_{∞}^I sets used in the above procedure are based on local (linearized) models at various trim points. Approaches to account for the effects of nonlinearities will be considered in the future work.

The following example illustrates this procedure. The desired transition is from an initial trim condition of $h_0^0 = 800$ ft, $U_0^0 = 151.90$ fps (90 knots), and $\gamma_0^0 = 0.0873$ rad (5 deg) to $h_0^N = 800$ ft, $U_0^N = 151.90$ fps (90 knots), and $\gamma_0^N = -0.0873$ rad (-5 deg). This

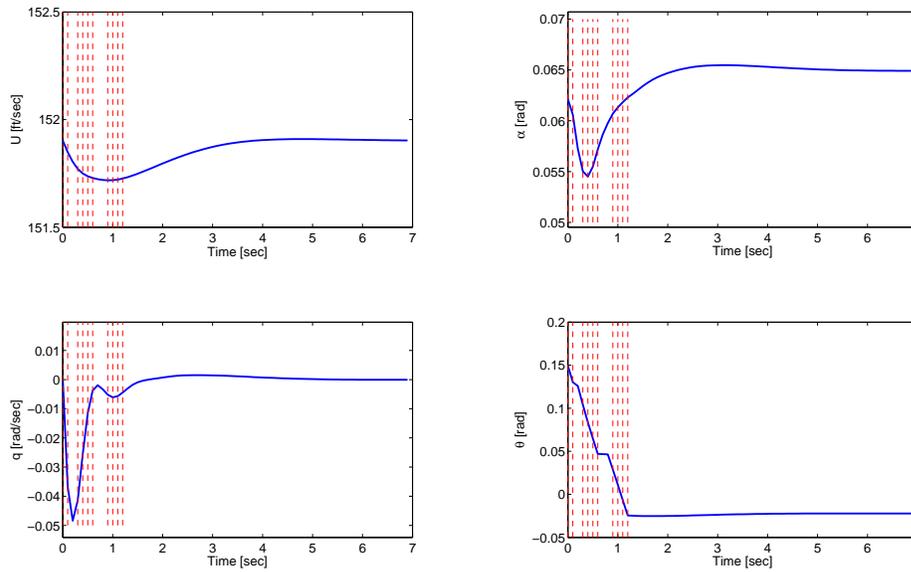


Figure 5.4: The time histories of aircraft states as the aircraft transitions through the selected trim points. The transitions between selected trim points occur at time instants designated by vertical dashed lines.

transition is not allowed initially due to imposed constraints. However, if intermediate trim points are added with $\gamma_0^1 = 0.0698$ rad (4 deg), $\gamma_0^2 = 0.0524$ rad (3 deg), $\gamma_0^3 = 0.0349$ rad (2 deg), $\gamma_0^4 = 0.0175$ rad (1 deg), $\gamma_0^5 = 0$ rad (0 deg), $\gamma_0^6 = -0.0175$ rad, $\gamma_0^7 = -0.0349$ rad, $\gamma_0^8 = -0.0524$ rad, and $\gamma_0^9 = -0.0698$ rad, thus $N = 10$, this allows for the desired transition to occur with a constraint-admissible trajectory.

Figures 5.4 and 5.5 show the state and control trajectories of this transition. Note that these Figures plot absolute values (i.e., not values relative to a trim condition) of the states and the control. The red vertical dashed lines show at what time instants the switch has occurred to a new trim condition. The first switch occurs at $k = 0$ as the aircraft begins its transition from its initial trim of $\gamma_0^0 = 5$ deg to the next trim of $\gamma_0^1 = 4$ deg. Note that in these Figures, the integrator state is reset at each time instant and this accounts for the vertical jumps in the integrator state plot in Figure 5.5.

It should be noted that the original transition from $\gamma_0 = 5$ deg to $\gamma_0 = -5$ deg may have actually been *safe* in that the aircraft would not experience ill effects by making this transition. However, because our models are local models and the constraints that are imposed reflect this, this transition was deemed to be not constraint admissible.

We note that references [71, 74] produce sets of forward and backward reachable states based on nonlinear aircraft models. To make the computation time feasible on board an

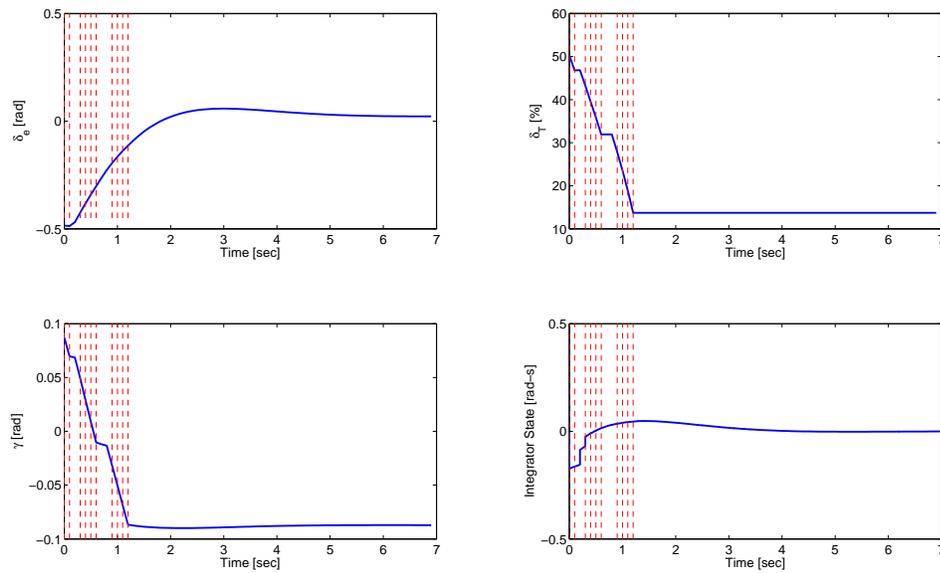


Figure 5.5: The elevator and thrust control inputs, flight path angle γ , and integrator state trajectories as the aircraft transitions through the selected trim points. Note the vertical jumps in the integrator state (bottom right) plot indicate the integrator state reset. The transitions between selected trim points occur at time instants designated by vertical dashed lines.

aircraft (they report around 4 to 5 seconds), these sets are computed for a relatively short time horizon using the slow aircraft dynamics, while neglecting the fast aircraft dynamics, and considering flight envelope constraints. In our case we focus on quick and computationally inexpensive local estimates that consider flight envelope and linear model validity constraints. While both can be performed iteratively on board an aircraft as dynamics and constraints change, they produce different, albeit similar, sets of *safe* states. A possible comparison could be made between the maneuverability envelopes in [71, 74] and a union of safe, recoverable, or integral safe sets over all interconnected trim conditions according to the criteria (5.35), (5.36), (5.37), respectively. This graph would describe, or approximate, which trim conditions are connected based on the above analysis for the entire known flight envelope. This analysis is left to future work and publications.

As stated above, the described safe, recoverable, and integral safe sets, along with the associated maneuver planning example, were developed for linearized models of the scaled NASA GTM. While the results indicate that this procedure works for the linearized models, this procedure needs to be tested and validated on the nonlinear aircraft model. This analysis is left for future work.

5.6 Further examples of safe and recoverable sets for aircraft dynamics

This section will discuss the properties of safe and recoverable sets including the effects of set-bounded disturbances on safe and recoverable sets, the dependence of recoverable sets on the trim condition for which they are calculated, and auxiliary system and recovery sequence length selection.

Subsections 5.6.1 and 5.6.2 compare the shapes and sizes of safe and recoverable sets at different flight conditions, different recovery horizons, and different recovery sequence generators. This provides important insights and the motivation behind the scaling transformations presented in the next chapter. Subsection 5.6.3 demonstrates how the safe and recoverable sets change under additive set-bounded disturbances.

We consider the linearized aircraft dynamics based on the NASA GTM [107, 108], see also Appendix C. The nominal trim condition for which the linearization is generated is defined by $U_0 = 515.55 \frac{\text{ft}}{\text{s}}$, $h_0 = 1000 \text{ ft}$, $\alpha_0 = 5 \text{ deg}$, $\beta_0 = 0 \text{ rad}$, and $\dot{\psi}_0 = 0 \frac{\text{rad}}{\text{s}}$.

The lateral model is

$$\begin{bmatrix} \Delta \dot{\beta} \\ \Delta \dot{p} \\ \Delta \dot{r} \\ \Delta \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0.1646 & 0.0033 & -1.0018 & 0.0625 \\ -10.5393 & -3.0745 & 0.5421 & 0 \\ 2.7760 & 0.0174 & -0.6265 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \\ \Delta \phi \end{bmatrix} + \begin{bmatrix} -0.0059 & -0.0728 \\ 7.3283 & 1.6432 \\ 0.0226 & -2.2790 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_r \\ \Delta \delta_a \end{bmatrix}, \quad (5.39)$$

and the longitudinal model is

$$\begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \\ \Delta \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.0102 & -9.7264 & -1.4988 & -32.2 \\ -0.0004 & -0.9992 & 0.9617 & 0 \\ 0.0016 & -2.4752 & -0.7972 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} 4.4286 & 3.33 \\ -0.0856 & 0 \\ -3.3372 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_T \end{bmatrix}, \quad (5.40)$$

with the same state and control definitions from Section 5.4.

The continuous time models are transformed to discrete-time models with a time step of $T_s = 0.2$ sec. A stabilizing controller is designed using discrete-time linear quadratic regulator (DLQR) theory. The weighting matrices used in the DLQR problem are diagonal, and chosen as $Q_{\text{lon}} = \text{diag}([0.001, 0, 0, 10])$ and $R_{\text{lon}} = I$ for the longitudinal system and $Q_{\text{lat}} = \text{diag}([1, 0, 0, 1])$ and $R_{\text{lat}} = I$ for the lateral system. The controller has the form

$$u = -Kx + v$$

where x , u , are the state and control vectors of the lateral and longitudinal aircraft models ((5.39) or (5.40), respectively) and v is the output of the auxiliary subsystem (5.12).

The constraints imposed on the lateral dynamics are

$$\begin{aligned}
-0.1745 &\leq \Delta\beta \leq 0.1745, \\
-0.0873 &\leq \Delta p \leq 0.0873, \\
-0.0873 &\leq \Delta r \leq 0.0873, \\
-0.3491 &\leq \Delta\phi \leq 0.2491, \\
-0.2618 &\leq \Delta\delta_a \leq 0.2618, \\
-0.3491 &\leq \Delta\delta_r \leq 0.3491,
\end{aligned} \tag{5.41}$$

and the constraints imposed on the longitudinal dynamics are

$$\begin{aligned}
-50 &\leq \Delta u \leq 50, \\
-0.0349 &\leq \Delta\alpha \leq 0.2094, \\
-0.0873 &\leq \Delta q \leq 0.0873, \\
-0.6981 &\leq \Delta\theta \leq 0.6981, \\
-0.4363 &\leq \Delta\delta_e \leq 0.4363, \\
-20 &\leq \Delta\delta_T \leq 20.
\end{aligned} \tag{5.42}$$

5.6.1 Recoverable set examples for aircraft dynamics

Recoverable sets depend on the selection of recovery sequence length and recovery sequence generation mechanism. These issues will now be examined.

Figures 5.6 and 5.7 show projections of safe and recoverable sets for $H = 1$, $H = 2$, and $H = 3$ for the lateral and longitudinal dynamics, respectively. In these figures, note that as H increases, so does the size of R_∞ . Also note that R_∞ approaches a maximum set as H increases which corresponds to the original constraints. It took a total of 0.58 sec to compute the longitudinal and lateral safe sets, 1.89 sec to compute the $H = 1$ recoverable sets, 3.55 sec to compute the $H = 2$ recoverable sets, and 6.96 sec to compute the $H = 3$ recoverable sets. The computations were performed using MATLAB code based on the work of Pluymers et al [83], see Appendix B, running on a laptop with a 2.66 GHz Intel Core 2 Duo processor and 4 GB of RAM. Note that these computing times are small. The ability to compute recoverable sets fast is attractive from a practical standpoint. This ability is facilitated by our use of discrete-time linear models and restricting recovery sequences to those that can be generated by a low dimensional auxiliary subsystem.

Figures 5.8-5.9 demonstrate the difference in recoverable sets for the above trim con-

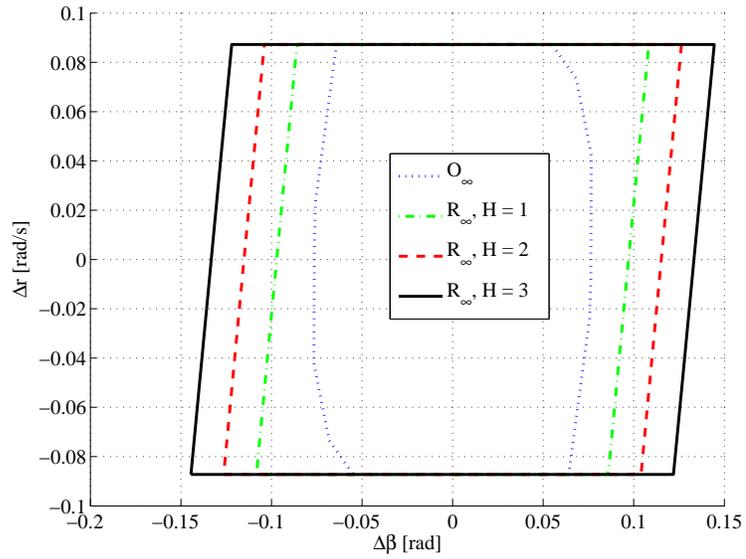


Figure 5.6: Projections of O_∞ and R_∞ onto the $\Delta\beta$ - Δr plane. Note that as H increases, so does R_∞ .

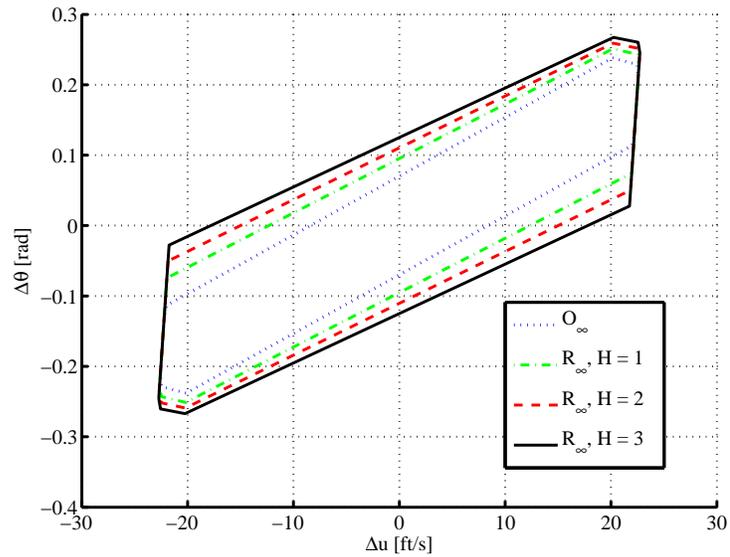


Figure 5.7: Projections of O_∞ and R_∞ onto the Δu - $\Delta\theta$ plane. Note that as H increases, so does R_∞ .

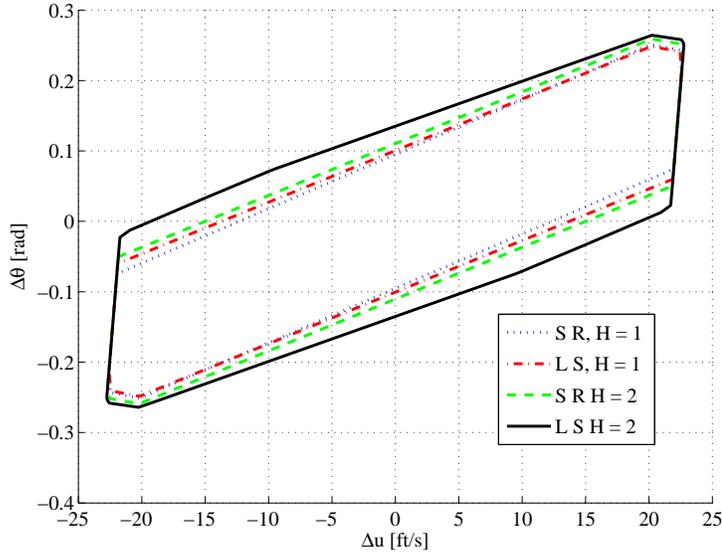


Figure 5.8: The difference between using a shift register (S R) and Laguerre's sequence generator (L S) for $H = 1$ and $H = 2$ recoverable sets with the longitudinal dynamics projected onto the Δu - $\Delta\theta$ plane. Note the Laguerre's sequence generator produces larger set in general than the shift register.

dition when a Laguerre's sequence generator (5.18) is used to augment the nominal system instead of the shift register. Note that while the recoverable sets are similar, the recoverable sets generated using the Laguerre's sequence are generally larger. For the computations, $\alpha_L = 0.5$ in (5.18).

5.6.2 Recoverable set dependence on flight condition

The recoverable sets are dependent on the flight condition for which they are being calculated. The dependence on flight condition comes primarily from two sources. The first source is the dependence of the matrix entries on the dynamic pressure, $p_d = 1/2\rho U_0^2$, see equations summarized in Appendix C. The second source is in the dependence of the stability derivatives on Mach number and angle of attack.

Figures 5.10 and 5.11 demonstrate the effect that different flight conditions have on the recoverable sets. A large difference in altitude, 1000 ft to 20000 ft has less of an effect than a moderate change in speed, $500 \frac{\text{ft}}{\text{s}}$ to $800 \frac{\text{ft}}{\text{s}}$. This is reasonable when considering the dependence of the dynamic pressure on the density and speed: Though p_d is dependent on the air density ρ , it also depends on the square of the airspeed U_0 . Hence, changes in U_0 affect the dynamic pressure more than changes in ρ . Table 5.1 summarizes four different

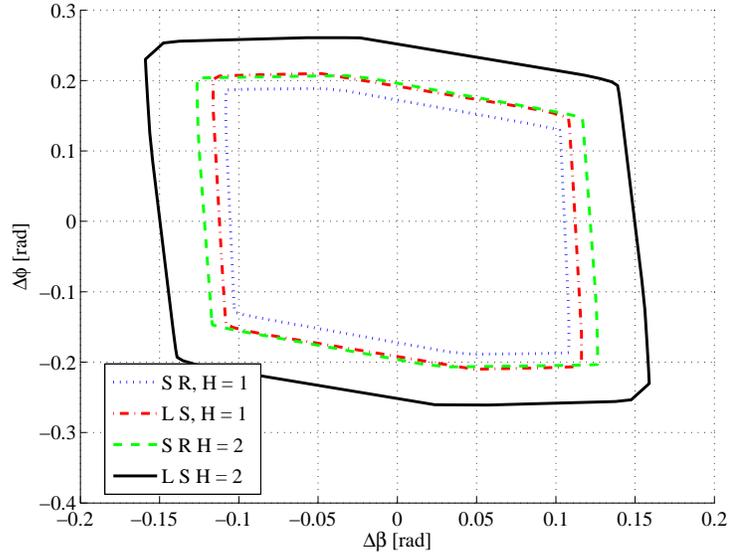


Figure 5.9: The difference between using a shift register (S R) and Laguerre's sequence generator (L S) for $H = 1$ and $H = 2$ recoverable sets with the lateral dynamics projected onto the $\Delta\beta$ - $\Delta\phi$ plane. Note the Laguerre's sequence generator produces larger set in general than the shift register.

flight conditions (FC1, FC2, FC3, FC4) for which the computations of the recoverable sets for the shift register implementation and $H = 1$ were performed.

5.6.3 Set-bounded disturbance effects on safe sets

The safe and recoverable set approach presented above can be modified to account for additive set-bounded disturbances, see reference [79]. To describe briefly the main ingredients, in the case of additive set-bounded disturbances/uncertainties, the system has the following form

$$\begin{aligned} x(k+1) &= A_{CL}x(k) + B_W w(k) + Bv(k) \\ y(k+1) &= C_{CL}x(k) + D_W w(k) + Dv(k), \end{aligned} \quad (5.43)$$

where the values of w are members of the known disturbance set, W . Under these disturbances, (5.6) becomes

$$O_\infty = \{x(0) \in \mathbb{R}^n : y(k) \in Y, \forall k \geq 0 \text{ and } \forall w \in W\}, \quad (5.44)$$

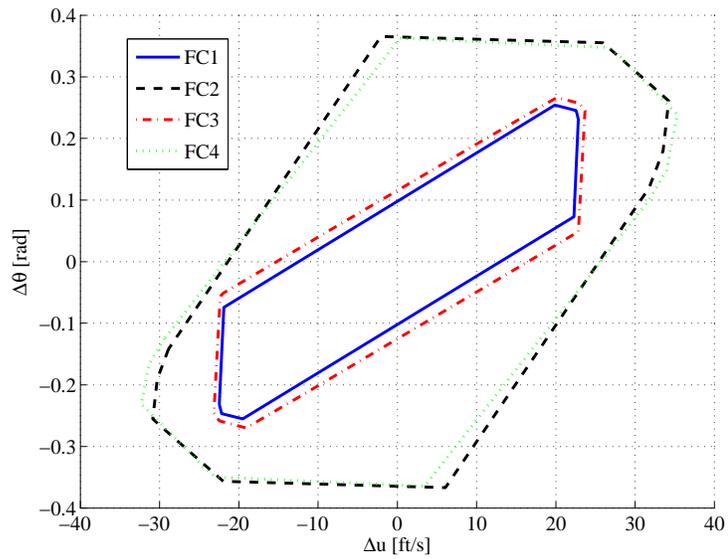


Figure 5.10: Projections of longitudinal recoverable sets with $H = 1$ at different flight conditions.

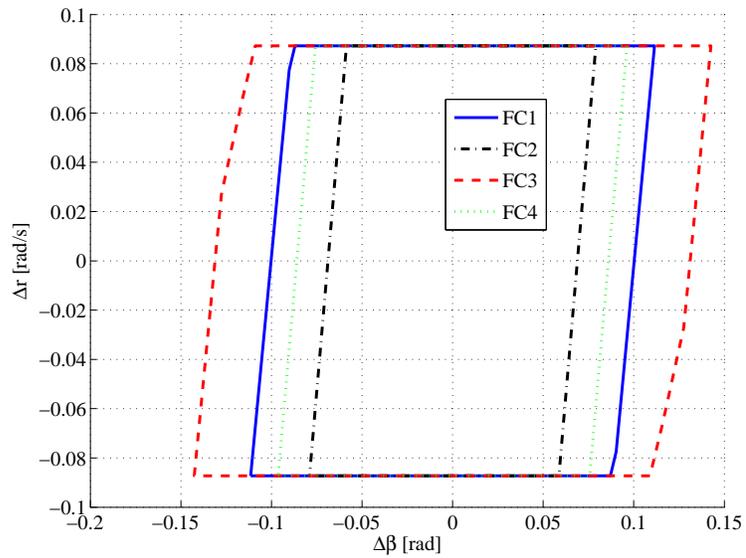


Figure 5.11: Projections of lateral recoverable sets for $H = 1$ at different flight conditions.

Table 5.1: Table of flight conditions used in Figures 5.10 and 5.11.

Flight Condition	$h[ft]$	$u_0[\frac{ft}{s}]$	$\alpha_0[deg]$
1	1000	500	4
2	1000	800	4
3	20000	500	4
4	20000	800	4

where

$$\begin{aligned}
 y(k) = C_{CL}A_{CL}^k x(0) &+ \sum_{t=0}^{k-1} C_{CL}A_{CL}^{(k-t-1)} B_W w(t) + D_W w(k) \\
 &+ \sum_{t=0}^{k-1} C_{CL}A_{CL}^{(k-t-1)} B v(t),
 \end{aligned} \tag{5.45}$$

where W is a bounded set of possible disturbances values and v is the output of the auxiliary system (5.12). Figures 5.12 and 5.13 demonstrate the effects of set-bounded disturbances for $H = 1$ recoverable sets for the flight condition $U_0 = 500 \frac{ft}{s}$, $h_0 = 1000$ ft, and $\alpha_0 = 2$ deg. For Figure 5.12, the bounded disturbance occurs in the Δu state and for Figure 5.13 the bounded disturbance occurs in the Δp state. Note that in Figure 5.13, though this projection does not include the Δp state, the disturbance effects are still observed.

The technique used to calculate the set-bounded disturbance recoverable sets is described in Appendix B.

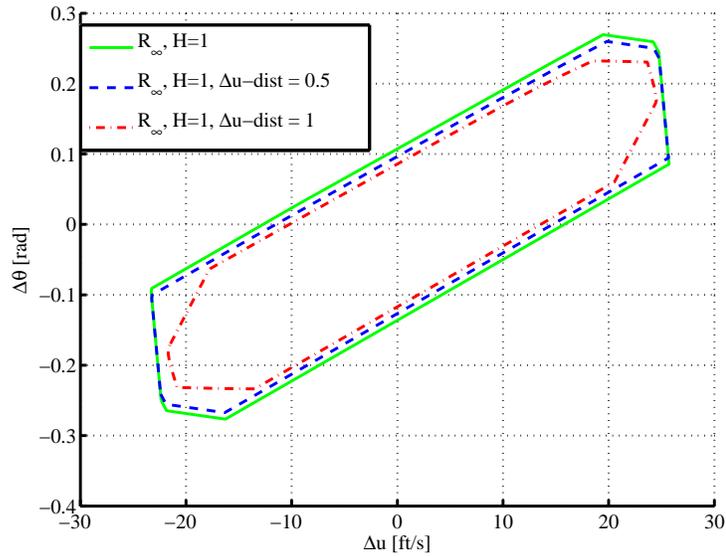


Figure 5.12: The effects of additive set-bounded disturbances on $H = 1$ recoverable sets. Note that as the disturbance increases in size, the recoverable set decreases.

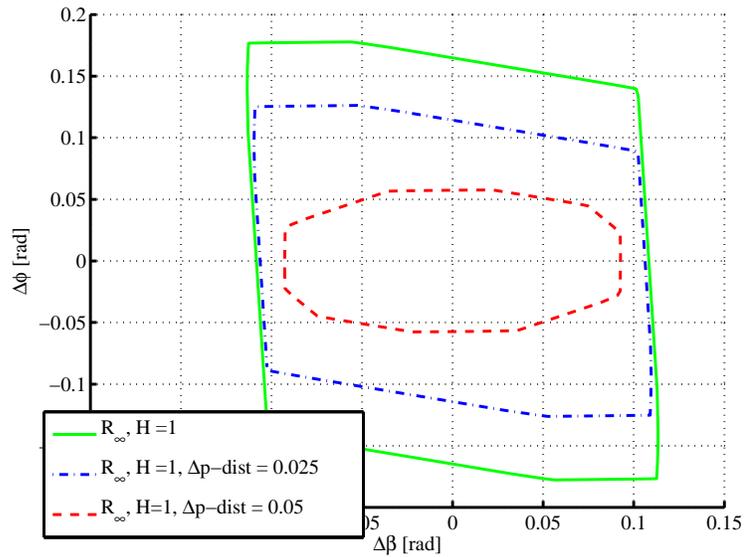


Figure 5.13: The effects of additive set-bounded disturbances on $H = 1$ recoverable sets. Note that as the disturbance increases in size, the recoverable set decreases.

CHAPTER 6

Scaling Between Safe and Recoverable Sets at Different Flight Conditions

This chapter discusses approaches to estimating safe and recoverable sets at unexplored trim conditions through the use of scalar (Section 6.1) and vector (Section 6.2) scaling transformations. Numerical results are presented in Section 6.3 with two examples. A simple sensitivity technique that can be used for quick adjustment of the safe or recoverable set representation in response to changes in model parameters is considered in Section 6.4. Similar scaling approaches can be developed for integral safe sets, but we do not consider this explicitly in this chapter.

If parameters of the system model change, it is of interest to develop simple computational procedures to predict changes in safe and recoverable sets so that they can be incorporated into the maneuver planning scheme of the type considered in the previous chapter. Scaling transformations provide a method of estimating safe and recoverable sets at unexplored trim conditions once model parameters have changed and these parameter changes have been identified. Thus, an approach based on scaling transformations that can be applied to safe or recoverable sets computed at one flight condition to predict subsets of safe or recoverable sets at another flight condition is considered in this chapter.

The scaling transformation is applied to a safe or recoverable set at one flight condition to guarantee that the scaled set is a subset (not necessarily positively-invariant) of the safe or recoverable set at another flight condition. The scaling transformation is developed conservatively off-line and stored for on-board use so that it can be applied irrespective of degradation or damage conditions. The use of progressively more conservative scalings depending on the damage assessment index can also be considered. Note if damage or aircraft model parameters change, trim states may change as well. Estimating changes in trim states is considered in [76] but not included in this dissertation.

In the following, we consider two polyhedral recoverable sets, each computed for a

different flight condition. The first is described by the set of linear inequalities

$$G_1 y \leq g_1, \quad (6.1)$$

where $G_1 \in \mathbb{R}^{m \times n_y}$, and $g_1 \in \mathbb{R}^m$ and has a set of n_1 vertices organized into an $n_y \times n_1$ matrix, V_1 . The second is described by the set of linear inequalities

$$G_2 y \leq g_2, \quad (6.2)$$

where $G_2 \in \mathbb{R}^{k \times n_y}$, and $g_2 \in \mathbb{R}^k$ and has a set of n_2 vertices organized into an $n_y \times n_2$ matrix, V_2 . We assume that the origin is in the interior of both (6.1) and (6.2). The objective is to determine a scaling transformation such that (6.1) is guaranteed to be inside (6.2). Thus, if the polyhedron (6.1) is known and a scaling transformation is known, subsets of (6.2) can be computed without actually computing (6.2).

Two methods of scaling are now discussed. The first method is scalar scaling, while the second is vector scaling.

6.1 Scalar scaling

For the scalar scaling, a scalar, $\lambda > 0$, is applied to the polytope (6.1). The scalar scaling transformation is expressed as

$$T_{scalar} : x \rightarrow \lambda x. \quad (6.3)$$

To ensure that the scaled polytope is inside (6.2) it is necessary and sufficient to ensure that the vertices of (6.1) are mapped inside (6.2), i.e.,

$$G_2 \lambda V_{1i} \leq g_2, \quad i = 1, \dots, n_1. \quad (6.4)$$

The optimal scaling factor, λ^* , is the largest value of λ for which (6.4) is satisfied. It can be computed as

$$\lambda^* = \min_{i,j} \left[\max \left[\frac{g_{2j}}{G_{2j} V_{1i}}, 0 \right] \right]. \quad (6.5)$$

In (6.5), G_{2j} denotes the j th row of G_2 , V_{1i} denotes the i th column of V_1 , g_{2j} denotes the j element of g_2 , and $1 \leq i \leq n_1$, $1 \leq j \leq k$.

6.2 Vector scaling

For the vector scaling transformation, scalars, $\lambda_1 > 0, \dots, \lambda_{n_y} > 0$, scale each coordinate individually so that

$$T_{vector} : x = (x_1, \dots, x_{n_y}) \rightarrow (\lambda_1 x_1, \dots, \lambda_{n_y} x_{n_y}). \quad (6.6)$$

Note that

$$T_{vector} = \sum_{k=1}^{n_y} \lambda_k T_k, \quad (6.7)$$

where T_k is a diagonal matrix with the (k, k) element equal to 1 and all other elements being zero.

To ensure that the polytope (6.1), after the transformation (6.7), is inside (6.2) it is necessary and sufficient to ensure that the vertices of (6.1) are mapped inside (6.2), i.e.,

$$\sum_{k=1}^{n_y} \lambda_k (G_2 T_k V_1) \leq g_2. \quad (6.8)$$

The optimal vector transformation is determined by maximizing

$$J = \sum_{k=1}^{n_y} \lambda_k \rightarrow \max, \quad (6.9)$$

subject to the constraints

$$\lambda_k \geq 0, \quad k = 1, \dots, n_y, \quad (6.10)$$

and subject to the constraint (6.8). The problem (6.8)-(6.10) is a linear programming problem.

6.3 Numerical examples

Two numerical examples will now be presented. The first is an example of scalar scaling transformations calculated for a grid of trim conditions. The second is a simple aircraft icing example.

6.3.1 Scalar scaling over a grid of trim conditions

In this example, the scalar scaling transformations (6.3) from a single trim condition to a grid of trim conditions is considered. The initial longitudinal trim condition is at an airspeed

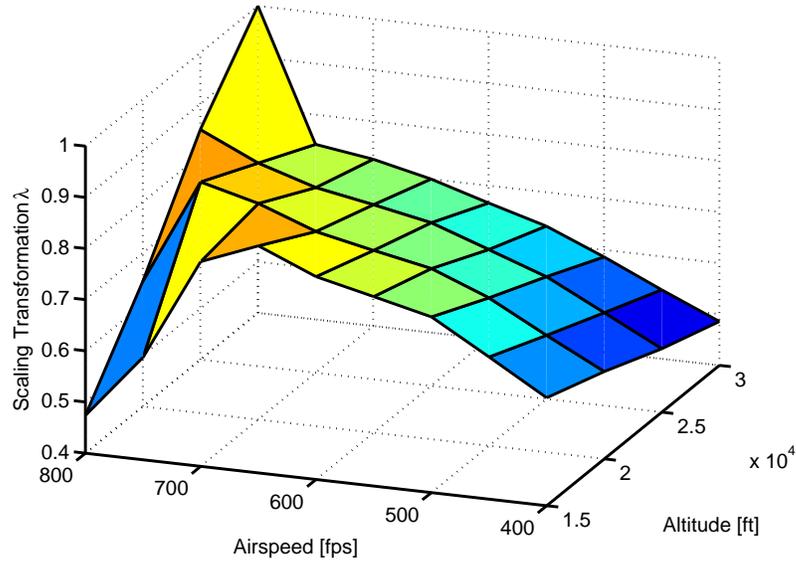


Figure 6.1: λ for scalar scaling transformations from the longitudinal trim condition of $U_0 = 800 \frac{\text{ft}}{\text{s}}$, $h_0 = 30000 \text{ ft}$, and $\alpha_0 = 2 \text{ deg}$ to all others in the trim condition grid.

$U_0 = 800 \frac{\text{ft}}{\text{s}}$, an altitude $h_0 = 30000 \text{ ft}$, and an angle of attack $\alpha_0 = 2 \text{ deg}$ and the grid is defined for each of the following combinations of trim conditions: $U_0 \in \{400, 450, \dots, 800\} \frac{\text{ft}}{\text{s}}$, $h_0 \in \{15000, 20000, \dots, 30000\} \text{ ft}$, with $\alpha_0 = 2 \text{ deg}$. Figure 6.1 presents the scalar scaling transformation factors, λ , from the original trim condition to all others within the grid. Note that the value of λ for the scaling transformation from $U_0 = 800 \text{ fps}$, $h_0 = 30000 \text{ ft}$, and $\alpha_0 = 2 \text{ deg}$ to itself is exactly 1.

Tables of scaling transformations (for both the scalar and vector scalings), similar to what is presented in Figure 6.1, can be calculated off-line and stored for on-board use. Linear interpolation can be used to generate scaling transformations between the grid points or, alternatively, a neural network [97] can be constructed where the inputs are the current trim condition and the objective trim condition to be scaled whilst the output is the scaling factor or factors. To utilize the neural network, a dataset of scaling factors would have to be calculated and used as training/testing data for the neural network.

Consider now a scenario where an aircraft is experiencing some malfunction and that an on-board path planning algorithm exists. Also assume that this aircraft has the capability of real-time linear model identification. In the case of emergency path planning, the aircraft can then use the on-board identification algorithm to identify the current linear model. A safe, recoverable, or integral safe set would then be calculated for this identified model and using the off-line generated scaling transformations (like those in Figure 6.1) and the

appropriate transition criterion ((5.35), (5.36), or (5.37) for safe, recoverable or integral safe sets, respectively), a connected graph of trim conditions between which safe transitions exist can be generated. A graph search algorithm can then be applied to find a path through this graph to get as close to the desired trim condition as possible.

6.3.2 Aircraft icing example

In this subsection, an aircraft icing example is presented. Two trim conditions are identified and scaling (both scalar and vector) transformations are calculated off-line for the nominal $H = 1, O_{\infty}^{ext}$ sets. These scaling transformations are then applied to $H = 1, O_{\infty}^{ext}$ sets calculated for the iced aircraft dynamics.

The effects of wing icing are modeled as reduced lift and increased drag [109, 110]. That is, the coefficients of lift, C_L , and drag, C_D , are altered in order to reflect increased drag and decreased lift consistent with wing icing. To do this, each stability derivative that is a component of the lift force or the drag force and appears in the linear models is multiplied by a factor so that

$$C_{L,new} = 1/2C_{L,nom}, \quad C_{D,new} = 2C_{D,nom}.$$

Figures 6.2 and 6.3 and Tables 6.1 and 6.2 show the results of computing the scaling transformations between the O_{∞}^{ext} sets for two different flight conditions with the nominal $H = 1, O_{\infty}^{ext}$ sets. These sets were calculated using the linearized GTM model, Appendix C, with constraints (5.41) and (5.42). In these figures, *original* indicates the set to be scaled, *objective* indicates the set to which the original set is being scaled, *scalar* indicates the scalar transformation is being used, and *vector* indicates the vector scaling transformation is being used. The first condition, labeled as 1 in Tables 6.1 (scalar scaling results) and 6.2 (vector scaling results), is $U_0 = 500 \frac{\text{ft}}{\text{s}}, h_0 = 1000 \text{ ft},$ and $\alpha_0 = 2 \text{ deg}$ and the second flight condition, labeled as 2 in Tables 6.1 and 6.2, is $U_0 = 580 \frac{\text{ft}}{\text{s}}, h_0 = 1000 \text{ ft},$ and $\alpha_0 = 2 \text{ deg}.$

For the longitudinal dynamics, the optimal scalar scaling factor, given in Table 6.1, is small and produces estimated $H = 1, O_{\infty}^{ext}$ sets that are conservative. This is assumed to be due to the auxiliary states introduced to in the $H = 1, O_{\infty}^{ext}$ set formulation and their interaction with constraints. The phenomenon, however, is not observed in the lateral dynamics.

Figure 6.4 presents the changes in the $H = 1, O_{\infty}^{ext}$ set for the changed dynamics in the $\Delta u-\Delta\theta$ and $\Delta u-\Delta\alpha$ planes. The observed variation, particularly in the $\Delta u-\Delta\theta$ plane, is consistent with what physically occurs when an aircraft experiences an increase in drag and a decrease in lift. The aircraft must pitch more at lower deviations from nominal airspeed

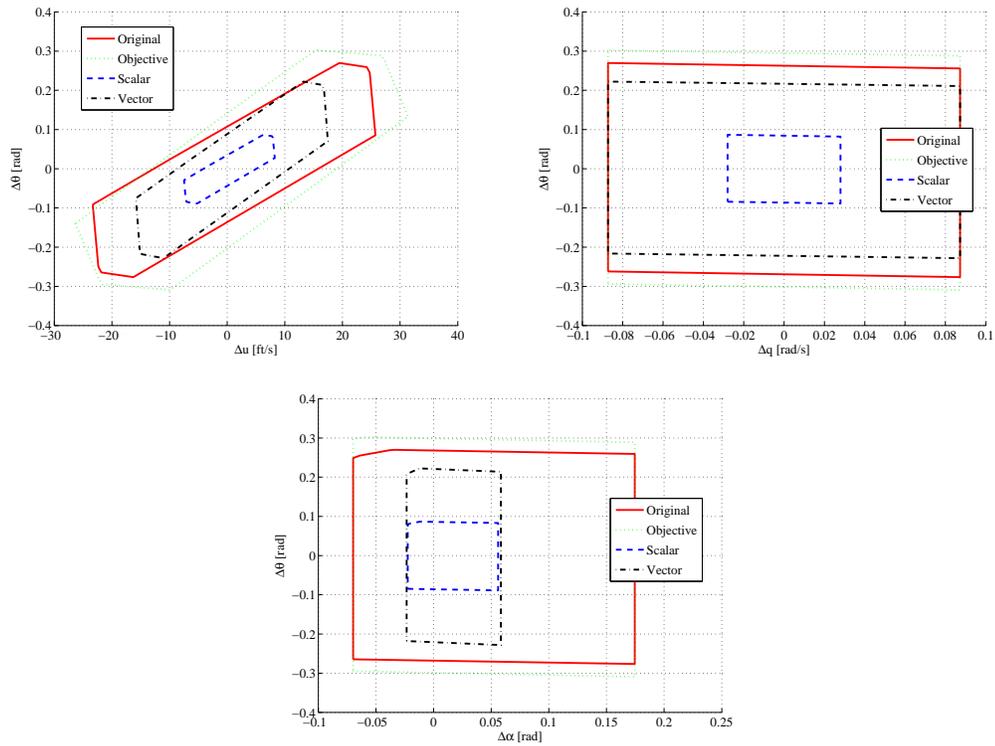


Figure 6.2: Example of longitudinal $H = 1$, O_{∞}^{ext} sets scaled from an original $H = 1$, O_{∞}^{ext} set to an objective $H = 1$, O_{∞}^{ext} set. Note that scalar scaling in this case is conservative where as the vector scaling is significantly better.

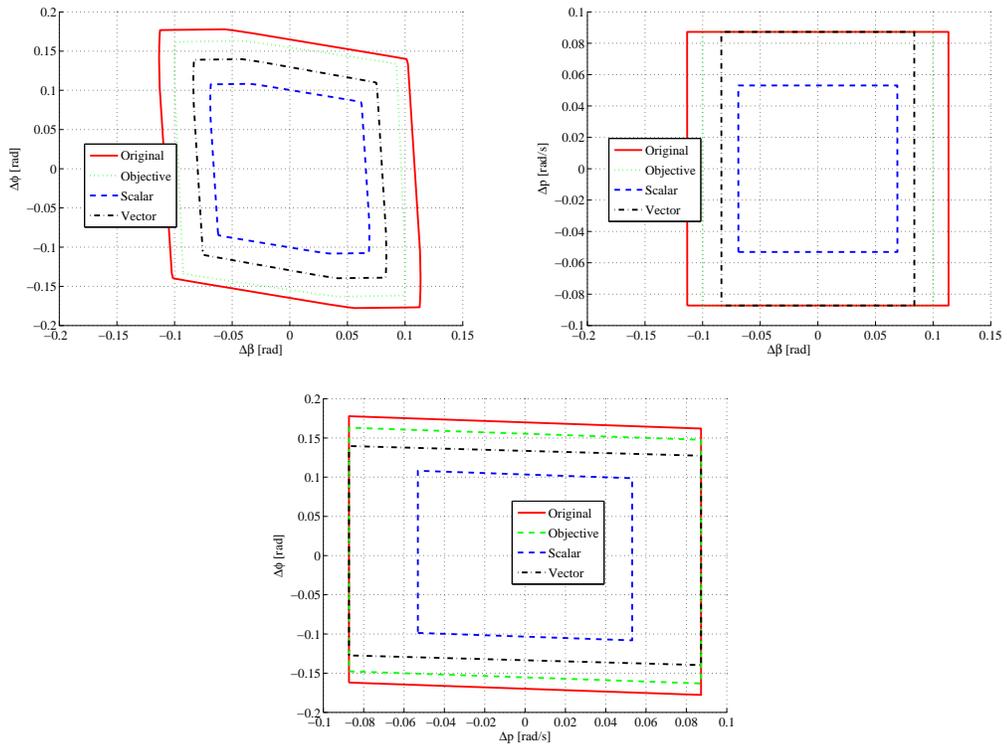


Figure 6.3: Example of lateral $H = 1$, O_{∞}^{ext} sets scaled from an original $H = 1$, O_{∞}^{ext} set to an objective $H = 1$, O_{∞}^{ext} set. Note that both scalar and vector scaling produce decent approximations in this case. Also note that the vector scaling comes pretty close to replicating the objective set in some examples.

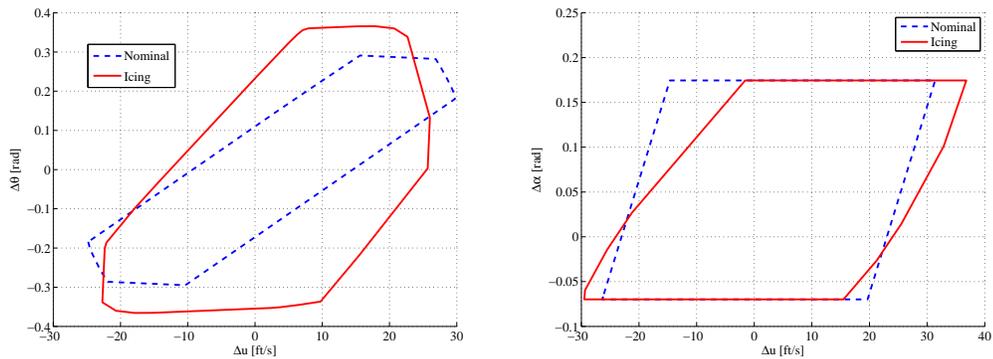


Figure 6.4: Examples of how the icing affects the $H = 1$, O_{∞}^{ext} sets in the $\Delta u-\Delta\theta$ plane (left) and the $\Delta u-\Delta\alpha$ plane (right). Note the icing is more predominately seen in the $\Delta u-\Delta\theta$ plane than in the $\Delta u-\Delta\alpha$ plane.

Table 6.1: Examples of optimal scaling parameters λ^*

Motion	Original	Objective	λ^*
Longitudinal	1	2	0.3210
Longitudinal	2	1	0.3695
Lateral	1	2	0.6087
Lateral	2	1	0.9688

Table 6.2: Examples of optimal scaling parameters $\hat{\lambda}^*$

Motion	Original	Objective	$\hat{\lambda}^*$
Longitudinal	1	2	[0.6792, 0.3346, 1.000, 0.8251, 0.6390, 0.5099, 0.8209, 0.8255]
Longitudinal	2	1	[0.6553, 0.9570, 1.000, 0.5092, 0.8470, 0.8765, 0.7584, 0.7619]
Lateral	1	2	[0.7384, 1.0000, 1.0000, 0.7861, 0.6840, 0.6567, 0.7005, 0.6660]
Lateral	2	1	[0.9803, 1.0000, 0.9851, 0.9203, 1.0639, 1.0869, 0.9692, 0.9712]

($-\Delta u$ direction) to avoid constraint violation.

Figure 6.5 provides the results of the scalings. The scalings that are used to perform these transformations are the scalings calculated for the nominal dynamics and found in Tables 6.1 and 6.2. Also note that with the performed transformation, the scaled $H = 1$, O_∞^{ext} sets are within the actual calculated $H = 1$, O_∞^{ext} sets.

This approach is dependent on the validity of the assumption that the linear model changes due to icing can be identified and that these identified changes can be translated to changes at other trim conditions. While this assumption will not hold for the entire flight envelope, it is plausible to assume that it will hold for small transitions in trim condition space. Icing largely effects the aerodynamics which are dependent on aircraft altitude, airspeed, and attitude. Small changes in these states should not dramatically change how the aircraft is responding under the icing conditions (assume the ice accretion is not too rapid). However, testing this approach on a nonlinear aircraft system with a valid icing model should be performed and is left for future work.

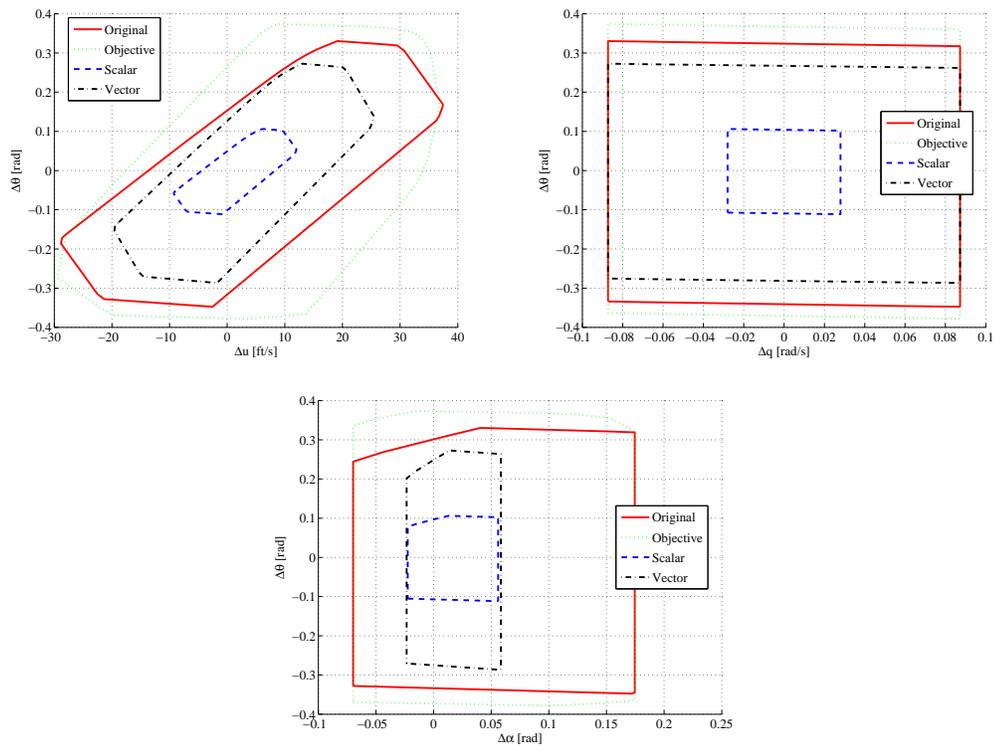


Figure 6.5: Scaling of the icing scenario $H = 1$, O_{∞}^{ext} sets

6.4 Sensitivity technique to predict changes in safe and recoverable sets

Given that the recoverable set computation reduces to the computation of the safe set, O_∞^{ext} , we consider the safe set computations in more detail. The constraints forming the safe set, O_∞ , are of the form (5.2), i.e.,

$$GC_{CL}A_{CL}^k x(0) \leq g, \quad (6.11)$$

defined for different values of k .

Suppose that the matrix A_{CL} is a function of an n_θ -vector parameter, θ , and suppose that the current estimate, $\hat{\theta}$, of this parameter has deviated from a nominal value, θ^* . Then, for small differences between $\hat{\theta}$ and θ^* we can approximate the constraints (6.11) as

$$\left(GC_{CL}A_{CL}^k(\theta^*) + GC_{CL}(kA_{CL}^{k-1}(\theta^*)) \sum_{i=1}^{n_\theta} \frac{\partial A_{CL}}{\partial \theta_i}(\theta^*)(\hat{\theta}_i - \theta_i^*) \right) x(0) \leq g. \quad (6.12)$$

The expression (6.12) indicates that the bounds of the inequalities defining O_∞ can be simply tightened or relaxed to accommodate the effect of parameter changes.

The approach based on (6.12) is applied only to the non-redundant constraints in the representation of O_∞ for $\theta = \theta^*$. In principle, large deviations of $\hat{\theta}$ from θ^* can lead to a change in redundancy status of the constraints, which is not accounted for by this technique. This approach, with minor modifications, is applicable to estimating changes in integral safe sets.

Figure 6.6 illustrates this predictive method. In these examples, θ is the vector of stability derivatives $C_{L0}, C_{Lu}, C_{L\alpha}, C_{Lq}, C_{D0}, C_{Du}, C_{D\alpha}, C_{Dq}$. In Figure 6.6, all of the stability derivatives associated with lift are decreased by the given percent and all of the stability derivatives associated with drag are increased by the given percent thereby emulating the effects of icing. Note that for small percent changes, the predicted recoverable set is fairly close to the actual set, but this changes quickly as the percent difference increases. These figures are for the $H = 1$ recoverable set at the flight condition $U_0 = 500 \frac{\text{ft}}{\text{s}}$, $h_0 = 1000 \text{ ft}$, and $\alpha_0 = 2 \text{ deg}$ as in Subsection 6.3.1.

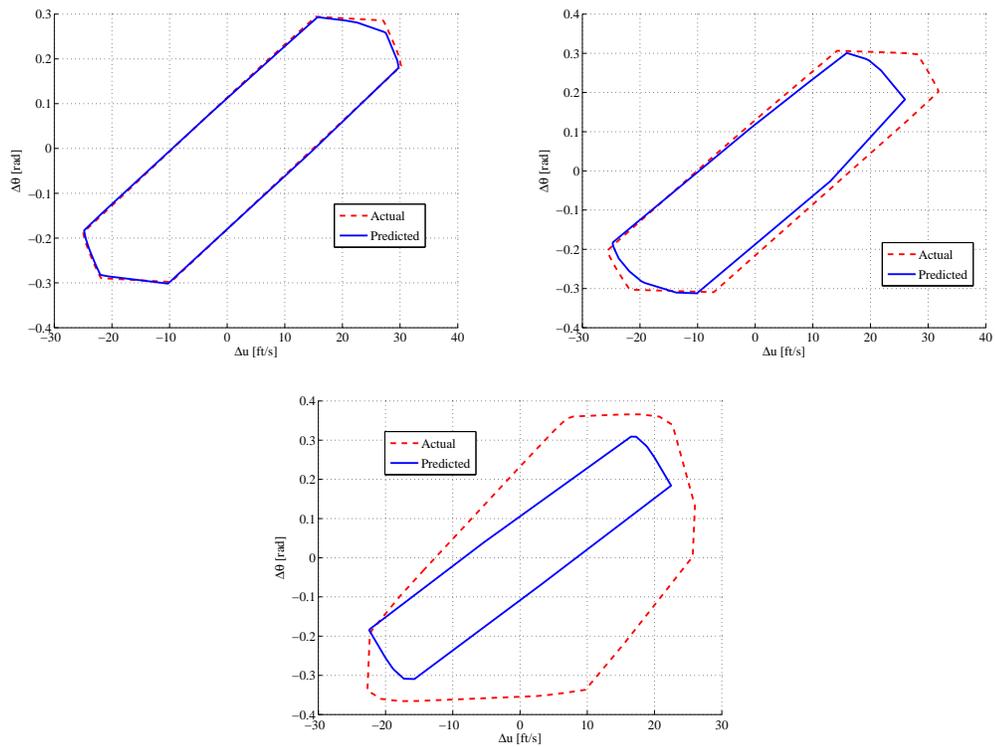


Figure 6.6: Examples of the accuracy of the sensitivity technique. Top left 2% change, Top right 10%, bottom 50% change. Note for large parameter changes, accuracy is poor.

CHAPTER 7

Controller State and Reference Governor with Aircraft Applications

In this chapter, we consider two classes of control schemes that reset controller states: the controller state governor for constrained stabilization problems and controller state and reference governor for constrained set-point tracking problems. See Figure 7.1 which shows the schematics of a controller state and reference governor that responds to a reference command $\bar{r}(t)$ by modifying $r(t)$ and resetting the controller state $x_I(t)$ based on the plant state estimate, $\hat{x}(t)$ to ensure that the constraints $y(t) \in Y$ are enforced. Subsequently, both schemes will be described, conditions will be provided under which they guarantee constraint enforcement and desirable convergence properties, and application-oriented simulation examples to demonstrate their benefits will be reported.

The development of these two control schemes is motivated by the desire to mitigate aircraft loss of control (LOC) situations. LOC, according to [54], is the violation of three or more critical envelopes. These critical envelopes are described by state, control, and structural constraints. Reference governors are add-on control schemes that augment well-designed, closed-loop systems and guarantee constraint enforcement. Aircraft have numerous constraints that can be more effectively enforced by enhancing reference governor techniques with the capability to reset dynamic controller states. Thus, constraints, like those that generate the critical envelopes, are more easily enforced without substantially changing the existing control architecture.

Section 7.1 introduces the controller state governor (CSG) for constrained stabilization problems and includes a proof of the maintenance of stability under controller state reset. Section 7.2 introduces the controller state and reference governor (CSRG) and includes a proof of finite-time convergence to a desired reference using the control scheme. Section 7.3 discusses the implications of the controller state and reference governor in conjunction with an LQ-I control scheme and nonlinear implementation. Finally, Section 7.4 presents

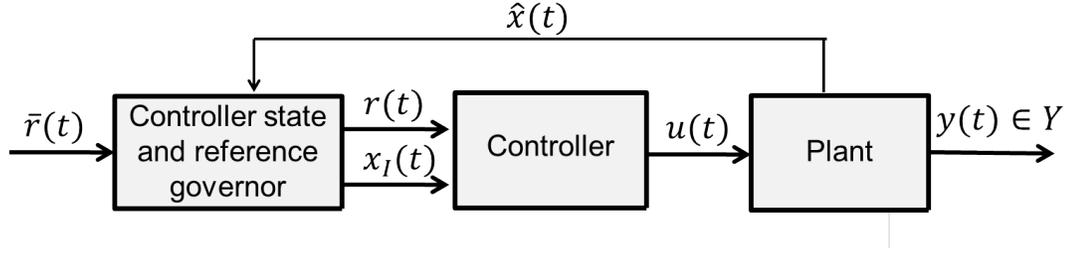


Figure 7.1: Controller state and reference governor schematic.

a constrained aircraft flight control example and Section 7.5 presents a constrained gas turbine engine control example.

7.1 Controller state governor

The controller state governor resets controller states in discrete-time linear closed-loop systems with constraints,

$$\bar{x}(k+1) = \bar{A}\bar{x}(k), \quad (7.1)$$

where the state \bar{x} consists of the plant states, $x(k)$, and resettable controller states, $x_I(k)$,

$$\bar{x}(k) = \begin{bmatrix} x(k) \\ x_I(k) \end{bmatrix}. \quad (7.2)$$

The matrix \bar{A} is assumed to be Schur (exponentially stable), i.e., eigenvalues of \bar{A} are strictly inside the unit disk in the complex plane. The constraints are imposed on the output of the system (7.1),

$$y(k) = \bar{C}\bar{x}(k), \quad (7.3)$$

and have the form,

$$y(k) \in Y = \{y \in \mathbb{R}^{n_y} : Gy \leq g\}, \quad (7.4)$$

where Y is a given compact and convex polytope with $0 \in \text{int } Y$ specified by a set of affine inequalities.

Example: Consider a linear discrete-time system which operates with the conventional proportional-plus integral controller of the form,

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ x_I(k+1) &= C_I x(k) + x_I(k), \end{aligned} \quad (7.5)$$

where $x(k) \in \mathbb{R}^n$, $x_I(k) \in \mathbb{R}^m$, and $u(k) \in \mathbb{R}^p$ is given by

$$u(k) = -K_1x(k) - K_2x_I(k). \quad (7.6)$$

The control gains K_1 and K_2 are obtained based on the infinite horizon LQR formulation in which the cost function,

$$J_{\text{LQ-PI}}(k) = \frac{1}{2} \sum_{k=0}^{\infty} \bar{x}^T(k)Q\bar{x}(k) + u^T(k)Ru(k), \quad (7.7)$$

is minimized. The weights Q and R in (7.7) satisfy the usual assumptions, $Q = Q^T \geq 0$, $R = R^T > 0$. The resulting closed-loop system has the form (7.1) with

$$\bar{A} = \begin{bmatrix} A - BK_1 & -BK_2 \\ C_I & I \end{bmatrix}, \quad (7.8)$$

$$\bar{x}(k) = \begin{bmatrix} x(k) \\ x_I(k) \end{bmatrix}. \quad (7.9)$$

Assuming that the feedback controller (7.6) is stabilizing, \bar{A} is Schur. The integral state x_I is a part of the controller and hence is resettable.

As in Chapter 5, a set of initial plant and integral states, $\bar{x}(0)$, for which the closed-loop trajectories of (7.1) satisfy the constraints is defined by

$$O_{\infty}^I = \{\bar{x}(0) : \bar{C}\bar{A}^k\bar{x}(0) \in Y \forall k \geq 0\}. \quad (7.10)$$

Under the assumptions that the pair (\bar{C}, \bar{A}) is observable, \bar{A} is Schur, Y is given by (7.4), compact, and $0 \in \text{int } Y$, O_{∞}^I is a positively-invariant polytope defined by a finite number of affine inequalities [58].

As in [92], the reset of the controller states is based on the approach of decreasing the closed-loop energy as defined by a Lyapunov function. For a linear system (7.1), a natural choice is

$$V(\bar{x}(k)) = \bar{x}(k)^T P \bar{x}(k), \quad (7.11)$$

where $P = P^T > 0$ is the positive-definite symmetric matrix satisfying the following Lyapunov equation

$$\bar{A}^T P \bar{A} - P + Q_L = 0, \quad (7.12)$$

and where we assume that $Q_L = Q_L^T > 0$. To satisfy constraints and guarantee recur-

sive feasibility, the state $x_I(k)$ is reset based on the solution of the following quadratic programming problem:

$$\begin{aligned} x_I^*(k) &= \arg \min_{x_I(k)} V(\bar{x}(k)) \\ \text{subject to } \bar{x}(k) &= [x(k)^T, x_I(k)^T]^T \in O_\infty^I. \end{aligned} \quad (7.13)$$

The following theorem states that the integrator reset scheme enforces constraints, assures recursive feasibility and guarantees closed-loop asymptotic stability:

Theorem 1 *Consider the discrete-time system (7.1) with a resettable controller state based on (7.13). Under the above assumptions, if a feasible solution to (7.13) exists at the initial time instant $k = 0$, then the solution exists for all future times $k \geq 0$, and the constraints (7.4) are enforced for all $k \geq 0$. Furthermore, the closed-loop system with the reset is asymptotically stable at the origin and $\bar{x}^*(k) \rightarrow 0$ as $k \rightarrow \infty$.*

Proof: Define

$$\tilde{x}(k+1) = \bar{A}\bar{x}^*(k), \quad (7.14)$$

with

$$\bar{x}^*(k) = \begin{bmatrix} x(k) \\ x_I^*(k) \end{bmatrix}, \tilde{x}(k+1) = \begin{bmatrix} \tilde{x}(k+1) \\ \tilde{x}_I(k+1) \end{bmatrix}. \quad (7.15)$$

Due to the positive invariance of O_∞^I , $\tilde{x}(k+1) \in O_\infty^I$ if $\bar{x}^*(k) \in O_\infty^I$ and a feasible choice for $x_I(k+1)$ is to set it to $\tilde{x}_I(k+1)$. Thus problem feasibility at time $k = 0$ implies its feasibility for all $k \geq 0$. Since $\bar{x}^*(k) \in O_\infty^I$, the constraints in (7.4), i.e., $G\bar{C}\bar{x}^*(k) \leq g$, are satisfied.

The stability follows from the observation

$$V(\bar{x}^*(k)) \geq V(\tilde{x}(k+1)) \geq V(\bar{x}^*(k+1)),$$

which implies that $V(\bar{x}(0)) \geq V(\bar{x}^*(0)) \geq V(\tilde{x}(1)) \geq V(\bar{x}^*(1)) \geq \dots \geq V(\bar{x}^*(k))$ for all $k \geq 0$. To prove convergence, note that

$$\begin{aligned} & V(\bar{x}^*(k+1)) - V(\bar{x}^*(k)) \\ & \leq V(\tilde{x}(k+1)) - V(\bar{x}^*(k)) \\ & = \tilde{x}(k+1)^T P \tilde{x}(k+1) - \bar{x}^*(k)^T P \bar{x}^*(k) \\ & = \bar{x}^*(k)^T \bar{A}^T P \bar{A} \bar{x}^*(k) - \bar{x}^*(k)^T P \bar{x}^*(k) \\ & = \bar{x}^*(k)^T (\bar{A}^T P \bar{A} - P) \bar{x}^*(k) \\ & = -\bar{x}^*(k)^T (Q_L) \bar{x}^*(k), \end{aligned} \quad (7.16)$$

and hence

$$V(\bar{x}^*(k)) - V(\bar{x}^*(0)) \leq - \sum_{i=0}^{k-1} \bar{x}^*(i)^T Q_L \bar{x}^*(i) \leq 0. \quad (7.17)$$

Therefore,

$$\begin{aligned} V(\bar{x}^*(0)) &\geq V(\bar{x}^*(0)) - V(\bar{x}^*(k)) \\ &\geq \sum_{k=0}^{i-1} \bar{x}^*(k)^T Q_L \bar{x}^*(k). \end{aligned} \quad (7.18)$$

for all i . Hence the series $\sum_{k=0}^{\infty} \bar{x}^*(k)^T Q_L \bar{x}^*(k)$ is convergent. Since $Q_L > 0$, it follows that $\lim_{k \rightarrow \infty} \bar{x}^*(k) = 0$.

7.2 Controller state and reference governor

The controller state and reference governor (CSRG) applies to tracking closed-loop systems of the form,

$$\begin{aligned} \bar{x}(k+1) &= \bar{A}\bar{x}(k) + \bar{B}r(k), \\ y(k) &= \bar{C}\bar{x}(k) + \bar{D}r(k), \end{aligned} \quad (7.19)$$

where $r(k) \in \mathbb{R}^q$ is the reference command modified by the the governor, $\bar{B} \in \mathbb{R}^{(n+m) \times q}$, and $\bar{D} \in \mathbb{R}^{p \times q}$. The state $\bar{x}(k)$ is of the form (7.9) and is composed of resettable components, x_I , and non-resettable components, x . For instance, (7.19) may result from replacing the integrator dynamics in (7.5) by

$$x_I(k+1) = x_I(k) + C_I x(k) - r(k). \quad (7.20)$$

Let \bar{r} denote the nominal reference command to the controller state and reference governor. See Figure 7.1.

The matrix \bar{A} is assumed to be Schur, and let $\bar{x}_e = Hr$ denote the equilibrium state corresponding to a constant command $r(k) \equiv r$.

Consider the set

$$\begin{aligned} O_{\infty}^r &= \{(\bar{x}(0), r) : (\bar{C}H + \bar{D})r \in (1 - \epsilon)Y, \\ &r(k) = r \forall k \geq 0 \Rightarrow y(k) \in Y \forall k \geq 0\}, \end{aligned} \quad (7.21)$$

where $\epsilon > 0$ is sufficiently small. The set O_{∞}^r is the set of all initial states, $\bar{x}(0)$, and

constant reference commands r that lead to responses satisfying the constraints. For technical reasons [102], the set of admissible constant commands, r , is tightened slightly from $(\bar{C}H + \bar{D})r \in Y$ to

$$(\bar{C}H + \bar{D})r \in (1 - \epsilon)Y. \quad (7.22)$$

The technique used to calculate O_∞^r is described in Appendix B.

The controller state and reference governor resets $x_I(k)$ and determines $r(k)$ based on the minimization of the following cost function

$$J(\bar{x}(k), r(k)) = \|\bar{x}(k) - \bar{x}_e(k)\|_P^2 + \|r(k) - \bar{r}(k)\|_\Gamma^2, \quad (7.23)$$

where for a matrix M , $\|z\|_M^2 = z^T M z$, $\Gamma = \Gamma^T > 0$ and $P = P^T > 0$ satisfies the Lyapunov equation (7.12) with $Q_L > 0$. The optimization problem is a quadratic program similar to (7.13),

$$\begin{aligned} (\bar{x}^*(k), r^*(k)) &= \arg \min_{x_I(k), r(k)} J(\bar{x}(k), r(k)) \\ \text{subject to } &(\bar{x}(k), r(k)) \in O_\infty^r, \bar{x}(k) = [x(k)^T, x_I(k)^T]^T \end{aligned} \quad (7.24)$$

The convergence properties of the controller state and reference governor are characterized by the following result:

Theorem 2 *Suppose that \bar{r} is a constraint-admissible constant reference command with $(H\bar{r}, \bar{r}) \in O_\infty^r$. Consider the discrete-time system (7.19) with $x_I(k)$ and $r(k)$ determined based on (7.24). Under the above assumptions, if a feasible solution to (7.24) exists at the initial time instant $k = 0$, the solution exists for all future times $k \geq 0$, and the constraints (7.4) are enforced for all $k \geq 0$. Furthermore, the closed-loop system equilibrium at $H\bar{r}$ is asymptotically stable, and there exists a finite k^* such that $r(k) = \bar{r}$ for $k \geq k^*$.*

Proof: The proof follows similar steps as the proof found in [88] for the extended command governor case. Define,

$$J^*(k) = \|\bar{x}^*(k) - \bar{x}_e^*(k)\|_P^2 + \|r^*(k) - \bar{r}\|_\Gamma^2, \quad (7.25)$$

where $(\bar{x}^*(k), r^*(k))$ are the solutions to (7.24) and $\bar{x}_e^*(k) = Hr^*(k)$. Define also

$$\begin{aligned} \tilde{J}(k) &= \|\bar{A}\bar{x}^*(k-1) + \bar{B}r^*(k-1) - \bar{x}_e^*(k-1)\|_P^2 \\ &+ \|r^*(k-1) - \bar{r}\|_\Gamma^2, \end{aligned} \quad (7.26)$$

which corresponds to evaluating the cost on a feasible (feasible by positive invariance prop-

erties of O_∞^r for constant commands) solution, $(\bar{A}\bar{x}^*(k-1) + \bar{B}r^*(k-1), r^*(k-1))$. Since P satisfies (7.12) with $Q_L > 0$, it follows that $\tilde{J}(k) \leq J^*(k-1)$ and, hence,

$$J^*(k) \leq \tilde{J}(k) \leq J^*(k-1).$$

Consequently, $J^*(k)$ is a monotonically non-increasing sequence, hence it converges $J^*(k) \rightarrow J_{op}^* \geq 0$ as $k \rightarrow \infty$.

We exploit the following result:

Lemma 1 ([88]): Suppose $Z \in \mathbb{R}^q$ is closed and convex, $z_s \in \mathbb{R}^q$, $Q \in \mathbb{R}^{q \times q} > 0$, and $z^* = \arg \min_{z \in Z} \|z - z_s\|_Q^2$. Then

$$\|z - z^*\|_Q^2 \leq \|z - z_s\|_Q^2 - \|z^* - z_s\|_Q^2 \quad \forall z \in Z.$$

Considering $\bar{x}^*(k)$, $r^*(k)$ as the minimum norm projection on a closed and convex set O_∞^r , and applying Lemma 1, it follows that

$$\begin{aligned} & \|\bar{A}\bar{x}^*(k-1) + \bar{B}r^*(k-1) - \bar{x}^*(k)\|_P^2 + \|r^*(k-1) - r^*(k)\|_\Gamma^2 \\ & \leq \tilde{J}(k) - J^*(k). \end{aligned} \quad (7.27)$$

Since $\tilde{J}(k) - J^*(k) \rightarrow 0$ as $k \rightarrow \infty$, it follows that $\bar{A}\bar{x}^*(k-1) + \bar{B}r^*(k-1) \rightarrow \bar{x}^*(k)$ and $r^*(k) - r^*(k-1) \rightarrow 0$ as $k \rightarrow \infty$. Hence, $\bar{x}^*(k) = \bar{A}\bar{x}^*(k-1) + \bar{B}r^*(k-1) + \epsilon_1(k-1)$, $r^*(k) = r^*(k-1) + \epsilon_2(k-1)$, with $\epsilon_1(k) \rightarrow 0$, $\epsilon_2(k) \rightarrow 0$ as $k \rightarrow \infty$. Since \bar{A} is Schur, it follows that $\bar{x}^*(k) \rightarrow Hr^*(k)$. If $J_{op}^* > 0$, based on (7.25) $r^*(k)$ does not converge to \bar{r} , yet $(\bar{x}^*(k), r^*(k)) \in \text{int}O_\infty^r$. Assuming, without loss of generality, that $\Gamma = I$, consider

$$\tilde{r}(k) = r^*(k) + \delta \frac{\bar{r} - r^*(k)}{\|\bar{r} - r^*(k)\|}.$$

It can be shown, utilizing properties of O_∞^r , (7.22), and $\bar{x}^*(k) \rightarrow Hr^*(k)$ as $k \rightarrow \infty$, that there exists $\delta > 0$ such that $(\bar{x}^*(k), \tilde{r}(k))$ is feasible for all k sufficiently large. Moreover, $J(\bar{x}^*(k), \tilde{r}(k)) < J(\bar{x}^*(k), r^*(k))$ which contradicts $\bar{x}^*(k), r^*(k)$ being the optimal solution unless $r^*(k) = \bar{r}$. The proof of the latter assertion follows by demonstrating, based on the Taylor series expansion, and with H.O.T. denoting Higher Order Terms, that for $\delta > 0$

small,

$$\begin{aligned}
& J(\bar{x}^*(k), \tilde{r}(k)) - J(\bar{x}^*(k), r^*(k)) \\
&= \nabla_r J(\bar{x}^*(k), r^*(k))[\tilde{r}(k) - r^*(k)] + H.O.T. \\
&= 2(\bar{x}^*(k) - Hr^*(k))P(-H)(\tilde{r}(k) - r^*(k)) \\
&\quad + 2(\tilde{r}(k) - r^*(k))^T(\tilde{r}(k) - r^*(k)) + H.O.T.,
\end{aligned}$$

where in the first term the coefficient $2(\bar{x}^*(k) - Hr^*(k))P(-H)$ multiplying $(\tilde{r}(k) - r^*(k))$ converges to 0 as $k \rightarrow \infty$ while the second term dominates and gives the decrease in J for large k as $2(\tilde{r}(k) - r^*(k))^T(\tilde{r}(k) - r^*(k)) = -2\delta\|r^*(k) - \bar{r}\|$

7.3 Remarks

7.3.1 Linear quadratic proportional plus integral controller

Consider the stabilization problem for system (7.5) with the linear quadratic proportional plus integral controller (7.6) minimizing the cost (7.7) and with the constrained output (7.3). Note that in the unconstrained case, the optimal cost is given by

$$J = \frac{1}{2}\bar{x}(k)^T \hat{P}\bar{x}(k), \quad (7.28)$$

where \hat{P} is the solution of the Discrete-Time Algebraic Riccati Equation (DARE). By Bellman's optimality principle, the control $u(k)$, generated by (7.6), minimizes the following cost

$$J^+ = \frac{1}{2}\bar{x}(k+1)^T \hat{P}\bar{x}(k+1) + \frac{1}{2}u(k)^T Ru(k). \quad (7.29)$$

One approach to implementing the controller state governor in the constrained case is to minimize the same cost (7.29) with respect to $x_I(k)$ subject to the constraint

$$\bar{x}(k) = [x(k)^T, x_I(k)^T]^T \in O_\infty^I.$$

After algebraic manipulations and omitting terms that do not depend on x_I , it follows that minimizing (7.29) is equivalent to minimizing

$$\tilde{J}^+ = \frac{1}{2}x_I(k)^T (F_2^T \hat{P} F_2 + S_2^T R S_2)x_I(k) + (F_1^T \hat{P} F_2 + S_1^T R S_2)x_I(k), \quad (7.30)$$

where

$$F_1 = \begin{pmatrix} Ax(k) - BK_1x(k) \\ C_Ix(k) \end{pmatrix}, \quad F_2 = \begin{pmatrix} -BK_2 \\ I \end{pmatrix}, \quad S_1 = K_1x(k), \quad S_2 = K_2.$$

The constrained minimization of (7.30) can be performed by an on-board quadratic programming solver or explicitly using multi-parametric programming techniques [111], giving a feedback law for the integrator state,

$$x_I^*(k) = U^*(x(k)). \quad (7.31)$$

Note that in absence of disturbances, the optimization problem needs to be solved only once at $k = 0$. Indeed, by positive invariance properties of O_∞^I , if $\bar{x}^*(k) \in O_\infty^I$, then the one step ahead propagated state,

$$\tilde{\bar{x}}(k+1) = \bar{A}\bar{x}^*(k),$$

is feasible and the resulting $\tilde{x}_I(k+1)$ coincides with $x_I^*(k+1)$ due to the special choice of the cost function in (7.30). Thus the controller state is reset once and then the optimal action of the unconstrained linear quadratic optimal controller is recovered.

Consider now the design of a linear quadratic proportional-plus-integral controller for set-point tracking. For a linear system model given by,

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y_r(k) &= C_r x(k). \end{aligned} \quad (7.32)$$

and tracking error defined as

$$e(k) = C_r x(k) - r, \quad (7.33)$$

where r is the set-point, the so called rate-based model is defined as

$$\begin{aligned} e(k+1) &= C(x(k+1) - x(k)) + e(k) = C\Delta x(k) + e(k), \\ \Delta x(k+1) &= x(k+1) - x(k) = A\Delta x(k) + B\Delta u(k), \end{aligned} \quad (7.34)$$

where

$$\Delta x(k) = x(k+1) - x(k), \quad \Delta u(k) = u(k+1) - u(k).$$

The linear-quadratic proportional-plus-integral controller has the form,

$$\Delta u(k) = K_I e(k) + K_P \Delta x(k), \quad (7.35)$$

and minimizes the cost,

$$J = \frac{1}{2} \sum_{k=0}^{\infty} e(k)^T Q_e e(k) + \Delta u(k)^T R \Delta u(k).$$

For implementation, (7.35) is re-written as

$$u(k) = K_p x(k) + K_I x_I(k), \quad x_I(k+1) = x_I(k) + e(k), \quad (7.36)$$

where $x_I(k)$ is the state of the integrator.

Based on the solution, P , of the DARE and Bellman's optimality principle, (7.35) minimizes the cost

$$J^+ = \frac{1}{2} \begin{pmatrix} e(k+1) \\ \Delta x(k+1) \end{pmatrix}^T P \begin{pmatrix} e(k+1) \\ \Delta x(k+1) \end{pmatrix} + \frac{1}{2} \Delta u(k)^T R \Delta u(k). \quad (7.37)$$

Omitting terms that do not depend on $x_I(k)$ or r , minimizing J^+ is equivalent to minimizing

$$\begin{aligned} \tilde{J}^+(\bar{x}(k), r) = & \frac{1}{2} x_I(k)^T (F_2^T P F_2 + S_2^T R S_2) x_I(k) + \\ & \frac{1}{2} r^T (L_1^T P L_1 + L_2^T R L_2) r + r^T (L_1^T P F_2 + L_2^T) x_I(k) + \\ & F_1^T P F_2 x_I(k) + S_1^T R S_2 x_I(k) + F_1^T P L_1 r + S_1^T R L_2 r, \end{aligned} \quad (7.38)$$

where

$$F_1 = \begin{pmatrix} C_r A x(k) + C_r B K_P x(k) \\ A x(k) + B K_P x(k) - x(k) \end{pmatrix},$$

$$F_2 = \begin{pmatrix} C_r B K_I \\ B K_I \end{pmatrix},$$

$$S_1 = (K_I C_r + K_P A + K_P B K_P - K_P) x(k), \quad S_2 = K_P B K_I,$$

$$L_1 = \begin{pmatrix} -I \\ 0 \end{pmatrix}, \quad L_2 = -K_I.$$

The controller state and reference governor can now be defined by replacing the cost (7.23) with the cost

$$J(\bar{x}(k), r(k)) = \tilde{J}^+(\bar{x}(k), r) + \|r(k) - \bar{r}(k)\|_I^2. \quad (7.39)$$

Formally, (7.39) has a different form than (7.23).

7.3.2 Nonlinear systems

The proposed approach of resetting the state of the dynamic controller is applicable to nonlinear systems, [112]. Consider, for instance, the problem of stabilization to the origin.

Suppose the model of the open-loop system has the form,

$$x(k+1) = f(x(k), u(k)), \quad (7.40)$$

and the controller is given by

$$\begin{aligned} u(k) &= U(x(k), x_I(k)), \\ x_I(k+1) &= g(x(k), x_I(k)), \end{aligned} \quad (7.41)$$

with the resettable dynamic controller state $x_I(k)$. Let the constraints be given by

$$y(k) = h(x(k)) \in Y.$$

Let $\bar{x}(0) = [x^T(0), x_I^T(0)]^T$ and

$$O_\infty^I = \{\bar{x}(0) : y(k) \in Y \ \forall k \geq 0\}.$$

Note that O_∞^I is positively-invariant.

Let $V(\bar{x})$ be a Lyapunov function verifying asymptotic stability of the closed-loop system. To satisfy constraints and guarantee recursive feasibility, the state $x_I(k)$ can be reset based on the solution of the optimization problem,

$$\begin{aligned} x_I^*(k) &= \arg \min_{x_I(k)} V(\bar{x}(k)) \\ \text{subject to } \bar{x}(k) &= [x(k)^T, x_I(k)^T]^T \in O_\infty^I. \end{aligned} \quad (7.42)$$

Following the same line of reason as in Section 7.1, a result similar to Theorem 1 can be obtained. The main challenge from a computational standpoint is obtaining O_∞^I . Note that O_∞^I can be replaced by its positively invariant subset S which may be constructed as a constraint-admissible sublevel set of V . The use of non positively invariant subsets, S , of O_∞^I is also possible; in this case, (7.42) may become infeasible at certain time instants (feasibility at the initial time needs to be assumed), in which case the update (7.41) can simply be applied at these time instants. The stability is still maintained with such an approach. The use of non-positively invariant subsets of O_∞^I can lead to computational simplifications.

7.4 Constrained aircraft flight control

In this example of implementing the CSRG for aircraft flight control, the linearized models and constraints in Section 5.4 are used. The control architecture is designed as follows.

For the lateral model, C_I in (7.20) is selected as

$$C_{I,\text{lat}} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (7.43)$$

and for the longitudinal model,

$$C_{I,\text{lon}} = \begin{bmatrix} 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \quad (7.44)$$

To generate a stable closed-loop controller, an LQR problem is formulated with the following weights: $Q_{\text{lat}} = \text{diag}([1 \ 1 \ 1 \ 1 \ 100 \ 100])$, $R_{\text{lat}} = \text{diag}([1 \ 1])$, $Q_{\text{lon}} = \text{diag}([1 \ 1 \ 1 \ 1 \ 100 \ 100])$, and $R_{\text{lon}} = \text{diag}([1 \ 1])$.

For both the lateral and longitudinal dynamics, \bar{B} is of the form

$$\bar{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad (7.45)$$

\bar{C} is the identity, and $\bar{D} = 0$. The cost function (7.23) was developed for both the longitudinal and lateral closed-loop dynamics. In each case, P is the solution to (7.12), $Q_L = I$, for the respective closed-loop dynamics and $\Gamma = I$. The imposed constraints are the same as found in Section 5.4.

The selected initial conditions were $\bar{x}(0)_{\text{lon}} = [6, 0.1, 0.1, -0.1, 0, 0]^T$ with the two references set initially to zero and $\bar{x}(0)_{\text{lat}} = [0, 0, 0.2, 1, 0, 0]^T$ with the two references set initially to zero. In both cases, $\bar{r} = 0$. Figures 7.2 through 7.4 provide the results for the lateral dynamics and Figures 7.5 through 7.7 provide the results for the longitudinal dynamics. Note that in both cases, the trajectories of $r(t)$ converge to zero in finite time. In Figures 7.2 and 7.5, which are projections of various sets onto two-dimensions, a set labeled as " $O_\infty^I, x_I(0) = 0$ " is displayed. This set is the projected cross section of O_∞^I when the integrator is initially zero. Figures 7.2-7.4 show the lateral state trajectories and control and

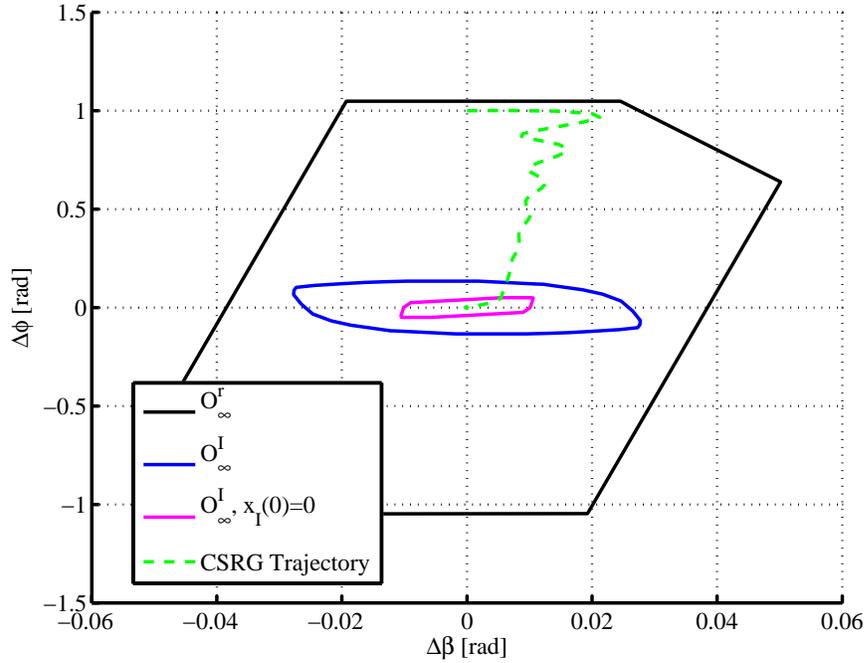


Figure 7.2: Projection of the lateral O_∞^I , O_∞^I with $x_I(0) = 0$, and O_∞^r sets in the $\Delta\beta$ - $\Delta\phi$ plane with sample state trajectory.

reference time histories while Figures 7.5-7.7 show the same for the longitudinal dynamics. Note that $O_\infty^I \subset O_\infty^r$ in both Figures 7.2 and 7.5.

7.5 Constrained gas turbine engine control

The gas turbine engine model has two states, the fan speed (N_f) and the core speed (N_c), and a single control input which is the fueling rate (W_f). The linearization of the gas turbine engine model of [113] at the altitude of 20000 ft, 0.5 Mach and 60 percent PLA was used for the simulations. The constraints are imposed on the Low Pressure Compressor (LPC) and the High Pressure Compressor (HPC) surge margins such that they cannot decrease below 2.5 percent, and the engine fueling rate has to be maintained above 0.1 to avoid the combustion blowout. The linearized continuous-time model used is

$$\begin{bmatrix} \Delta \dot{N}_f \\ \Delta \dot{N}_c \end{bmatrix} = \begin{bmatrix} -1.7980 & 0.7177 \\ 0.1503 & -1.6692 \end{bmatrix} \begin{bmatrix} \Delta N_f \\ \Delta N_c \end{bmatrix} + \begin{bmatrix} 704.3 \\ 1401.4 \end{bmatrix} \Delta W_f, \quad (7.46)$$

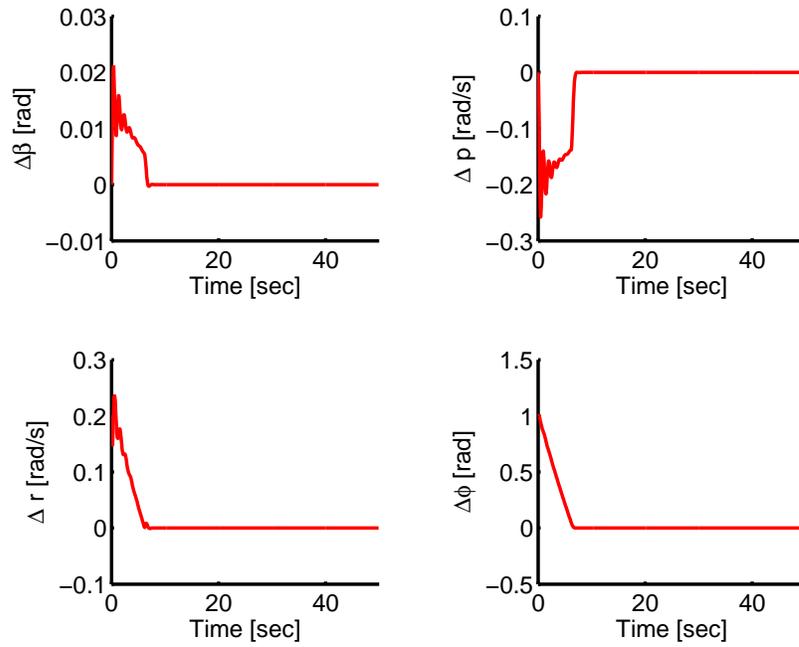


Figure 7.3: Lateral state trajectories of the sample trajectory in Figure 7.2.

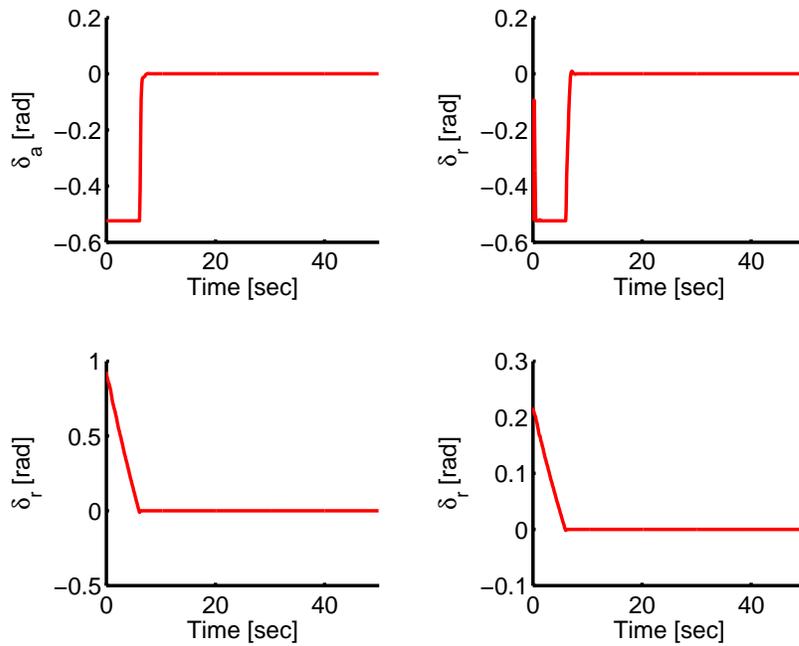


Figure 7.4: Lateral reference and control time histories of the sample trajectory in Figure 7.2.

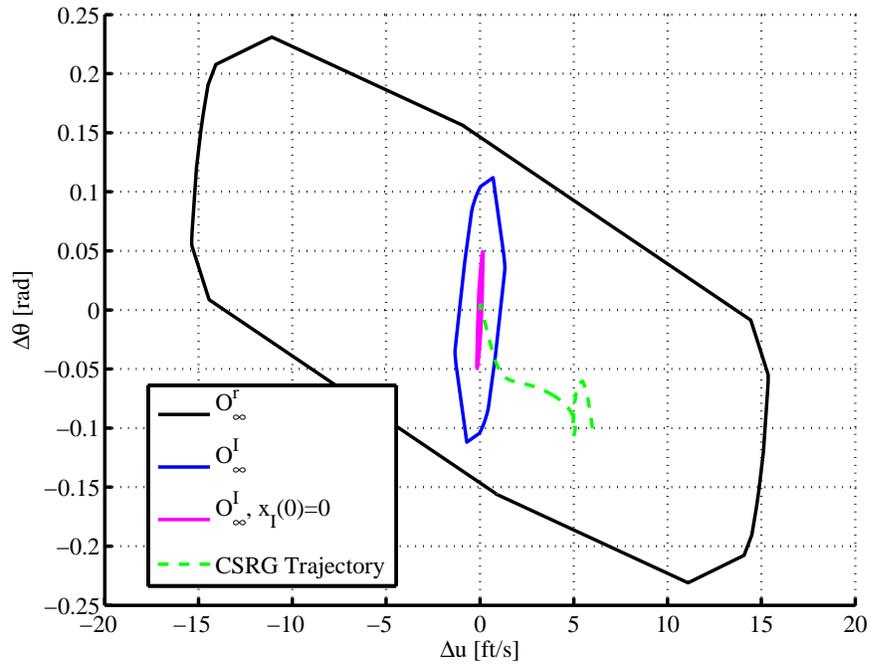


Figure 7.5: Projection of the longitudinal O_∞^I , O_∞^I with $x_I(0) = 0$, and O_∞^F sets in the Δu - $\Delta\theta$ plane with sample state trajectory.

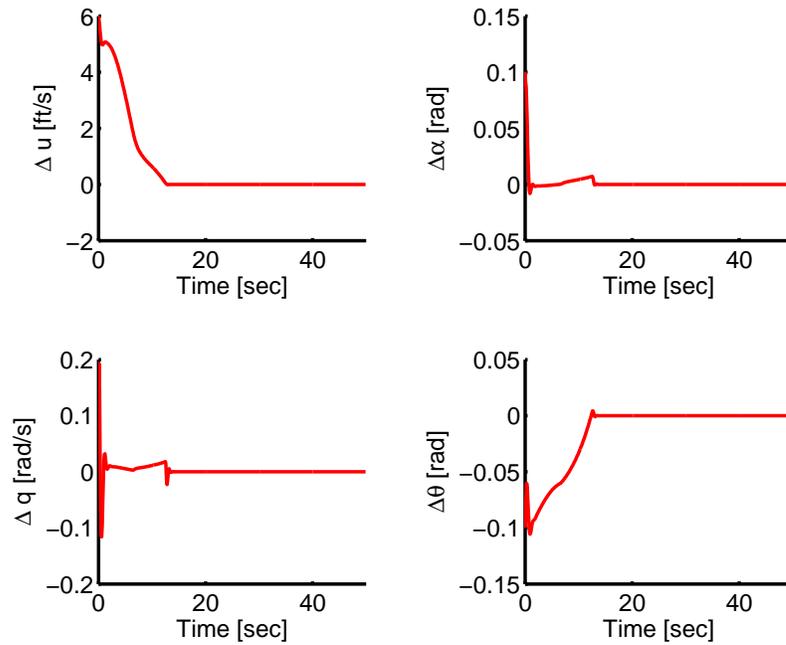


Figure 7.6: Longitudinal state trajectories of the sample trajectory in Figure 7.5.

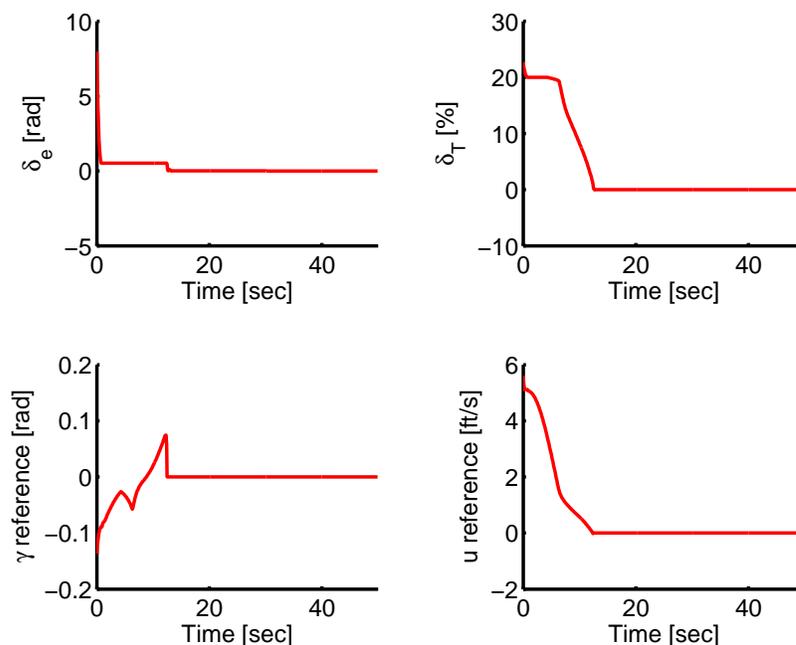


Figure 7.7: Longitudinal reference and control time histories of the sample trajectory in Figure 7.5.

with

$$C_I = \begin{bmatrix} 1 & 0 \end{bmatrix}. \quad (7.47)$$

The model was converted to discrete-time with the sampling period of $T_s = 0.1$ sec.

The LQ-I tracking controller of the form (7.36) was designed for W_f to track a fan-speed set-point, \bar{r} . The weights used were $Q = \text{diag}([1, 0, 0])$ and $R = 1000$.

Figure 7.8 illustrates a substantial increase in the size of the set of recoverable values of $N_c(0)$ and $N_f(0)$ if the integrator state can be reset freely versus if it is set to 0. The domain of recoverable states in the former case is that of the controller state and reference governor and is much larger than that of the reference governor alone in the latter case.

Figures 7.9-7.13 illustrate the response of the closed-loop system with the controller state and reference governor. The constraints are strictly enforced by appropriately setting the initial value of the integrator (Figure 7.10), and finite time convergence of $r(t)$ to constraint-admissible values of \bar{r} is observed. Note that the initial state is such that initializing $x_I(0) = 0$ will lead to responses violating constraints.

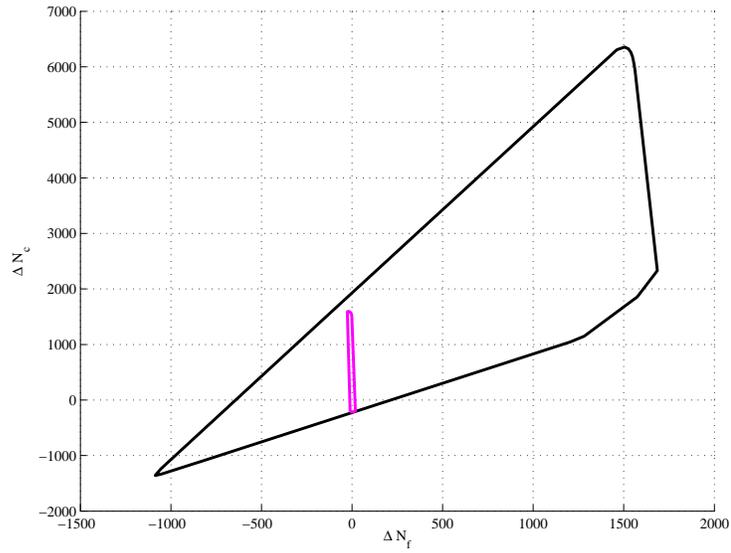


Figure 7.8: The projection of the set O_∞^r on ΔN_f - ΔN_c plane with x_I resettable (larger set) and with $x_I = 0$ (smaller set). The variables ΔN_f , ΔN_c denote the deviations from the equilibrium.

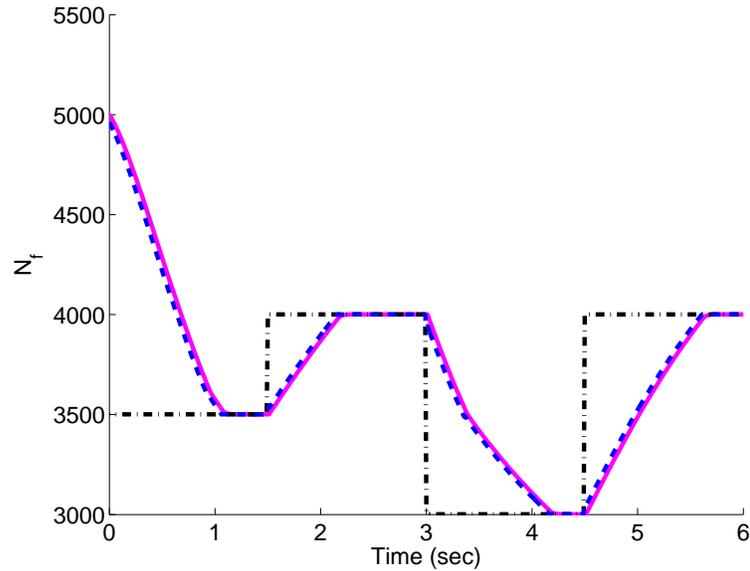


Figure 7.9: The time histories of the set-point \bar{r} for N_f (dash-dotted), governed set-point r and actual N_f .

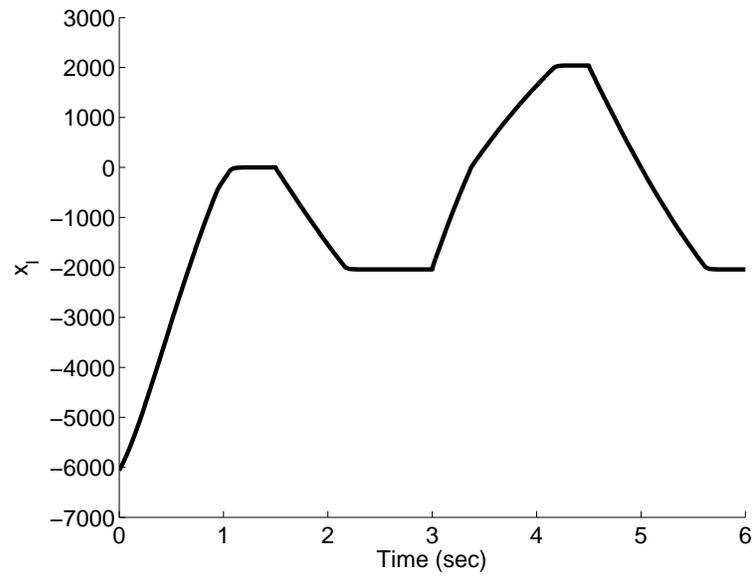


Figure 7.10: The time history of the state of the integrator.

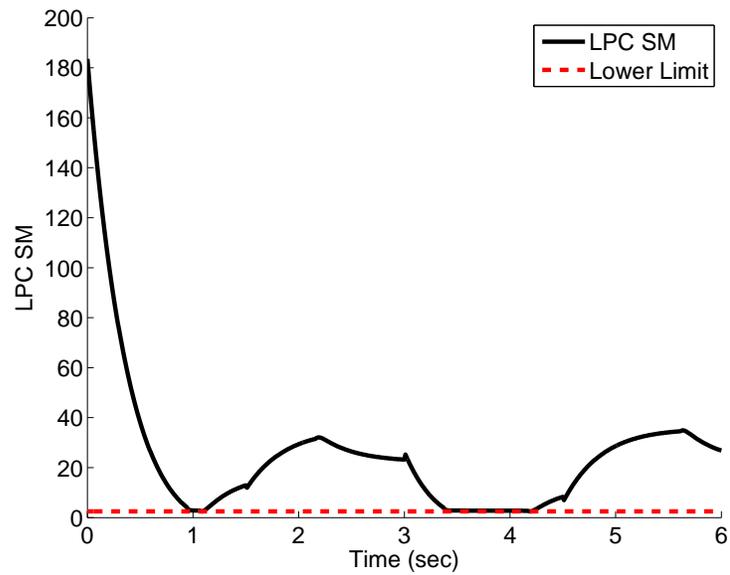


Figure 7.11: The time history of the LPC surge margin (solid) and constraint (dashed).

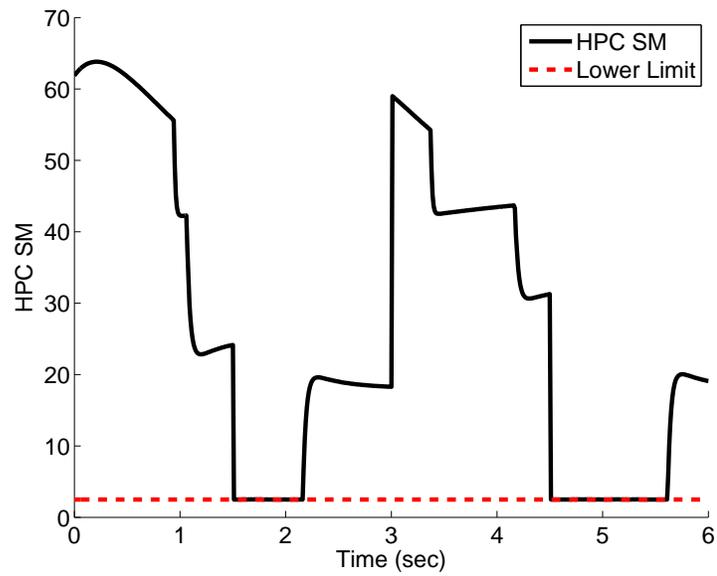


Figure 7.12: The time history of the HPC surge margin (solid) and constraint (dashed).

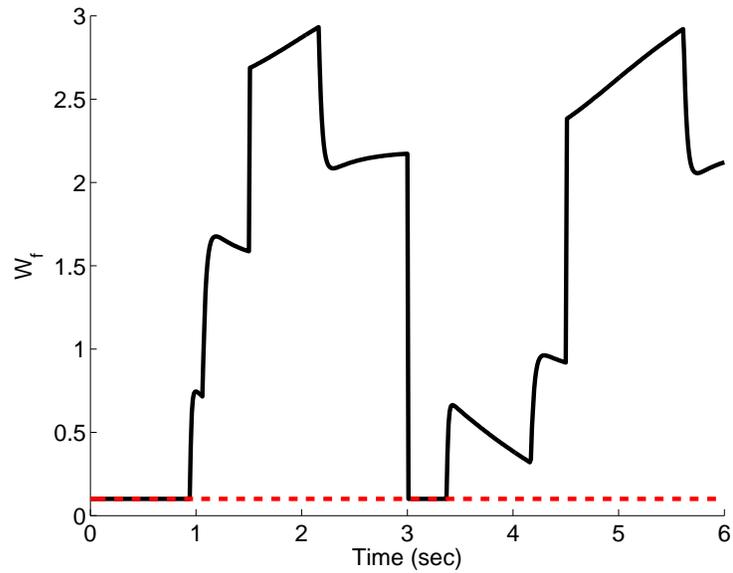


Figure 7.13: The time history of the engine fueling rate (solid) and constraint (dashed).

CHAPTER 8

Summary, Conclusions, and Future Work

8.1 Summary and conclusions

This dissertation presented developments and advancements in the area of automotive fuel-efficient cruise control and in the area of constrained control with aircraft applications.

- A stochastic dynamic programming (SDP) approach to the optimization of vehicle speed control policies was developed. Results for both simulation and in-vehicle testing were presented indicating that SDP policies can produce vehicle speed profiles that reduce, on average, fuel consumption. In particular, in following a vehicle moving at constant speed under relative distance constraints, these policies have led to time-varying vehicle speed profiles. These time-varying speed profiles resemble periodic cruise or pulse and glide (PnG) strategies reported in previous literature and were validated as being more fuel efficient than driving at a constant average speed in our actual vehicle experiments. The vehicle testing demonstrated an average fuel economy improvement of 4.51% (with benefit reduced to 2.74% if average speed reduction is accounted for) over twelve drives, with a maximum improvement of 11.58%.
- The rule based cruise control (RBCC) scheme was developed and the method by which its transition thresholds are optimized was defined. The RBCC was tested through simulation and vehicle experiments. It was shown that by breaking the time-varying speed profiles into four distance phases (normal, glide, recover, and bleed) and mapping these phases into states of a finite state machine, a practical implementation for a time-varying, PnG like, cruise strategy can be achieved. The RBCC vehicle tests demonstrated an average dynamic fuel economy improvement (a fuel economy metric that discounts the fuel economy gained from driving at a slower average speed) of 2.76% with a maximum of 5.47%.

- Procedures for stochastic modeling of traffic conditions (road grade and traffic speed) and methods for stochastic model characterization were developed. It was shown that Markov chain models of road grade and traffic speed can be categorized, using the KL divergence rate, based on attributes of the data including where it was collected and the driving style of the vehicle operator. Also, a novel technique for predicting step changes in road grade modeled as a stochastic jump process was demonstrated. This prediction method is more accurate when the prediction horizon is less than or equal to 20 meters as compared to the Markov chain model prediction method. Finally, it was demonstrated that the Markov chain models and the stochastic jump models can be learned online and the KL divergence can be used to test model convergence.
- Recoverable and integral safe sets were introduced, defined, and developed for aircraft applications. A comparison of projections of safe, recoverable, and integral safe sets demonstrated that larger constrained domains of attraction are available through the use of recoverable and integral safe sets. An aircraft maneuver planning example demonstrated the utility of integral safe sets for producing constraint-admissible trajectories through an aircraft's trim envelope.
- Two scaling transformations that exploit scalar and vector scaling, respectively, were developed for safe, recoverable, or integral safe sets. These transformations were motivated by the need for fast estimation of safe, recoverable, and integral safe sets at different trim conditions during a loss of control event. These scaling transformations were demonstrated through an aircraft icing example which suggested that they can be useful to estimate subsets of other trim conditions' recoverable sets.
- The controller state governor (CSG) and controller state and reference governor (CSRG) control schemes were developed for control of systems with constraints. These control schemes take advantage of a resettable controller state (e.g., an integrator state) to enforce constraints, reduce closed-loop system energy, and improve system performance. The CSRG additionally uses an adjustable reference for constraint enforcement which further increases the constrained domain of attraction. A proof of maintenance of stability under dynamic controller reset for the CSG was presented and a proof of finite-time convergence of the CSRG was also presented. Two aircraft examples were used to demonstrate the effectiveness of these control schemes.

8.2 Future work

Many opportunities for future work exist motivated by the developments in this dissertation.

8.2.1 Stochastic fuel-efficient cruise control

- *Simultaneous treatment of traffic speed, road grade, and relative distance between vehicles as states in the SDP.* Due to computational reasons, we have been able to generate the control policies for only two less general cases. It is of interest to generate a single control policy, potentially using approximate dynamic programming [114].
- *Extending the generated SDP policies and their analysis to a large geographical area, possibly the entire United States.* The current developments were limited by available data around M-39 near Dearborn, Michigan. Future research needs to extend the area of coverage of the policies and provide insight into sensitivity to geographical areas. To this end, road topographical information databases and traffic speed measurements from multiple sources can be utilized to help produce the necessary information for the stochastic models.
- *The application of the SDP and RBCC techniques on hybrid and electric powertrains.* The integration of vehicle speed control into hybrid electric vehicle energy management and integrated powertrain control more generally may lead to further fuel economy improvements. As an example, the pulsing of the engine appears to produce better fuel economy (hence the PnG behaviors). However, vehicle acceleration and increased aerodynamic drag reduce the improvements. If the engine pulsing can be used to charge a energy storage device (batteries perhaps) while the vehicle speed is maintained, and then the energy storage device discharges energy on the down slope of the pulse, it may be possible to generate an engine pulsing behavior that is more fuel-efficient without altering the vehicle speed outside of the normally allowed variance using standard cruise control.
- *The application of the stochastic jump prediction method to model step changes in traffic speed.* By using the stochastic jump process to predict step changes in road grade, improved prediction accuracies were obtained for prediction horizons less than 20 meters as compared to the Markov chain model prediction method. It is expected that similar results may be obtained for the short distance traffic speed prediction. It

is also of interest to understand how to incorporate these jump process models into SDP optimization of vehicle speed control policies.

8.2.2 Constrained control with aircraft applications

- *Completing additional case studies for specific aircraft impairments where recoverable and integral safe sets are used for planning a constraint-admissible aircraft trajectory.* Such case studies may consider wing icing or rudder jam simulations.
- *Further research on the CSG and CSRG.* In particular, similar stability and convergence results in the case of nonlinear systems appears tractable.
- *Developing procedures for fast construction of recoverable and integral safe sets for certain classes of nonlinear systems.* The focus in this dissertation has been on exploiting discrete-time linear models. Extending the results to the case when the model also includes a nonlinearity by taking advantage of bounds on the nonlinearities and of convex optimization techniques appears feasible following related developments in the literature, see, e.g., [115].
- *The formulation of safe, recoverable, and integral safe sets to include the full twelve dimensional aircraft dynamics model so that objects and obstacles can be included as constraints.* This could be of particular interest in the integration of unmanned aerial systems (UAS) into the existing airspace for the purpose of mitigating collisions between UAS and other aircraft.
- *The formulation of recoverable integral sets and associated control schemes.* In this case, the recovery sequences would be implemented in conjunction with resettable controller state and adjustable references. We expect that this extension is relatively straight forward theoretically and computationally but may produce practical advantages, e.g., increase constrained domains of attraction.
- *Convergence properties of R_∞ to V_∞ .* We hypothesize that as the dimensionality of the auxiliary subsystem grows (in particular, in the shift register case) the recoverable set R_∞ will converge to the viability domain, V_∞ , in the Hausdorff norm sense. The analysis of this convergence and the derivation of specific results is left to future work.
- *Perform nonlinear aircraft simulations using the CSG/CSRG with gain scheduling and compare these results with the nonlinear maneuvering envelopes.* While the

boundaries of the safe, recoverable, and integral safe sets we generate have been confirmed by linearized model simulations, confirming these boundaries by nonlinear simulations, and comparing these boundaries with maneuvering envelopes like those found in [71, 74], is left to future work [116].

APPENDIX A

Auto-Regressive Grade Prediction Technique

In this appendix, the auto-regressive grade prediction method, developed by Dr. Tony D'Amato of the Ford Motor Company, used in Chapter 3 for forecasting road grade, $\theta(k)$, r -steps into the future is presented. Assume the road grade propagates according to

$$\theta(k + 1) = \delta(k)\phi(k),$$

where $\delta(k) \in \mathbb{R}^{1 \times n}$ is a vector of model parameters, and $\phi(k) \in \mathbb{R}^n$ is a vector of previous grades from time $k - n + 1$ to k .

Next, let r , a positive integer, be the width of the prediction horizon. Then the measurement $\theta(k + r)$, is obtained in state space form as

$$\begin{bmatrix} \theta(k + 1) \\ \vdots \\ \theta(k - n + 2) \end{bmatrix} = A(k)\phi(k), \quad (\text{A.1})$$

where

$$A(k) = \begin{bmatrix} \delta(k) \\ A_{12} \end{bmatrix} \in \mathbb{R}^{n \times n},$$

reflects the measurement dynamics, and

$$A_{12} \triangleq [I \ 0] \in \mathbb{R}^{(n-1) \times n}.$$

Next, it follows from A.1 that the road grade at $k + r$ is

$$\theta(k + r) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \prod_{i=1}^r A(k + i - 1)\phi(k). \quad (\text{A.2})$$

Assuming that $A(k)$ evolves slowly over time, i.e., for $i = 1, \dots, r$, $A(k+i) \approx A(k)$, and A.2 is rewritten as

$$\theta(k+r) \approx \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} A^r(k)\phi(k).$$

The objective is then to obtain an estimate, $\hat{A}(k)$, of $A(k)$ in order to calculate the estimate $\theta(k+r)$ of the future road grade measurement $\theta(k+r)$.

Let $\hat{\theta}(k)$ be an estimate of the road grade $\theta(k)$, then

$$\hat{\theta}(k+r) = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \hat{A}^r(k)\phi(k),$$

where

$$\hat{A}(k) = \begin{bmatrix} \hat{\delta}(k) \\ A_{12} \end{bmatrix} \in \mathbb{R}^{n \times n},$$

and $\hat{\delta}(k) \in \mathbb{R}^{1 \times n}$ is an estimate of $\delta(k)$. We compute $\hat{\delta}(k)$ using the recursive least squares update

$$\begin{aligned} \hat{\delta}(k) \triangleq & \hat{\delta}(k-1) + \beta(k)[\hat{\delta}(k-1)\phi(k-1) - \theta(k-1)] \\ & \cdot [\phi^T P(k-1)\phi(k-1) + \lambda]^{-1} \\ & \cdot \phi^T(k-1)P(k-1), \end{aligned} \tag{A.3}$$

where $P(0) \in \mathbb{R}^{n \times n}$ is symmetric positive definite, $\lambda \in (0, 1]$ is the forgetting factor, $\beta(k)$ is either 1 or 0 to enable or disable updates of the form

$$\begin{aligned} P(k) = & \lambda^{-1}P(k-1) - \beta(k)\lambda^{-1}P(k-1)\phi(k-1) \\ & \cdot [\phi^T P(k-1)\phi(k-1) + \lambda]^{-1} \\ & \cdot \phi^T(k-1)P(k-1). \end{aligned} \tag{A.4}$$

We initialize $P(0) = \alpha I$, where $\alpha > 0$.

APPENDIX B

Computation of Invariant Sets

This appendix describes the methods used to compute the safe, recoverable, integral safe, set-bounded disturbance safe/recoverable, and reference integral safe sets. Three methods will be described. The first, in Section B.1, is adapted from [83] and is used to calculate safe, recoverable, and integral safe sets. The second, in Section B.2, is adapted from [79] and is used to calculate the safe set under additive set-bounded disturbances (the disturbance safe/recoverable set). The final method, in Section B.3, is used to calculate the reference integral safe set and was used in [96].

B.1 Calculation of safe, recoverable, and integral safe sets

The calculation of safe, recoverable, and integral safe sets is now described. The following generically describes the process for the calculation of a safe set resulting from constrained closed-loop dynamics, but the process is the same for the closed-loop dynamics considered for O_∞^{ext} and O_∞^I . The following technique is adapted from [83], but note that this is not the only method for calculating safe, recoverable, and integral safe sets.

Consider a closed-loop system of the following form

$$\begin{aligned}x(k+1) &= Ax(k) \\ y(k) &= Cx(k)\end{aligned}\tag{B.1}$$

where A models the stable closed-loop dynamics of the system and $y(k)$ is the output with $x(k) \in \mathbb{R}^{n_x}$ and $y(k) \in \mathbb{R}^{n_y}$. We constrain $y(k) \in Y$ with Y being the output constraint set defined as

$$Y = \{y(k) : Gy(k) \leq g\},\tag{B.2}$$

with $G \in \mathbb{R}^{n_G \times n_y}$ and $g \in \mathbb{R}^{n_G}$.

The following algorithm calculates a positively-invariant constraint-admissible set of initial conditions, $x(0)$, which is referred to as a safe set.

1. Set the initial values of A_y and b_y to

$$A_y = GC \quad b_y = g.$$

2. Initialize the counter index $i = 1$.
3. Perform the following steps iteratively while $i < n_G$

- A) Select row i from A_y and b_y :

$$a = A_{y,i}, \quad b = b_{y,i}$$

- B) Check whether adding any constraints of the form $aCAx(0) \leq b$ to A_y and b_y would constrict the size of O_∞ by solving the following LP

$$c = \max_{x(0)} aCAx(0) - b \tag{B.3}$$

such that $A_yx(0) \leq b_y$.

- If $c > 0$, then add the constraints $aCAx(0) \leq b$ to A_y and b_y as follows:

$$A_y = \begin{bmatrix} A_y \\ aCA \end{bmatrix}, \quad b_y = \begin{bmatrix} b_y \\ b \end{bmatrix}. \tag{B.4}$$

- C) Increment $i : i = i + 1$.

4. Eliminate redundant constraints.

B.2 Calculation of disturbance safe sets

In this section, we discuss the calculation of set-bounded disturbance safe/recoverable sets described in Section 5.6.3. The algorithm described in this section is adapted from [79].

Consider the linear, discrete-time system with additive set-bounded disturbances

$$\begin{aligned} x(k+1) &= Ax(k) + Bw(k) \\ y(k) &= Cx(k) + Dw(k) \end{aligned} \tag{B.5}$$

where $x(k) \in \mathbb{R}^n$, $w(k) \in \mathbb{R}^m$, $y(k) \in \mathbb{R}^p$, A , B , C , and D are appropriate size, and $w \in W$ denotes that the disturbance, w , belongs to the compact set W . We constrain the output $y \in Y$ where

$$Y = \{y(k) : Gy(k) \leq g\}, \quad (\text{B.6})$$

with $G \in \mathbb{R}^{n_G \times n_y}$ and $g \in \mathbb{R}^{n_G}$.

We seek the set, O_∞ , defined as

$$O_\infty = \{x(0) \in \mathbb{R}^n : y(k) \in Y, \forall k \geq 0 \text{ and } \forall w \in W\}, \quad (\text{B.7})$$

where

$$y(k) = CA^k x(0) + \sum_{t=0}^{k-1} CA^{(k-t-1)} Bw(t) + Dw(k). \quad (\text{B.8})$$

This is the set of all initial conditions such that under all disturbances $w \in W$, the set is positively invariant.

In this section, we will utilize the Minkowski set subtraction defined for two sets $U \in \mathbb{R}^N$ and $V \in \mathbb{R}^N$ as

$$U \sim V = \{z \in \mathbb{R}^N : z + v \in U \forall v \in V\}. \quad (\text{B.9})$$

Let

$$Y_0 = Y \sim DW,$$

$$Y_k = Y \sim DW \sim \dots \sim CA^{k-1}BW,$$

for $k \geq 1$, and

$$O_k = \{x(0) \in \mathbb{R}^n : CA^t x(0) \in Y_t, \quad t = 0, \dots, k\},$$

which can be recursively rewritten as

$$Y_{k+1} = Y \sim CA^k BW, \quad Y_0 = Y \sim DW, \quad (\text{B.10})$$

and

$$\begin{aligned} O_{k+1} &= O_k \cap \{x \in \mathbb{R}^n : CA^{k+1}x \in Y_{k+1}\}, \\ O_0 &= \Gamma = \{x : Cx \in Y_0\}. \end{aligned} \quad (\text{B.11})$$

Thus, O_∞ is equivalent to

$$O_\infty = \bigcap_{k \geq 0} O_k. \quad (\text{B.12})$$

In practice, an intersection of infinite sets is difficult. However, [79] goes on to prove that O_∞ is finitely-determined at a time $k^* \geq 0$, thus

$$O_\infty = \bigcap_{k \geq 0}^{k^*} O_k. \quad (\text{B.13})$$

For a proof of this, see [79].

The following algorithm is used to calculate O_∞ .

1. Set $k = 0$ and $O_0 = \Gamma = \{x \in \mathbb{R}^n : Cx \in Y \sim DW\}$. If $O_0 = \emptyset$, set $O_\infty = \emptyset$, $k^* = 0$ and stop.
2. If not stopped, determine Y_{k+1} by (B.10). If $Y_{k+1} = \emptyset$, set $O_\infty = \emptyset$, $k^* = k + 1$, and stop.
3. If not stopped, determine O_{k+1} by (B.11). If $O_{k+1} = \emptyset$, set $O_\infty = \emptyset$, $k^* = k + 1$, and stop.
4. If not stopped and if $O_{k+1} = O_k$, set $O_\infty = O_k$, $k^* = k$, eliminate redundant constraints, and stop.
5. If not stopped, replace k by $k + 1$ and return to step 2.

B.3 Calculation of reference integral safe sets

This section describes the calculation of O_∞^r from Section 7.2.

Consider the reference tracking closed-loop system of the form,

$$\begin{aligned} \bar{x}(k+1) &= \bar{A}\bar{x}(k) + \bar{B}r(k), \\ y(k) &= \bar{C}\bar{x}(k) + \bar{D}r(k), \end{aligned} \quad (\text{B.14})$$

where $r(k) \in \mathbb{R}^q$ is the reference command, $\bar{B} \in \mathbb{R}^{(n+m) \times q}$, and $\bar{D} \in \mathbb{R}^{p \times q}$. The state $\bar{x}(k)$ is of the form (7.9) composed of resettable components, x_I , and non-resettable components, x , with

$$x_I(k+1) = x_I(k) + C_I x(k) - r(k), \quad (\text{B.15})$$

and let \bar{r} denote the nominal reference command.

The matrix \bar{A} is assumed to be Schur, and let $\bar{x}_e = Hr$ denote the equilibrium state corresponding to a constant command $r(k) \equiv r$.

Consider the set

$$\begin{aligned} O_\infty^r &= \{(\bar{x}(0), r) : (\bar{C}H + \bar{D})r \in (1 - \epsilon)Y, \\ r(k) &= r \forall k \geq 0 \Rightarrow \bar{y}(k) \in Y \forall k \geq 0\}, \end{aligned} \quad (\text{B.16})$$

where $\epsilon > 0$ is sufficiently small and Y is defined the same as (7.4). The set O_∞^r is the set of all initial states, $\bar{x}(0)$, and constant reference commands, r , that lead to responses satisfying the constraints.

A similar argument can be made for finite-determination of the reference integral safe sets as was made for the set-bounded disturbance safe/recoverable sets. Thus, in the following, if $k_{max} \geq k^*$ is selected to be sufficiently large, the calculation procedure described below will converge onto the true set.

The calculation of this set is accomplished using the following procedure.

1. Initialize $H_0 = G$ and $h_0 = g$, with G and g from (7.4). Initialize $k_{max} \geq 0$ and counter index $k = 1$.
2. While $k < k_{max}$ repeat the following step

A)

$$H_k = \begin{bmatrix} H_{k-1} \\ H(\bar{C}(I - A^k)(I - A)^{-1}\bar{B} + \bar{D}) \end{bmatrix}, \quad h_k = \begin{bmatrix} h_{k-1} \\ g \end{bmatrix}. \quad (\text{B.17})$$

3. Eliminate redundant constraints.
4. O_∞^r is then polytope described by $H_{k_{max}}x(0) \leq h_{k_{max}}$.

APPENDIX C

Scaled NASA Generic Transport Model

This appendix presents the information regarding the scaled NASA generic transport model (GTM), [107, 108]. The GTM is designed to be scaled representation of a typical transport aircraft such as the Boeing 757 or Airbus A320. In this dissertation, two versions of the model were used. The first version, used in Sections 5.1–5.5 and Chapter 7, is a high-fidelity Simulink model. It was linearized about several trim conditions, the necessary information was then saved into databases and used to develop the results in those chapters.

The second model, used in Section 5.6 and Chapter 6, uses a database of aerodynamic coefficients calculated using the VorView software package [117] for the scaled NASA GTM aircraft. These aerodynamic coefficients were then used to generate the linear models using equations (C.1)-(C.8). These equations are linearizations of the nonlinear aircraft dynamics and can be found in many books describing aircraft dynamics such as [118].

The longitudinal equations are:

$$\begin{aligned} \dot{\Delta u} = \frac{-p_d S}{m U_0} (2C_{D0} + C_{Du}) \Delta u + \frac{p_d S}{m} (C_{L0} + C_{D\alpha}) \Delta \alpha + \frac{p_d S c}{2m U_0} C_{Dq} \Delta q \\ -g \Delta \theta + \frac{p_d S}{m} C_{D\delta_e} \delta_e + \frac{T_{max}}{m} \delta_{th}, \end{aligned} \quad (C.1)$$

$$\begin{aligned} \dot{\Delta \alpha} = \frac{-p_d S}{m U_0^2} (2C_{L0} + C_{Lu}) \Delta u + \frac{-p_d S}{m U_0} (C_{L\alpha} + C_{D0}) \Delta \alpha \\ + (1 - \frac{p_d S c}{2m U_0^2} C_{Lq}) \Delta q - \frac{p_d S}{m U_0} C_{L\delta_e} \delta_e, \end{aligned} \quad (C.2)$$

$$\begin{aligned} \dot{\Delta q} = \frac{p_d S C}{I_{yy} U_0} (2C_{m0} + C_{mu}) \Delta u + \frac{p_d S c}{I_{yy}} C_{m\alpha} \Delta \alpha \\ + \frac{p_d S c^2}{2I_{yy} U_0} C_{mq} \Delta q - \frac{p_d S c}{I_{yy}} C_{L\delta_e} \delta_e, \end{aligned} \quad (C.3)$$

$$\dot{\Delta\theta} = \Delta q. \quad (\text{C.4})$$

The lateral equations are:

$$\begin{aligned} \dot{\Delta\beta} &= \frac{-p_d S}{mU_0} (C_{D0} + C_{E\beta}) \Delta\beta + \frac{-p_d S b}{2mU_0^2} C_{Ep} \Delta p \\ &+ \left(\frac{p_d S b}{2mU_0^2} C_{Er} - 1 \right) \Delta r - \frac{p_d S}{mU_0} C_{E\delta a} \delta a - \frac{p_d S}{mU_0} C_{E\delta r} \delta r, \end{aligned} \quad (\text{C.5})$$

$$\begin{aligned} \dot{\Delta p} - \frac{I_{xz}}{I_{xx}} \dot{\Delta r} &= \frac{p_d S b}{I_{xx}} C_{l\beta} \Delta\beta + \frac{p_d S b^2}{2I_{xx} U_0} C_{lp} \Delta p + \frac{p_d S b^2}{2I_{xx} U_0} C_{lr} \Delta r \\ &+ \frac{p_d S b}{I_{xx}} C_{l\delta a} \delta a + \frac{p_d S b}{I_{xx}} C_{l\delta r} \delta r, \end{aligned} \quad (\text{C.6})$$

$$\begin{aligned} \dot{\Delta r} - \frac{I_{xz}}{I_{zz}} \dot{\Delta p} &= \frac{p_d S b}{I_{zz}} C_{n\beta} \Delta\beta + \frac{p_d S b^2}{2I_{zz} U_0} C_{np} \Delta p + \frac{p_d S b^2}{2I_{zz} U_0} C_{nr} \Delta r \\ &+ \frac{p_d S b}{I_{zz}} C_{n\delta a} \delta a + \frac{p_d S b}{I_{zz}} C_{n\delta r} \delta r, \end{aligned} \quad (\text{C.7})$$

$$\dot{\Delta\phi} = \Delta p. \quad (\text{C.8})$$

BIBLIOGRAPHY

- [1] Bastani, P., Heywood, J. B., and Hope, C., “Us cafe standards,” 2012.
- [2] Coltman, D., Turner, J., Curtis, R., Blake, D., Holland, B., Pearson, R., Arden, A., and Nuglisch, H., “Project Sabre: A close-spaced direct injection 3-cylinder engine with synergistic technologies to achieve low CO₂ output,” Tech. rep., SAE Technical Paper, 2008.
- [3] Larimore, J., Hellstrom, E., Sterniak, J., Jiang, L., and Stefanopoulou, A. G., “Experiments and analysis of high cyclic variability at the operational limits of spark-assisted HCCI combustion,” *American Control Conference (ACC), 2012*, IEEE, 2012, pp. 2072–2077.
- [4] Jade, S., Hellstrom, E., Jiang, L., and Stefanopoulou, A. G., “Fuel governor augmented control of recompression HCCI combustion during large load transients,” *American Control Conference (ACC), 2012*, IEEE, 2012, pp. 2084–2089.
- [5] Sciarretta, A. and Guzzella, L., “Fuel-Optimal Control of Rendezvous Maneuvers for Passenger Cars (Treibstoffoptimale Annäherung von Straßenfahrzeugen),” *at-Automatisierungstechnik*, Vol. 53, No. 6/2005, 2005, pp. 244–250.
- [6] Sciarretta, A. and Guzzella, L., “Control of hybrid electric vehicles,” *Control systems, IEEE*, Vol. 27, No. 2, 2007, pp. 60–70.
- [7] Stoicescu, A., “On fuel-optimal velocity control of a motor vehicle,” *International Journal of Vehicle Design*, Vol. 16, No. 2/3, 1995.
- [8] Kolmanovsky, I. V. and Filev, D. P., “Stochastic optimal control of systems with soft constraints and opportunities for automotive applications,” *Control Applications, (CCA) & Intelligent Control, (ISIC), 2009 IEEE*, IEEE, 2009, pp. 1265–1270.
- [9] Kolmanovsky, I. V. and Filev, D., “Terrain and traffic optimized vehicle speed control,” *Advances in Automotive Control*, 2010, pp. 378–383.
- [10] EPA, “Regulations and Standards <http://www.epa.gov/fueleconomy/regulations.htm>,” July 2014.
- [11] Girard, A. R., de Sousa, J. B., Misener, J. A., and Hedrick, J. K., “A control architecture for integrated cooperative cruise control and collision warning systems,”

- Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, Vol. 2, IEEE, 2001, pp. 1491–1496.
- [12] Liang, C.-Y. and Peng, H., “Optimal adaptive cruise control with guaranteed string stability,” *Vehicle System Dynamics*, Vol. 32, No. 4-5, 1999, pp. 313–330.
- [13] Naus, G., Vugts, R., Ploeg, J., van de Molengraft, R., and Steinbuch, M., “Cooperative adaptive cruise control, design and experiments,” *American Control Conference (ACC), 2010*, IEEE, 2010, pp. 6145–6150.
- [14] Morbidi, F., Colaneri, P., and Stanger, T., “Decentralized optimal control of a car platoon with guaranteed string stability,” *Control Conference (ECC), 2013 European*, IEEE, 2013, pp. 3494–3499.
- [15] Bareket, Z., Fancher, P. S., Peng, H., Lee, K., and Assaf, C. A., “Methodology for assessing adaptive cruise control behavior,” *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 4, No. 3, 2003, pp. 123–131.
- [16] Girard, A. R., de Sousa, J. B., and Hedrick, J. K., “An overview of emerging results in networked multi-vehicle systems,” *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, Vol. 2, IEEE, 2001, pp. 1485–1490.
- [17] Gausemeier, S. et al., “Multi-objective optimization of a vehicle velocity profile by means of dynamic programming,” *Advances in Automotive Control*, 2010, pp. 366–371.
- [18] Hellström, E., Ivarsson, M., Åslund, J., and Nielsen, L., “Look-ahead control for heavy trucks to minimize trip time and fuel consumption,” *Control Engineering Practice*, Vol. 17, No. 2, 2009, pp. 245–254.
- [19] Hellström, E., Åslund, J., and Nielsen, L., “Design of an efficient algorithm for fuel-optimal look-ahead control,” *Control Engineering Practice*, Vol. 18, No. 11, 2010, pp. 1318–1327.
- [20] Ozatay, E., Onori, S., Wollaeger, J., Ozguner, U., Rizzoni, G., Filev, D., Michelini, J., and Di Cairano, S., “Cloud-Based Velocity Profile Optimization for Everyday Driving: A Dynamic-Programming-Based Solution,” 2013.
- [21] Kamal, M., Mukai, M., Murata, J., and Kawabe, T., “Development of ecological driving system using model predictive control,” *ICCAS-SICE, 2009*, IEEE, 2009, pp. 3549–3554.
- [22] Li, S., Li, K., Rajamani, R., and Wang, J., “Model predictive multi-objective vehicular adaptive cruise control,” *Control Systems Technology, IEEE Transactions on*, Vol. 19, No. 3, 2011, pp. 556–566.
- [23] Stanger, T. and del Re, L., “A model predictive cooperative adaptive cruise control approach,” *American Control Conference (ACC), 2013*, IEEE, 2013, pp. 1374–1379.

- [24] Lang, D., Stanger, T., and del Re, L., “Fuel efficient quasi optimal adaptive cruise control by control identification,” *Control Applications (CCA), 2013 IEEE International Conference on*, IEEE, 2013, pp. 229–234.
- [25] Lang, D., Stanger, T., and del Re, L., “Opportunities on fuel economy utilizing v2v based drive systems,” Tech. rep., SAE Technical Paper, 2013.
- [26] Lang, D., Stanger, T., Schmied, R., and del Re, L., “Predictive Cooperative Adaptive Cruise Control: Fuel Consumption Benefits and Implementability,” *Optimization and Optimal Control in Automotive Systems*, Springer, 2014, pp. 163–178.
- [27] Bichi, M., Ripaccioli, G., Di Cairano, S., Bernardini, D., Bemporad, A., and Kolmanovsky, I. V., “Stochastic model predictive control with driver behavior learning for improved powertrain control,” *Decision and Control (CDC), 2010 49th IEEE Conference on*, IEEE, 2010, pp. 6077–6082.
- [28] Gilbert, E. G., “Vehicle cruise: improved fuel economy by periodic control,” *Automatica*, Vol. 12, No. 2, 1976, pp. 159–166.
- [29] Lee, J., Nelson, D. J., and Lohse-Busch, H., “Vehicle inertia impact on fuel consumption of conventional and hybrid electric vehicles using acceleration and coast driving strategy,” Tech. rep., SAE Technical Paper, 2009.
- [30] Li, S. E., Peng, H., Li, K., and Wang, J., “Minimum fuel control strategy in automated car-following scenarios,” *Vehicular Technology, IEEE Transactions on*, Vol. 61, No. 3, 2012, pp. 998–1007.
- [31] Li, S. E. and Peng, H., “Strategies to minimize the fuel consumption of passenger cars during car-following scenarios,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, Vol. 226, No. 3, 2012, pp. 419–429.
- [32] Wikipedia, “Fuel economy-maximizing behaviors http://en.wikipedia.org/wiki/Fuel_economy-maximizing_behaviors,”.
- [33] Lin, C.-C., Peng, H., and Grizzle, J., “A stochastic control strategy for hybrid electric vehicles,” *American Control Conference, 2004. Proceedings of the 2004*, Vol. 5, IEEE, 2004, pp. 4710–4715.
- [34] Tate, E. D., Grizzle, J. W., and Peng, H., “Shortest path stochastic control for hybrid electric vehicles,” *International Journal of Robust and Nonlinear Control*, Vol. 18, No. 14, 2008, pp. 1409–1429.
- [35] Ripaccioli, G., Bernardini, D., Di Cairano, S., Bemporad, A., and Kolmanovsky, I., “A stochastic model predictive control approach for series hybrid electric vehicle power management,” *American Control Conference (ACC), 2010*, IEEE, 2010, pp. 5844–5849.

- [36] Opila, D. F., Wang, X., McGee, R., Gillespie, R. B., Cook, J. A., and Grizzle, J. W., “An energy management controller to optimally trade off fuel economy and drivability for hybrid vehicles,” *Control Systems Technology, IEEE Transactions on*, Vol. 20, No. 6, 2012, pp. 1490–1505.
- [37] Kolmanovsky, I., Siverguina, I., and Lygoe, B., “Optimization of powertrain operating policy for feasibility assessment and calibration: Stochastic dynamic programming approach,” *American Control Conference, 2002. Proceedings of the 2002*, Vol. 2, IEEE, 2002, pp. 1425–1430.
- [38] Filev, D. P. and Kolmanovsky, I., “Markov chain modeling approaches for on board applications,” *American Control Conference (ACC), 2010*, IEEE, 2010, pp. 4139–4145.
- [39] Filev, D. P. and Kolmanovsky, I., “A generalized Markov Chain modeling approach for on board applications,” *Neural Networks (IJCNN), The 2010 International Joint Conference on*, IEEE, 2010, pp. 1–8.
- [40] Altman, E., *Constrained Markov decision processes*, Vol. 7, CRC Press, 1999.
- [41] Dynkin, E. B., Iushkevich, A. A., and Yushkevich, A. A., *Markov processes; theorems and problems*, Plenum Press, 1969.
- [42] Bertsekas, D. P., Bertsekas, D. P., Bertsekas, D. P., and Bertsekas, D. P., *Dynamic programming and optimal control*, Vol. 1, Athena Scientific Belmont, MA, 1995.
- [43] McDonough, K., Kolmanovsky, I., Filev, D., Yanakiev, D., Szwabowski, S., and Michelini, J., “Stochastic dynamic programming control policies for fuel efficient in-traffic driving,” *American Control Conference (ACC), 2012*, IEEE, 2012, pp. 3986–3991.
- [44] McDonough, K., Kolmanovsky, I., Filev, D., Yanakiev, D., Szwabowski, S., and Michelini, J., “Stochastic dynamic programming control policies for fuel efficient vehicle following,” *American Control Conference (ACC), 2013*, IEEE, 2013, pp. 1350–1355.
- [45] McDonough, K., Kolmanovsky, I., Filev, D., Szwabowski, S., Yanakiev, D., and Michelini, J., “Stochastic Fuel Efficient Optimal Control of Vehicle Speed,” *Optimization and Optimal Control in Automotive Systems*, Springer, 2014, pp. 147–162.
- [46] McDonough, K., D’Amator, A., Mullen, J., Petersen, C., Kolmanovsky, I., and Filev, D., “Transition threshold optimization for a rule based automotive cruise control,” *Conference on Decision and Control (CDC), 2014*, IEEE, 2014.
- [47] Hoekstra, A., Filev, D., Szwabowski, S., McDonough, K., and Kolmanovsky, I., “Evolving Markov chain models of driving conditions using onboard learning,” *Cybernetics (CYBCONF), 2013 IEEE International Conference on*, IEEE, 2013, pp. 1–6.

- [48] McDonough, K., Kolmanovsky, I., Filev, D., Yanakiev, D., Szwabowski, S., Micheli, J., and Abou-Nasr, M., “Modeling of vehicle driving conditions using transition probability models,” *Control Applications (CCA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 544–549.
- [49] Belcastro, C. M., Groff, L., Newman, R. L., Foster, J. V., Crider, D. A., Klyde, D. H., and Huston, A. M., “Preliminary Analysis of Aircraft Loss of Control Accidents: Worst Case Precursor Combinations and Temporal Sequencing,” *AIAA SciTech Conference*, 2014, pp. 13–17.
- [50] Hersman, D., Hart, C., and Sumwalt, R., “Loss of Thrust in Both Engines After Encountering a Flock of Birds and Subsequent Ditching on the Hudson River,” Tech. rep., Accident Report NTSB/AAR-10/03, National Transportation Safety Board, Washington DC, 2010.
- [51] Belcastro, C. M. and Foster, J. V., “Aircraft loss-of-control accident analysis,” *AIAA Guidance, Navigation and Control Conference, Toronto, Canada*, 2010.
- [52] Chongvisal, J., Talleur, D., Kirlik, A., Hovakimyan, N., and Belcastro, C., “Loss-of-control prediction and prevention for NASAs Transport Class Model,” *AIAA Guidance, Navigation and Control Conference, National Harbor, MD*, 2014.
- [53] Kwatny, H. G., Dongmo, J.-E. T., Chang, B.-C., Bajpai, G., Yasar, M., and Belcastro, C., “Nonlinear Analysis of Aircraft Loss of Control,” *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 1, 2012, pp. 149–162.
- [54] Wilborn, J. E. and Foster, J. V., “Defining commercial transport loss-of-control: A quantitative approach,” *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 2004.
- [55] Boeing, N., “Statistical Summary of Commercial Jet Airplane Accidents—Worldwide Operations 1959–2003,” *Boeing Commercial Airplanes, Seattle, Washington*, 2004.
- [56] Kwatny, H. G., Dongmo, J.-E. T., Allen, R. C., Chang, B.-C., and Bajpai, G., “Loss-of-control: Perspectives on flight dynamics and control of impaired aircraft,” *AIAA Conference on Guidance, Navigation, and Control, Toronto*, 2010.
- [57] Kwatny, G. and Allen, R. C., “Safe Set Maneuverability of Impaired Aircraft,” *Proceedings AIAA Guidance Navigation and Control Conference and Exhibit, Chicago*, 2012, pp. 12–16.
- [58] Gilbert, E. G. and Tan, K. T., “Linear systems with state and control constraints: The theory and application of maximal output admissible sets,” *Automatic Control, IEEE Transactions on*, Vol. 36, No. 9, 1991, pp. 1008–1020.
- [59] Gilbert, E. G. and Ong, C., “Linear systems with hard constraints and variable set points: Their robustly invariant sets,” *Dept. of Mech. Eng., Nat. Univ. Singapore, Tech. Rep. C08-002*, 2008.

- [60] Weiss, A., Baldwin, M., Erwin, R. S., and Kolmanovsky, I., “Spacecraft constrained maneuver planning using positively invariant constraint admissible sets,” *Proceedings of Itzhack Y. Bar-Itzhack Memorial Symposium on Estimation, Navigation and Spacecraft Control*, 2012, pp. 14–17.
- [61] Kolmanovsky, I., Garone, E., and Di Cairano, S., “Reference and command governors: A tutorial on their theory and automotive applications,” *American Control Conference (ACC), 2014*, IEEE, 2014, pp. 226–241.
- [62] Poloni, T., Kalabic, U., McDonough, K., and Kolmanovsky, I., “Disturbance canceling control based on simple input observers with constraint enforcement for aerospace applications,” *Control Applications (CCA), 2014 IEEE Conference on*, IEEE, 2014, pp. 158–165.
- [63] Tomlin, C., Lygeros, J., and Sastry, S., “Aerodynamic envelope protection using hybrid control,” *American Control Conference, 1998. Proceedings of the 1998*, Vol. 3, IEEE, 1998, pp. 1793–1796.
- [64] Lygeros, J., Tomlin, C., and Sastry, S., “Controllers for reachability specifications for hybrid systems,” *Automatica*, Vol. 35, No. 3, 1999, pp. 349–370.
- [65] Lygeros, J., “On reachability and minimum cost optimal control,” *Automatica*, Vol. 40, No. 6, 2004, pp. 917–927.
- [66] Bayen, A. M., Mitchell, I. M., Osihi, M. K., and Tomlin, C. J., “Aircraft autolander safety analysis through optimal control-based reach set computation,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, 2007, pp. 68–77.
- [67] Pandita, R., Chakraborty, A., Seiler, P., and Balas, G., “Reachability and region of attraction analysis applied to GTM dynamic flight envelope assessment,” *AIAA Guidance, Navigation, and Control, Chicago, Illinois*, 2009, pp. 1–21.
- [68] Oort, E., Chu, Q., and Mulder, J., “Maneuver Envelope Determination through Reachability Analysis,” *Advances in Aerospace Guidance, Navigation and Control*, 2011, pp. 91–102.
- [69] Gillula, J. H., Hoffmann, G. M., Huang, H., Vitus, M. P., and Tomlin, C., “Applications of hybrid reachability analysis to robotic aerial vehicles,” *The International Journal of Robotics Research*, 2011, pp. 0278364910387173.
- [70] Allen, R. C. and Kwatny, H. G., “Maneuverability and envelope protection in the prevention of aircraft loss of control,” *Control Conference (ASCC), 2011 8th Asian*, IEEE, 2011, pp. 381–386.
- [71] Lombaerts, T., Schuet, S., Wheeler, K., Acosta, D., and Kaneshige, J., “Robust maneuvering envelope estimation based on reachability analysis in an optimal control formulation,” *Control and Fault-Tolerant Systems (SysTol), 2013 Conference on*, IEEE, 2013, pp. 318–323.

- [72] Lombaerts, T., Chu, Q., Mulder, J. A., and Joosten, D., “Flight control reconfiguration based on a modular approach,” *Fault Detection, Supervision and Safety of Technical Processes*, 2009, pp. 259–264.
- [73] Lombaerts, T., Chu, Q., Mulder, J., and Joosten, D., “Modular flight control reconfiguration design and simulation,” *Control Engineering Practice*, Vol. 19, No. 6, 2011, pp. 540–554.
- [74] Schuet, S., Acosta, D., Wheeler, K., Kaneshige, J., and Lombaerts, T., “An Adaptive Nonlinear Aircraft Maneuvering Envelope Estimation Approach for Online Applications,” *AIAA Guidance, Navigation and Control Conference, accepted for presentation*, 2014.
- [75] McDonough, K., Kolmanovsky, I., and Atkins, E., “Recoverable sets of initial conditions and their use for aircraft flight planning after a loss of control event,” 2014.
- [76] Yu, M.-J., McDonough, K., Bernstein, D. S., and Kolmanovsky, I., “Retrospective cost model refinement for aircraft fault signature detection,” *American Control Conference (ACC), 2014*, IEEE, 2014, pp. 2486–2491.
- [77] McDonough, K. and Kolmanovsky, I., “Integrator resetting for enforcing constraints in aircraft flight control systems,” 2015.
- [78] Kolmanovsky, I. and Gilbert, E. G., “Maximal output admissible sets for discrete-time systems with disturbance inputs,” *American Control Conference, Proceedings of the 1995*, Vol. 3, IEEE, 1995.
- [79] Kolmanovsky, I. and Gilbert, E. G., “Theory and computation of disturbance invariant sets for discrete-time linear systems,” *Mathematical Problems in Engineering*, Vol. 4, No. 4, 1998, pp. 317–367.
- [80] Blanchini, F. and Miani, S., *Set-theoretic methods in control*, Springer, 2007.
- [81] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O., “Constrained model predictive control: Stability and optimality,” *Automatica*, Vol. 36, No. 6, 2000, pp. 789–814.
- [82] Mayne, D. Q., Seron, M. M., and Raković, S., “Robust model predictive control of constrained linear systems with bounded disturbances,” *Automatica*, Vol. 41, No. 2, 2005, pp. 219–224.
- [83] Pluymers, B., Rossiter, J., Suykens, J., and De Moor, B., “The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty,” *American Control Conference, 2005. Proceedings of the 2005*, IEEE, 2005, pp. 804–809.
- [84] Athanasopoulos, N. and Bitsoris, G., “Invariant set computation for constrained uncertain discrete-time linear systems,” *Decision and Control (CDC), 2010 49th IEEE Conference on*, IEEE, 2010, pp. 5227–5232.

- [85] Weiss, A., Petersen, C., Baldwin, M., Erwin, R. S., and Kolmanovsky, I., “Safe Positively Invariant Sets for Spacecraft Obstacle Avoidance,” *Journal of Guidance, Control, and Dynamics*, 2014, pp. 1–13.
- [86] Chiang, L. H., Braatz, R. D., and Russell, E. L., *Fault detection and diagnosis in industrial systems*, Springer, 2001.
- [87] Urnes, J., Reichenbach Sr, E., and Smith, T. A., “Dynamic flight envelope assessment and prediction,” *AIAA Guidance Navigation and Control Conference*, 2008, pp. 18–21.
- [88] Tang, L., Roemer, M., Ge, J., Crassidis, A., Prasad, J., and Belcastro, C., “Methodologies for adaptive flight envelope estimation and protection,” *Proceedings of the Guidance, Navigation, and Control Conference, AIAA, Chicago, Ill*, 2009.
- [89] Lombaerts, T., Oort, E. V., Chu, Q., Mulder, J., and Joosten, D., “Online Aerodynamic Model Structure Selection and Parameter Estimation for Fault Tolerant Control,” *Journal of guidance, control, and dynamics*, Vol. 33, No. 3, 2010, pp. 707–723.
- [90] Stepanyan, V., Barlow, J., and Bjil, H., “Adaptive Estimation Based Loss of Control Detection and Mitigation,” *Proc. of AIAA Guidance, Navigation, and Control Conference, Portland, Oregon*, 2011.
- [91] Desoer, C. and Lin, C.-A., “Tracking and disturbance rejection of MIMO nonlinear systems with PI controller,” *Automatic Control, IEEE Transactions on*, Vol. 30, No. 9, 1985, pp. 861–867.
- [92] Bupp, R. T., Bernstein, D. S., Chellaboina, V. S., and Haddad, W. M., “Resetting virtual absorbers for vibration control,” *Journal of Vibration and Control*, Vol. 6, No. 1, 2000, pp. 61–83.
- [93] Mcclamroch, N. H. and Kolmanovsky, I., “Performance benefits of hybrid control design for linear and nonlinear systems,” *Proceedings of the IEEE*, Vol. 88, No. 7, 2000, pp. 1083–1096.
- [94] Haddad, W. M., Chellaboina, V., and Nersesov, S. G., *Impulsive and hybrid dynamical systems: stability, dissipativity, and control*, Princeton University Press, 2006.
- [95] Di Cairano, S. and Kolmanovsky, I. V., “Constrained actuator coordination by virtual state governing,” *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, IEEE, 2011, pp. 5491–5496.
- [96] McDonough, K. and Kolmanovsky, I., “Controller state and reference governors for discrete-time linear systems with pointwise-in-time state and control constraints,” *American Control Conference (ACC), 2015*, IEEE, 2015, p. accepted.
- [97] Demuth, H. and Beale, M., “Neural network toolbox for use with MATLAB,” 1993.
- [98] MSC, “Carsim Users Manual,” *Ann Arbor, MI*, Vol. 48013, 2002.

- [99] Lin, J., “Divergence measures based on the Shannon entropy,” *Information Theory, IEEE Transactions on*, Vol. 37, No. 1, 1991, pp. 145–151.
- [100] Vidyasagar, M., “Bounds on the Kullback-Leibler divergence rate between hidden Markov models,” *Decision and Control, 2007 46th IEEE Conference on*, IEEE, 2007, pp. 6160–6165.
- [101] Hanson, F. B., *Applied stochastic processes and control for jump-diffusions: modeling, analysis, and computation*, Vol. 13, Siam, 2007.
- [102] Gilbert, E. G. and Kolmanovsky, I. V., “Fast reference governors for systems with state and control constraints and disturbance inputs,” 1999.
- [103] Ong, C. J., Sui, D., and Gilbert, E. G., “Enlarging the terminal region of nonlinear model predictive control using the support vector machine method,” *Automatica*, Vol. 42, No. 6, 2006, pp. 1011–1016.
- [104] Aubin, J.-P., *Viability theory*, Springer Science & Business Media, 2009.
- [105] Kalabic, U., Kolmanovsky, I., Buckland, J., and Gilbert, E., “Reference and extended command governors for control of turbocharged gasoline engines based on linear models,” *Control Applications (CCA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 319–325.
- [106] Rossiter, J. A., *Model-based predictive control: a practical approach*, CRC press, 2013.
- [107] Cunningham, K., Foster, J., Morelli, E., and Murch, A., “Practical application of a subscale transport aircraft for flight research in control upset and failure conditions,” 2008.
- [108] Nguyen, N. T., Krishnakumar, K. S., Kaneshige, J. T., and Nespeca, P. P., “Flight dynamics and hybrid adaptive control of damaged aircraft,” *Journal of guidance, control, and dynamics*, Vol. 31, No. 3, 2008, pp. 751–764.
- [109] Bragg, M. B., Basar, T., Perkins, W. R., Selig, M. S., Voulgaris, P. G., Melody, J. W., and Sarter, N. B., “Smart icing systems for aircraft icing safety,” *AIAA Paper*, Vol. 813, 2002.
- [110] Reehorst, A. L., Addy Jr, H. E., and Colantonio, R. O., “Examination of Icing Induced Loss of Control and Its Mitigations,” *AIAA Paper*, Vol. 8140, 2010.
- [111] Tøndel, P., Johansen, T. A., and Bemporad, A., “An algorithm for multi-parametric quadratic programming and explicit MPC solutions,” *Automatica*, Vol. 39, No. 3, 2003, pp. 489–497.
- [112] Khalil, H. K. and Grizzle, J., *Nonlinear systems*, Vol. 3, Prentice hall Upper Saddle River, 2002.

- [113] May, R. D., Csank, J., Lavelle, T. M., Litt, J. S., and Guo, T.-H., *A high-fidelity simulation of a generic commercial aircraft engine and controller*, National Aeronautics and Space Administration, Glenn Research Center, 2010.
- [114] Powell, W. B., *Approximate Dynamic Programming: Solving the curses of dimensionality*, Vol. 703, John Wiley & Sons, 2007.
- [115] Yu, S., Maier, C., Chen, H., and Allgöwer, F., “Tube MPC scheme based on robust control invariant set with application to Lipschitz nonlinear systems,” *Systems & Control Letters*, Vol. 62, No. 2, 2013, pp. 194–200.
- [116] Balachandran, S., McDonough, K., di Donato, P., Atkins, E., and Kolmanovsky, I., “An Envelope Aware Flight Management System Applied to Icing Related Loss of Control Accidents,” *AIAA Journal of Guidance, Control, and Dynamics*, 2015, in preperation.
- [117] Nguyen, N., Trinh, K., Frost, S., and Reynolds, K., “Coupled Aeroelastic Vortex Lattice Modeling of Flexible Aircraft,” .
- [118] Stevens, B. L. and Lewis, F. L., *Aircraft control and simulation*, John Wiley & Sons, 2003.