

ROBUSTNESS AND OPTIMIZATION
IN
ANTI-WINDUP CONTROL

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER TOWARDS THE
DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

BY

RAZAK OLUSEGUN ALLI-OKE

2014

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

Contents

List of Figures	7
List of Tables	9
Nomenclature	10
List of Theorems	11
List of Algorithms	13
Abstract	14
Declaration	15
Copyright Statement	16
Acknowledgements	18
1 Introduction	19
1.1 Research Objectives	21
1.2 Aims of Thesis	21
1.3 Overview of Thesis	23
1.3.1 Robustness Preservation in Anti-windup Control	23
1.3.2 Accelerated gradient methods	23
1.4 Contributions of Thesis	24
1.5 Publications	25

I	Robustness Preservation in Anti-windup Control	26
2	Stability Theory	27
2.1	Vector Spaces & Norms	28
2.1.1	Banach Spaces	29
2.1.2	Hilbert Spaces	30
2.1.3	Operator Spaces	30
2.1.4	Extended Spaces	32
2.2	Lyapunov Stability Theory	32
2.3	Input-Output (IO) Stability Theory	35
2.3.1	Causal Systems	35
2.3.2	\mathcal{L} -Stability	36
2.3.3	Passivity Theory	37
2.3.3.1	Positive Real Systems	37
2.3.3.2	Passivity Theorems	38
2.3.4	Integral Quadratic Constraints (IQC) Theory	39
2.4	Summary	41
3	Introductory Background on Anti-Windup Control	42
3.1	Anti-Windup Control	43
3.1.1	Anti-Windup Compensators	43
3.1.2	Robust-Preserving Anti-windup	46
3.2	Summary	47
4	A Robust Kalman Conjecture	48
4.1	Introduction	48
4.2	Notation	51
4.3	Robust Interval	52
4.3.1	Graphical Interpretation	53
4.4	Robust Absolute Stability	54

4.5	Robust Kalman Conjecture	56
4.5.1	ADDITIVE UNCERTAINTY	57
4.5.2	INPUT-MULTIPLICATIVE UNCERTAINTY	61
4.5.3	FEEDBACK UNCERTAINTY	65
4.6	Conclusion	69
Summary of Part I		71
II Optimization in Anti-windup Control		72
5	Introductory Background on Optimization	73
5.1	What is Convex Optimization?	75
5.2	Unconstrained Optimization	75
5.2.1	Descent Methods	76
5.3	Constrained Optimization	79
5.3.1	Active-Set Method	81
5.3.2	Interior-Point Method	82
5.3.2.1	Infeasible-Interior-Point Method	85
5.3.3	Projected Gradient Method	86
5.3.4	Projected Fast Gradient Method	87
5.4	Summary	88
6	Algorithmic Analysis	89
6.1	The Big Oh (\mathcal{O}) Notation	90
6.2	Algorithmic Performance	90
6.2.1	Convergence Measures	91
6.2.2	Complexity Measures	92
6.2.3	Performance Profiles	94
6.3	Examples of Algorithm Analytics	95
6.4	Summary	98

7	A Secant-Based Nesterov Gradient Method	99
7.1	Introduction	99
7.2	Notation	101
7.3	Nesterov Gradient Method	102
7.3.1	NESTEROV'S CHOICE OF RECURSIVE RULE	104
7.3.2	NESTEROV'S CHOICE OF ESTIMATE-SEQUENCE	105
7.3.3	NESTEROV'S CHOICE OF β_k AND SEARCH POINT y_k	106
7.4	Quasi-Newton Method	109
7.5	A Secant-Based Nesterov Gradient Method	110
7.5.1	RECURSIVE RULE REVISITED	111
7.5.2	CONSTRUCTION OF Secant-Based-NGM	112
7.5.3	GLOBAL CONVERGENCE OF PROPOSED SCHEME	117
7.6	Simulation Examples	118
7.6.1	Simulation Setup	118
7.6.2	Test Functions and Solvers	118
7.6.3	Numerical Results	119
7.7	Conclusion	124
8	A Scaled Barzilai-Borwein Step-Size	125
8.1	Introduction	125
8.2	Notation	128
8.3	The Barzilai-Borwein step-size	128
8.3.1	GLOBALIZED BARZILAI-BOWEIN GRADIENT METHOD	131
8.4	A scaled-BB step-size: Scaled-BBGM	132
8.5	Simulation Examples I	136
8.5.1	Simulation Setup	136
8.5.2	Test Functions and Solvers	136
8.5.3	Numerical Results 1	137
8.6	Nonmonotone Lyapunov Functions	138

8.7	A Hybrid BB Gradient Method: Hybrid-BBGM	140
8.7.1	GLOBAL CONVERGENCE	142
8.8	Simulation Examples II	144
8.8.1	Numerical Results 2	144
8.9	Conclusion	146
9	Conclusion and Recommendations	147
9.1	Conclusion	147
9.2	Recommendations	148
9.2.1	Research Direction 1	148
9.2.2	Research Direction 2	148
9.2.3	Research Direction 3	149
	Appendix A	150
A.1	Robust Circle Criterion for Input-Multiplicative Uncertainty	150
	Bibliography	152

List of Figures

1.1	Feedback Control of a Simple Discrete-Integrator.	20
1.2	Effect of Integrator Wind-up.	20
2.1	Negative Feedback System.	38
2.2	Positive Feedback System.	39
3.1	Generic Anti-Windup Framework.	44
3.2	Classical Feedback Structure with Anti-Windup.	44
3.3	The Lur'e Structure.	46
4.1	The Lur'e Structure.	48
4.2	The Absolute Stability Conjecture.	49
4.3	The Robust Lur'e problem.	50
4.4	The Uncertain System.	52
4.5	Robust Nyquist Plot.	54
4.6	Augmented Lur'e problem.	54
4.7	Robust Kalman Conjecture (RKC).	57
4.8	Robust Lur'e for Additive Uncertainty.	57
4.9	Robust Nyquist Plot.	60
4.10	Robust Lur'e for Input-Multiplicative Uncertainty.	61
4.11	(a) Robust Nyquist and (b) Robust Popov Plots.	64
4.12	Robust Lur'e for Feedback Uncertainty.	65
4.13	Robust Nyquist Plot.	69

7.1	Nesterov's optimal concept.	103
7.2	Updating the local function $\Phi_k(x)$ for some y_k	105
7.3	Ensuring local condition (7.5) at the next iterate, $f(x_{k+1}) \leq \Phi_{k+1}^*(x)$	107
7.4	Quasi-Newton method, showing secant line P_1 - P_2	109
7.5	Effect of problem size (N) and condition number (L/μ) on the run-time.	120
7.6	Effect of problem size (N) and condition number (L/μ) on the run-time.	121
7.7	The performance profiles of the set of solvers \mathcal{S} for Ex. 1 and Ex. 2.	121
7.8	Effect of problem size (N) and condition number (L/μ) on the run-time.	122
7.9	Effect of problem size (N) and condition number (L/μ) on the run-time.	123
7.10	Effect of problem size (N) and condition number (L/μ) on the run-time.	123
8.1	Subclasses of continuous differentiable function f	125
8.2	The Gradient Method.	132
8.3	Graph of $\Delta V(x_k)$ against α_{k+1}	134
8.4	Effect of problem size (N) on the run-time.	138
8.5	Plot of Lyapunov Functions.	140
8.6	Effect of problem size (N) and condition number (L/μ) on the run-time.	145
8.7	Effect of problem size (N) and condition number (L/μ) on the run-time.	146

List of Tables

4.1	Conjectures	50
4.2	RKC for Additive Uncertainty	60
4.3	RKC for Input-Multiplicative Uncertainty	65
4.4	RKC for Feedback Uncertainty	69
4.5	Robust Kalman Conjecture For First-Order Plants	70

Nomenclature

Acronymns

IMC	Internal Model Control
IQC	Integral Quadratic Constraints
LHS	Left Half Side
LTI	Linear Time Invariant
MIMO	Multiple Input Multiple Output
QP	Quadratic Programme
RHP	Right Half Plane
SISO	Single Input Single Output
SPR	Strictly Positive Real

Binary Symbols

:	such that
\forall	for all
\Leftrightarrow	is equivalent to
\Rightarrow	implies
\rightarrow	tends to
\triangleq	is defined by

Fields of Numbers

\mathbb{R}^m	Real space of dimension m
\mathbb{C}^m	Complex space of dimension m
\mathcal{L}	Lebesgue space
\mathbb{F}	Real or Complex space

List of Theorems

2.1	Lemma ([40])	31
2.1	Theorem (Discrete Lyapunov Stability Theorem [43,44])	33
2.2	Theorem (Continuous Lyapunov Stability Theorem [46])	34
2.3	Theorem (Passivity Theorem [50])	38
2.4	Theorem (IQC Theorem [46])	40
2.2	Lemma (Schur Complement Lemma [52])	41
4.1	Theorem	52
4.2	Theorem (Nyquist Criterion [39])	53
4.3	Theorem (IQC Theorem [46])	55
4.1	Lemma	58
4.1	Result	59
4.2	Lemma	62
4.2	Result	63
4.3	Lemma	66
4.3	Result	67
7.1	Lemma ([23])	103
7.2	Lemma ([23])	103
7.3	Lemma ([23])	105
7.1	Theorem ([23])	108
7.4	Lemma	112
7.5	Lemma	112

7.6	Lemma	112
8.1	Theorem	133
8.2	Theorem	138
8.3	Theorem	139
8.1	Lemma	142
8.4	Theorem	143

List of Algorithms

5.1	Algorithm ([86])	76
5.2	Algorithm ([86])	78
5.3	Algorithm ([86])	81
5.4	Algorithm ([86])	85
5.5	Algorithm ([86])	87
5.6	Algorithm ([86])	88
7.1a	Algorithm ([23])	107
7.1b	Algorithm ([23])	108
7.2a	Algorithm (Basic Secant-Based-NGM)	115
7.2b	Algorithm (Simplified Secant-Based-NGM)	116
8.1	Algorithm (Scaled-BBGM)	135
8.2	Algorithm (Hybrid-BBGM)	141

Abstract

This thesis is broadly concerned with online-optimizing anti-windup control. These are control structures that implement some online-optimization routines to compensate for the windup effects in constrained control systems.

The first part of this thesis examines a general framework for analyzing robust preservation in anti-windup control systems. This framework - the robust Kalman conjecture - is defined for the robust Lur'e problem. This part of the thesis verifies this conjecture for first-order plants perturbed by various norm-bounded unstructured uncertainties. Integral quadratic constraint theory is exploited to classify the appropriate stability multipliers required for verification in these cases.

The remaining part of the thesis focusses on accelerated gradient methods. In particular, tight complexity-certificates can be obtained for the Nesterov gradient method, which makes it attractive for implementation of online-optimizing anti-windup control. This part of the thesis presents a proposed algorithm that extends the classical Nesterov gradient method by using available secant information. Numerical results demonstrating the efficiency of the proposed algorithm are analysed with the aid of performance profiles. As the objective function becomes more ill-conditioned, the proposed algorithm becomes significantly more efficient than the classical Nesterov gradient method. The improved performance bodes well for online-optimization anti-windup control since ill-conditioning is common place in constrained control systems.

In addition, this thesis explores another subcategory of accelerated gradient methods known as Barzilai-Borwein gradient methods. Here, two algorithms that modify the Barzilai-Borwein gradient method are proposed. Global convergence of the proposed algorithms for all convex functions are established by using discrete Lyapunov theorems.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

Copyright Statement

1. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.
2. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.
3. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.
4. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property

and/or Reproductions described in it may take place is available in the University IP Policy (see <http://www.campus.manchester.ac.uk/medialibrary/policies/intellectual-property.pdf>), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see <http://www.manchester.ac.uk/library/aboutus/regulations>) and in The University's policy on presentation of Theses.

Acknowledgments

The objectives of this work would not have been accomplished without the Grace of Almighty Allah (SWT), the most beneficent and the most merciful who spared my life up to this present moment.

I am indeed grateful to my parents, Engr. Wale Alli-Oke & Mrs Adetutu Alli-Oke for their financial, moral and spiritual support during my entire period at the university. My sincere gratitude to my supervisor/sponsor, Dr William Heath, for his unrelenting support and exceptional guidance in the course of this work.

I am also using this opportunity to express my gratitude to my colleagues at the Control Systems Center, University of Manchester, for their encouragement and support throughout my PhD studies here at the University of Manchester.

Chapter 1

Introduction

Constraints are present in all control systems. Most often constraints are identified as actuator magnitude and rate saturation, or output and state variable constraints. Practical examples of these constraints include motor valve openings, engine nozzle openings, torque/speed limitations of motors and flight control surface deflections. The destabilizing effects of such constraints include performance deterioration, oscillatory behaviour and even instability. The aforementioned destabilizing effects have been cited as contributing factors in several mishaps involving high performance aircraft [1] where pilot-induced oscillations were partly caused by rate saturation of control surfaces inducing time-delay effects in the control loop.

The control "windup" effect occurs when the controller is not capable of instantly responding to the changes in the control error. The control "windup" phenomena were initially encountered by practical control engineers when designing PID controllers but will also be present in any dynamical controller with relatively slow or unstable states as pointed out in Doyle et al [2]. The control "windup" problem has also been attributed to the Chernobyl disaster [54] where limits in the rate of change of the actuator pushing the control rods into the core, aggravated an already dangerous situation. A standard example of control windup is integrator windup. In this case, the mismatch between the controller output and the plant input causes the integrator mode to continue integrating the non-zero error and hence leads to accumulation of a significant error that would

then require that the error has an opposite sign for a long period before the controller action returns to normal. This behaviour results in large transients and in some cases the closed-loop system may also become unstable or exhibit an oscillating behaviour. A simulation example is considered to illustrate the effect of integrator windup in degrading system performance. Consider a unity-feedback interconnection of a discrete-integrator process as shown in Figure 1.1,

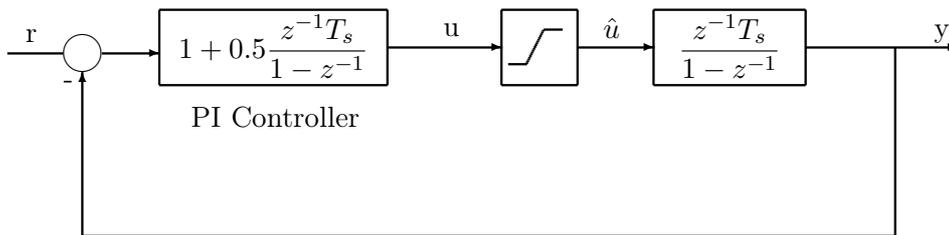


Figure 1.1: Feedback Control of a Simple Discrete-Integrator.

where $T_s = 0.1$ is the sampling time. The figure below, Figure 1.2, shows the unit-step response when the PI-controller output is unconstrained, constrained to $\pm 0.25V$ and

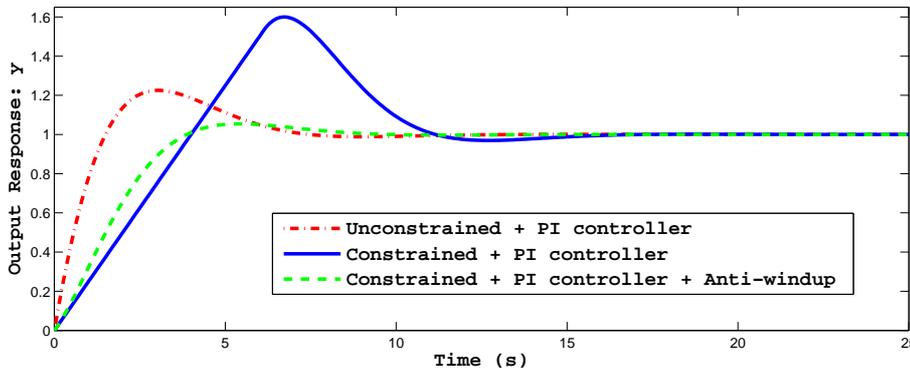


Figure 1.2: Effect of Integrator Wind-up.

The saturation constraints cause a windup effect which leads to the observed overshoot. The velocity-form PI (anti-windup) controller reduces the effect of this integral windup.

a velocity-form PI (anti-windup) controller is used. It can be observed from Figure 1.2 that the windup effect induces large transients in the output response and that this windup effect is significantly reduced by using a velocity-form PI (anti-windup) controller.

1.1 Research Objectives

Early ad-hoc efforts to address the windup effect include velocity-form [3], back-tracking [4] and conditional integration [4] of the controller. Kothare et al [5] have put forward a unified coprime factor framework which includes all previous ad-hoc anti-windup schemes as particular special cases. A few authors have attempted to analyze the stability properties of the resulting closed-loop behaviour of these anti-windup schemes using stability tests (either via Lyapunov functions or Integral Quadratic Constraint theory) such as the circle criterion, the Popov criterion and the use of Zames-Falb multipliers. Kothare and Morari [6] gave an overview of the theory applied to anti-windup of MIMO systems.

The subject of this research is concerned about the growing trend of an anti-windup paradigm known as the online-optimizing anti-windup scheme. These anti-windup schemes are control structures that implement some online-optimization routines to compensate for the windup effects. An important class of these online-optimizing anti-windup schemes is the well-known model predictive control (MPC), and more importantly is the recent one introduced by [11]. This scheme [11] represent a class of one-step prediction horizon MPC wherein the objective function is defined by the anti-windup stability and performance requirements. More recently, the quadratic program used in online-optimizing anti-windup schemes [8–12] has been shown to be sector bounded, so that analysis tools similar to those for saturation can be applied [13], albeit with a weaker set of available multipliers. The robust needs [14–16] of such anti-windup schemes are however yet to be fully explored. The objective of this thesis is to create a better understanding with regards to the robustness issues and the optimization cost associated with the implementation of these online-optimizing anti-windup schemes.

1.2 Aims of Thesis

The aims of this thesis revolve on the two main areas that constitute online-optimizing anti-windup control:

- Analysis

The internal model control (IMC) anti-windup (IMCA) scheme is a special case of notable anti-windup frameworks [5, 17, 18]). The IMCA scheme and many systems of practical interest can be recast as a Lur'e structure. As noted in [14], the IMCA scheme has guaranteed stability and "optimal robustness" property with respect to the additive plant unstructured uncertainty. The integral quadratic constraint theory has been exploited to extend the results of [14] to more general class of norm-bounded uncertainty [15]. An important conclusion of [15] is that there need not exist an anti-windup scheme that preserves the robustness of the linear controller. This question of robust preservation of anti-windup control systems has facilitated the investigation of a novel robust absolute stability conjecture. This conjecture, called the robust Kalman conjecture, generalizes the question of robust preservation of anti-windup control systems to include any system that can be recast as a Lur'e structure. This thesis presents a verification of this conjecture for first-order SISO plants [19].

- Implementation

Online-optimizing anti-windup schemes [8–12] represent a class of one-step prediction horizon MPC wherein the objective function is defined by the anti-windup stability and performance requirements. Practical implementation of online-optimizing anti-windup schemes requires numerical optimization algorithms that are not only fast but of low complexity-per-iteration. Richter et al [20, 21] provided certification guarantees to real-time model predictive control (MPC) applications based on projected fast gradient methods [22,23]. This, in addition to the efficiency of the fast gradient methods, has motivated the need to develop secant-based variants of the fast gradient method. Though the scope of this research is limited to unconstrained optimization, increased efficiency of unconstrained gradient methods is a recipe for improved practical implementation of large-scale optimization algorithms in MPC and related applications.

1.3 Overview of Thesis

The thesis is organized into two parts - 1) Analysis: In particular, the analysis of robust preservation in anti-windup control. 2) Optimization: Unconstrained minimization using accelerated gradient methods. The thesis consists of 9 chapters. Chapter 2 covers the stability theory of nonlinear systems. The concept of stability, passivity theory and the IQC theory are discussed in this chapter. Chapter 6 covers the theoretical tools for analyzing optimization algorithms. Both Chapter 2 and Chapter 6 serve as mathematical tools useful for discussion of the contributions of this thesis. The remaining chapters of the thesis are organized as follows:

1.3.1 Robustness Preservation in Anti-windup Control

Part I includes three chapters and focusses on analysis of saturated control systems. Chapter 3 reviews the background literature and provide a perspective of the work presented in Chapter 4. Chapter 4 addresses the question of "optimal robustness" [14, 24] of a saturated loop for first-order SISO plants.

Chapter 4:

In this chapter, a robust Kalman conjecture is defined for the robust Lur'e problem. Specifically, it is conjectured that the nonlinearity's slope interval for which robust absolute stability is guaranteed corresponds to the robust interval of the uncertain plant. It is shown that this robust Kalman conjecture is valid for first-order SISO plants perturbed by various norm-bounded unstructured uncertainties (i.e. additive, input-multiplicative and feedback uncertainty). The analysis classifies the appropriate stability multipliers required for verification in these cases.

1.3.2 Accelerated gradient methods

Part II includes four chapters. Chapter 5 discusses introductory background on convex optimization while Chapters 7 and 8 are focussed on developing efficient algorithms for

convex functions.

Chapter 7:

In this chapter, a simple secant-based fast gradient method has been developed for problems whose objective function is convex and well-defined. The proposed algorithm extends the classical Nesterov gradient method by updating the estimate-sequence parameter with secant information whenever possible. This is achieved by imposing a secant condition on the choice of search point. Furthermore, the proposed algorithm **Secant-Based-NGM** embodies an "update rule with reset" that parallels the restart rule recently suggested in [25]. The proposed algorithm applies to a large class of problems including logistic and least-square losses commonly found in the machine learning literature. Numerical results demonstrating the efficiency of the proposed algorithms are analyzed with the aid of performance profiles.

Chapter 8:

This chapter proposes two modified BB-gradient methods that are shown to be globally convergent for all convex functions. The algorithm **Scaled-BBGM** is a gradient method with a scaled Barzilai-Borwein step-size. The discrete Lyapunov theorem has been used to establish that the proposed scaled-BB gradient method is globally convergent. Furthermore, algorithm **Hybrid-BBGM** accelerates the scaled-BB gradient method by relaxing the monotonicity requirement of the discrete Lyapunov theorem.

The summary and recommendations for future research are made in chapter 9.

1.4 Contributions of Thesis

- The presentation in Chapter 4 provides a concise framework for investigating the robust preservation problem in anti-windup systems. The result of [14] is a special case of **Result 1** while **Results 2** and **3** in this chapter are entirely novel.
- The proposed algorithm (**Secant-Based-NGM**) presented in Chapter 7 is novel. The novelty of (**Secant-Based-NGM**) centers on the proposed "update rule

with reset" of the estimate-sequence parameter. Moreover, a theoretical justification is also given for the heuristic restart condition recently suggested in [25].

- The proposed algorithms (**Scaled-BBGM**, **Hybrid-BBGM**) presented in Chapter 8 are novel. The proposed algorithms are distinguished from other modified BB gradient methods in literature [26–35] by the fact that the proposed algorithms neither require any inexact line-search nor any problem-dependent parameter. In particular, the performance of **Hybrid-BBGM** is significantly more efficient than the Globalized BB gradient method reported in [26, 29, 35].

1.5 Publications

- “ A Robust Kalman Conjecture for First-Order Plants ”, *in: Proceedings of 7th IFAC Symposium on Robust Control Design*, Aalborg, Denmark, June 2012.
Authors: Razak Alli-Oke, Joaquin Carrasco, William Heath, Alexander Lanzon.
- “ Two-stage multivariable IMC anti-windup (TMIA) control of a quadruple tank process using a PLC ”, *in: Proceedings of IEEE Multi-conference on Systems and Control*, Antibes, France, October 2014.[★]
Authors: Awo King-Hans, William Heath, Razak Alli-Oke.
[★]This material has not been included in this thesis.
- “ A secant-based Nesterov gradient method for convex functions ”, *Optimization Letters*, 2014 (submitted).
Authors: Razak Alli-Oke, William Heath.

PART I:
Robustness Preservation in
Anti-windup Control

Chapter 2

Stability Theory

This chapter is a basic review of stability concepts. The stability is an important concept in the investigation and characterization of the behaviour of dynamic systems. Stability plays a crucial role in systems theory and control engineering and has been investigated extensively during the past century. The purpose of this chapter is to present some basic definitions and results on stability that are useful for the design and analysis of control systems. The two basic approaches to stability are

- The internal stability, which considers the stability of the state's trajectories of an autonomous system (i.e a system without inputs) $x(t) = f(x(t))$. Internal stability refers to stability of the response to initial conditions only, assuming zero inputs (i.e. free dynamics).
- The external stability, which is concerned with how much a system amplifies input signals. External stability is also known as input-output stability and it refers to stability of the response to the inputs only, assuming zero initial conditions (i.e. forced dynamics). An example of such notions is the "bounded input - bounded output" (BIBO) stability. The system is BIBO-stable if and only if any bounded input signals will necessarily produce bounded output signals.

In general, stability theory attempts to analyze and classify the behaviour/solutions of the underlying (partial) differential equations governing the considered system (without

explicitly solving these equations). However, recent availability of efficient and flexible software, such as MATLAB, has shifted the primal essence of (Lyapunov) stability theory from descriptive theory of solutions to stability analysis and design of stabilizing controllers [36, pg. 219]). Stability theory provides sufficient conditions to verify the stability of a proposed controller for a given application.

Consider an illustrative example to highlight the importance of conducting a thorough stability analysis of a given application, specifically an open-loop process whose dynamic is $G(s) = \frac{1}{s^2 + 4}$. The output of this process is bounded for most sinusoidal input signals; however, the output response becomes unbounded if the input signal is a sinusoidal with a frequency equal to the system's pole. This fact may not be immediately obvious from simulation studies unless the stability of the process has properly been analytically analyzed. Moreover, with stability analysis, one is able to provide with certainty, some measure that characterizes the domain of stability. In addition, stability analysis enables the quantification of admissible model-mismatch and unmodelled dynamics that the real system can tolerate.

In Section 2.1, vector spaces and operator theory are discussed. Section 2.2 presents Lyapunov (internal) stability while Section 2.2 discusses input-output stability vis-a-vis passivity and IQC theorems. In particular, the IQC theorem and the discrete Lyapunov theorem discussed in this chapter are subsequently called upon in Chapter 4 and Chapter 8 respectively.

2.1 Vector Spaces & Norms

A vector space \mathcal{V} is a set that is closed under finite vector addition and scalar multiplication such that for all elements $v_1, v_2, v_3 \in \mathcal{V}$ and any scalars $r, s \in \mathbb{F}$ (*i.e.* \mathbb{R}, \mathbb{C}), the following axioms are satisfied [37].

1. $v_1 + v_2 = v_2 + v_1$ (commutativity of vector addition).
2. $v_1 + (v_2 + v_3) = (v_1 + v_2) + v_3$ (associativity of vector addition).

3. $s(v_1 + v_2) = sv_1 + sv_2$ (distributivity of scalar multiplication over vector addition).
4. $(r + s)v_1 = rv_1 + sv_1$ (distributivity of scalar multiplication over field addition).
5. $r(sv_1) = s(rv_1)$ (associativity of scalar multiplication).
6. $1(s) = s$ (Identity element of scalar multiplication).
7. For all $v \in \mathcal{V}$, there exists an element $-v \in V$ such that $v + (-v) = 0$.
8. There exists an element $0 \in \mathcal{V}$ such that $v + 0 = v$ for all $v \in \mathcal{V}$.

Examples of vector spaces include:

- m dimensional Euclidean space - \mathbb{F}^m .
- The space of all continuous functions defined on $[0, \infty]$: $f : \mathbb{F}^{m+} \rightarrow \mathbb{F}^m$.
- Operators: e.g Matrices and transfer functions.

2.1.1 Banach Spaces

The norm [38] on \mathcal{V} is a non-negative real-valued functional $\|\cdot\|: \mathcal{V} \rightarrow \mathbb{R}^+$ satisfying, for all $u, v \in \mathcal{V}$ and for all $a \in \mathbb{F}$:

1. $\|v\|=0$ if and only if $v=0$.
2. $\|av\|=|a|\|v\|$.
3. $\|u + v\| \leq \|u\| + \|v\|$ (Triangle Inequality).

This norm induces a natural metric $d(u, v) \triangleq \|u - v\|$. A sequence v_k in \mathcal{V} is convergent if the sequence of real numbers $\|v_k - v^*\|$ converges to zero where v^* is the limit of the sequence. A sequence v_k is a Cauchy sequence if

$$\forall \epsilon > 0 \quad \exists \text{ integer } n : \|v_k - v_l\| < \epsilon \quad \forall k, l \geq n. \quad (2.1)$$

If every Cauchy sequence in \mathcal{V} is convergent (i.e. if every sequence which is trying to converge actually does converge), then \mathcal{V} is complete.

Banach Spaces: This is a normed vector space that is complete with respect to the

induced metric.

Lebesgue Spaces $\mathcal{L}_p(S)$ [39]: This is the Banach space of all functions $f : \mathbb{R}_+^m \rightarrow \mathbb{R}^m$ defined on a measurable set $S \in \mathbb{R}_+^m$ for which the p th power of the norm is integrable ($p = 1, 2, \dots, \infty$) i.e. have finite norms $\|f\|_p$:

$$\|f\|_p = \left(\int_S |f|^p dt \right)^{\frac{1}{p}}, \quad (2.2)$$

$$\|f\|_\infty = \lim_{p \rightarrow \infty} \|f\|_p = \text{ess sup}_{t \in \mathbb{R}^{m(+)}} |f(t)|_\infty. \quad (2.3)$$

2.1.2 Hilbert Spaces

Inner Product Spaces [38]: A vector space with an inner product bilinear mapping $\langle \cdot, \cdot \rangle$ from vector space $\mathcal{V} \times \mathcal{V}$ onto \mathbb{F} such that for all $u, v, w \in \mathcal{V}$ and for all $a, b \in \mathbb{F}$:

1. $\langle v, v \rangle$ is real and $\langle v, v \rangle \geq 0$.
2. $\langle v, v \rangle = 0$ if and only if $v = 0$.
3. Linearity: $\langle u, av + bw \rangle = a\langle u, v \rangle + b\langle u, w \rangle$.

The inner product induces a natural norm $\|v\| = \sqrt{\langle v, v \rangle}$.

Hilbert Spaces: This is an inner product space that is complete with respect to the metric induced by its natural norm.

Square-Integrable Lebesgue Spaces $\mathcal{L}_2(S)$: This is the Hilbert space of all functions $f : \mathbb{R}_+^m \rightarrow \mathbb{R}^m$ defined on a measurable set $S \in \mathbb{R}_+^m$ for which the norm is square integrable and with

$$\langle u, v \rangle_{\mathcal{L}_2} \triangleq \int_S \langle u(t), v(t) \rangle. \quad (2.4)$$

2.1.3 Operator Spaces

Operator Spaces: An operator H is a mapping from one normed (Banach) space to another. Consider the case when both spaces are the same i.e $H : \mathcal{L}_p \rightarrow \mathcal{L}_p$ and then

$H \in \mathcal{L}_p$ operator space¹. Consequently, $\|H\|_{\mathcal{L}_p}$ denotes the induced \mathcal{L}_p norm of an operator H and defined by:

$$\|H\|_{\mathcal{L}_p} = \sup_{\substack{v \in \mathcal{L}_p \\ v \neq 0}} \frac{\|H(v)\|}{\|v\|}. \quad (2.5)$$

An operator H is linear if it satisfies

$$H(\alpha f + \beta g) = \alpha H(f) + \beta H(g). \quad (2.6)$$

An example of such linear operator is the transfer function $G(s)$ which is defined on the complex plane $s = \sigma + j\omega$ for $\sigma, \omega \in \mathbb{R}$. For a transfer function $G(s)$, the $\|G\|_{\infty}$ gives a measure of worst-system gain over all frequencies and defined by

$$\|G\|_{\infty} = \|G\|_{\mathcal{L}_2} = \sup_{\omega \in \mathbb{R}} |G(j\omega)|. \quad (2.7)$$

Also, the $\|G\|_2$ is a measure of average-system gain over all frequencies and is defined by

$$\|G\|_2^2 = \int_{-\infty}^{\infty} \text{Trace} [g(t)^* g(t)] dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{Trace} [G(j\omega) * G(j\omega)] d\omega, \quad (2.8)$$

where $g(t)$ and $G(j\omega)$ represents the corresponding (matrix-valued) impulse response and Fourier transform of $G(s)$ respectively.

The Hardy space \mathcal{H}_p is a subspace of \mathcal{L}_p ($p=1, 2, \dots, \infty$) for which $G(s)$ is bounded and analytic in the closed right half s -plane.

Lemma 2.1 ([40]). *The $\|G\|_2$ is finite if and only if $G(s)$ is strictly proper and has no poles on the imaginary axis; the $\|G\|_{\infty}$ is finite if and only if $G(s)$ is proper and has no poles on the imaginary axis (see [40] for proof).*

Thus by Lemma 2.1, \mathcal{H}_{∞} corresponds to the space of (time invariant) proper Hurwitz-stable transfer function matrices and \mathcal{H}_2 corresponds to the space of (time invariant)

¹Operators are defined similarly for general vector-spaces \mathcal{V}, \mathcal{W} .

strictly proper Hurwitz-stable transfer function matrices. The subspace of real rational transfer function matrices in \mathcal{H}_p and \mathcal{L}_p are called \mathcal{RH}_p and \mathcal{RL}_p respectively.

2.1.4 Extended Spaces

This is an extension of a normed vector space which may not be bounded in the norm of their vector spaces but where any truncation to a finite time intervals is bounded.

Let $f : \mathbb{R} \rightarrow \mathcal{V}$, then

$$f_T(t) : \begin{cases} f(t), & t \leq T \in \mathbb{R}, \\ 0, & t \geq T. \end{cases} \quad (2.9)$$

Definition 2.1 ([41]). *The extended space \mathcal{L}_e is defined as*

$$\mathcal{L}_e = \{ f : \mathbb{R} \rightarrow \mathcal{V} : \|f_T\| < \infty, \forall T > 0. \}$$

where $\|\cdot\|$ is the norm on \mathcal{L} . It is assumed that the norm $\|\cdot\|$ is such that

- For every $f \in \mathcal{L}_e$, we have $\|f_{T_1}\| \leq \|f_{T_2}\|$ for all $T_1 \leq T_2$ i.e $\|f_T\|$ is monotonically non-decreasing.
- For every $f \in \mathcal{L}$, we have $\|f_T\| \rightarrow \|f\|$ as $T \rightarrow \infty$.

The above conditions hold for $\mathcal{L}_{pe}(0, \infty)$ spaces, $p = 1, 2, \dots, \infty$.

2.2 Lyapunov Stability Theory

Some of the fundamental concepts of internal stability were introduced by the Russian mathematician and engineer Alexandr Lyapunov in [42]. The Lyapunov stability theory essentially formalizes the idea that if the total energy is dissipated, then the system must be stable. Consider the autonomous (or time-invariant) system

$$\dot{x} = f(x, u), \quad (2.10)$$

where $x(u) \in X(U) \subset \mathbb{R}^{n(m)}$ and $f : D \rightarrow \mathbb{R}^n$ are state(input) vector variables and a locally Lipschitz² map from domain $D \subset \mathbb{R}^n$ into \mathbb{R}^n respectively.

The free dynamics of (2.10) possess an equilibrium point at the origin³ ($\bar{x} = 0$) that is:

1. (*Lyapunov*) *stable* if, for each $\epsilon > 0$, $\exists \delta = \delta(\epsilon) > 0$ such that

$$\|x(0)\| < \delta \implies \|x(t)\| < \epsilon \quad \forall t \geq 0.$$

2. *Asymptotically stable* if it is stable and $\exists \delta$ such that

$$\|x(0)\| < \delta \implies \lim_{t \rightarrow \infty} x(t) = 0.$$

3. *Exponentially stable* if \exists some constants $\rho_1, \rho_2 \geq 0$ such that

$$\|x(t)\| \leq \rho_1 \|x(0)\| e^{-\rho_2 t} \quad \forall t \geq 0.$$

The domain of attraction is the set of initial points x_0 such that the solution of (2.10) is asymptotically or exponentially stable. The origin is *globally lyapunov/asymptotically/exponentially stable* if the domain of boundedness/attraction is the entire state-space $X = \mathbb{R}^n$.

Theorem 2.1 (Discrete Lyapunov Stability Theorem [43,44]). *Let $x^* = 0$ be the equilibrium point of the autonomous system $x_{k+1} = F(x_k)$. Assume that F is locally Lipschitz and let $V : \mathbb{R}^n \rightarrow \mathbb{R}$ be a scalar-valued continuous differentiable function defined on \mathbb{R}^n . Define $\Delta V(x_k) := V(x_{k+1}) - V(x_k)$. Suppose there exists V such that*

- (i) $V(0) = 0$,
- (ii) $V(x_k) > 0$ for all $x_k \neq 0$ in \mathbb{R}^n ,
- (iii) $V(x) \rightarrow \infty$ as $\|x\| \rightarrow \infty$,

then the equilibrium point $x^* = 0$ is

- *globally stable if $\Delta V(x_k) \leq 0$ for all x_k in \mathbb{R}^n .*
- *globally asymptotically stable if $\Delta V(x_k) < 0$ for all $x_k \neq 0$ in \mathbb{R}^n .* ■

²This assumption ensures that (2.10) has unique solutions in the neighbourhood of the local point/initial condition.

³There is no loss of generality since any equilibrium point can be shifted to the origin via change of variables.

Furthermore [45], if there exists constants $c_1 > 0$ and $c_2 > 0$ such that for all x_k in \mathbb{R}^n ,

1. $c_1 \|x_k\|_2^2 < V(x_k) < c_2 \|x_k\|_2^2$,

2. $\Delta V(x_k) < -c_3 \|x_k\|_2^2$,

then the autonomous system $x_{k+1} = F(x_k)$ is globally exponentially stable with respect to the 2-norm.

Theorem 2.2 (Continuous Lyapunov Stability Theorem [46]). *Let $\dot{x} = F(x)$, $F(0) = 0$ where $x \in X \subset \mathbb{R}^n$ and X contains the origin. Assume that F is locally Lipschitz and $V : X \rightarrow \mathbb{R}$ is a continuous differentiable (C^1) function. If*

1. $V(0) = 0$,

2. $V(x) > 0$ for all $x \neq 0$,

3. $\dot{V}(x) < 0$ for all $x \neq 0$,

then $x = 0$ is locally asymptotically stable. If further

4. $\dot{V}(x) \rightarrow \infty$ as $\|x\| \rightarrow \infty$,

then $x = 0$ is globally asymptotically stable.

The Lyapunov function can also be considered as a storage function in the context of dissipative theory.

Definition 2.2 (Dissipation Inequality [47]). *Consider the following system*

$$H : \begin{cases} \dot{x} = f(x, u), \\ y = h(x, u), \end{cases} \quad (2.11)$$

where $x(u)(y) \in X(U)(Y) \subset \mathbb{R}^{l(n)(m)}$. The dissipation inequality states that the increase in its energy (non-negative real storage function) during the interval (t_0, t_1) cannot exceed the energy supplied (via integral of the supply rate) to it.

$$S(x(t_1)) - S(x(t_0)) \leq \int_{t_1}^{t_0} s(u(t), y(t)) dt, \quad (2.12)$$

for all $t_0 \leq t_1$; and all trajectories that satisfy the dynamical equations (2.11).

The Lyapunov function $V(x)$ is a special case of the storage function $S(x)$ in the dissipation inequality, if $S(x)$ is a continuously differentiable (C^1) function and the input is absent ($u = 0 \implies s = 0$).

2.3 Input-Output (IO) Stability Theory

Input-output relation specifies the output in terms of the input. It does not require the internal dynamics to be specified and thus allows consideration of rather general distributed parameter, large scale and nonlinear systems. This section briefly discusses the IQC theorem and passivity theorems, which give the stability conditions in the input-output sense. The input-output relationship is often more conveniently represented by system operators. A system operator represents a mapping from \mathcal{L}_{pe}^n into \mathcal{L}_{pe}^m ; given an input $u(t)$, then the output is given by $y(t)$ i.e.

$$H : u(t) \in \mathcal{L}_{pe}^n \rightarrow y(t) \in \mathcal{L}_{pe}^m \quad \forall t > 0.$$

2.3.1 Causal Systems

An operator is causal if the value of the output signal at time t depends only on the values of the input up to time t i.e for $t \in [0, T]$, the values of $Hu(t)$ depends only on the values of $u(t)$ over $[0, T]$.

Definition 2.3 ([41]). *A mapping $H : u(t) \in \mathcal{L}_{pe}^n \rightarrow y(t) \in \mathcal{L}_{pe}^m$ is said to be causal if*

$$(Hu(t))_T = (Hu_T(t))_T \quad \forall u(t)_T > 0 \quad \forall u \in \mathcal{L}_{pe}^n.$$

It is noncausal if $Hu(t)$ depends on the future (in addition to possible dependence on past or current) values of the input signal $u(t)$. It is anticausal if $Hu(t)$ depends only on the future values.

2.3.2 \mathcal{L} -Stability

Definition 2.4 ([48]). A mapping $H : \mathcal{L}_{pe}^n \rightarrow \mathcal{L}_{pe}^m$ is \mathcal{L}_{pe} stable if there exists a class \mathcal{K} function α , defined on $[0, \infty)$ and a nonnegative constant β such that

$$\|(Hu)_\tau\|_{\mathcal{L}_p} \leq \alpha(\|u_\tau\|_{\mathcal{L}_p}) + \beta, \quad (2.13)$$

for all $u \in \mathcal{L}_{pe}^n$ and $\tau \in [0, \infty)$. It is finite-gain \mathcal{L}_{pe} stable if there exist nonnegative constants λ and β such that

$$\|(Hu)_\tau\|_{\mathcal{L}_p} \leq \lambda\|u_\tau\|_{\mathcal{L}_p} + \beta, \quad (2.14)$$

for all $u \in \mathcal{L}_{pe}^n$ and $\tau \in [0, \infty)$. The definition of \mathcal{L}_∞ corresponds to the familiar concept of bounded-input-bounded-output stability.

Definition 2.5 ([46]). The positive feedback interconnection of H_1 and H_2 given by

$$\begin{cases} u_2 = H_1(u_1) + r_2, \\ u_1 = H_2(u_2) + r_1, \end{cases} \quad (2.15)$$

is well-posed (see Fig. 2.1) if the map $(u_1, u_2) \rightarrow (r_1, r_2)$ defined by (2.15) has a casual inverse on $\mathcal{L}_{pe}^{n+m}[0, \infty)$. The feedback system is said to be stable if it is well-posed and if for any $r_1, r_2 \in \mathcal{L}_{pe}^{n(m)}[0, \infty)$, then $u_1, u_2 \in \mathcal{L}_{pe}^{n(m)}[0, \infty)$. If in addition, the inverse is bounded i.e there exists a constant $\lambda > 0$ such that

$$\int_0^T (|u_1|^2 + |u_2|^2) dt \leq \lambda \int_0^T (|r_1|^2 + |r_2|^2) dt, \quad \text{for any } T \geq 0,$$

then the finite-gain \mathcal{L}_p ($p = 1, 2, \dots, \infty$) stable.

Remark 2.1. In this thesis, it is assumed that the signal spaces belong to \mathcal{L}_{2e} (\subset Hilbert space \mathbf{H}) in order to exploit its additional inner product structure. Furthermore, we would restrict our discussion to square transfer-functions i.e. $u, H(u) \in \mathcal{L}_{2e}^m$.

2.3.3 Passivity Theory

A system operator with input-output pair (u, y) is said to be passive [49] if there exists a nonnegative constant $\beta \geq 0$ such that:⁴

$$-\beta \leq \int_0^T u^T(t)y(t) \quad \forall u(y) \in \mathcal{L}_{2e}^m, \quad \forall T \geq 0. \quad (2.16)$$

2.3.3.1 Positive Real Systems

A commonly encountered input-output property of passive systems e.g electrical networks is positive realness. A system operator with input-output pair (u, y) is said to be positive real [49] if for all $t_1 \geq t_0 \geq 0$, $u(y) \in \mathcal{L}_{2e}^m$,

$$\int_{t_0}^{t_1} y^T(t)u(t)dt \geq 0, \quad \text{whenever } x(t_0) = 0.$$

An LTI system operator is (strictly) passive if and only if its transfer function matrix $G(s)$ is (strictly) positive real [see [48] for proof]. The positive realness property of $G(s)$ can also be defined in the frequency-domain as follows:

Definition 2.6 ([48]). *A real rational proper transfer function matrix $G(s)$ of the complex variable $s = \sigma + j\omega$ is called **Positive Real** if*

1. *all poles of all the elements of $G(s)$ are in $\mathbf{Re}(s) \leq 0$ ie $G(s)$ is analytic in the open RHP ($\mathbf{Re}(s) > 0$).*
2. *any pure imaginary pole $j\omega$ of any elements of $G(s)$ is simple and such that the associated residue matrix $\lim_{s \rightarrow j\omega} (s - j\omega)G(s)$ is positive semidefinite Hermitian (Stability in the Limit).*
3. *The matrix $G(j\omega) + G^*(j\omega) \geq 0$ for all real value ω for which $s = j\omega$ is not a pole of any element of $G(s)$.*

⁴In the state-space approach, this is equivalent to the system being dissipative with respect to supply rate $u^T(t)y(t)$ with the storage function $S(x(t_1))=0$ and $S(0)=\beta$.

Definition 2.7 ([48]). Assume that $\det[G(s) + G^T(-s)]$ is not identically zero for all s . Then $G(s)$ is **Strictly Positive Real (SPR)** i.e $G(s - \epsilon)$ is PR for some $\epsilon > 0$ if

1. $G(s)$ is Hurwitz; poles of all elements of $G(s)$ are in $\mathbf{Re}(s) < 0$ i.e $G(s)$ is analytic in the closed RHP ($\mathbf{Re}[s] \geq 0$).
2. The matrix $G(j\omega) + G^*(j\omega) > 0$, $\forall \omega \in \mathbb{R}$.
3. $G(\infty) + G^*(\infty)$ is positive definite OR it is positive semidefinite and having $(p - q)$ singular values with $\lim_{\omega \rightarrow \infty} \sigma_i(\omega) = 0^+$, i.e

$$\lim_{\omega \rightarrow \infty} \omega^2 M^T [G(j\omega) + G^T(-j\omega)] M \text{ is positive definite}$$

for any $p \times (p - q)$ full-rank matrix M such that

$$M^T [G(\infty) + G^T(\infty)] M = 0,$$

where q is the rank of $[G(\infty) + G^T(\infty)]$. If $[G(\infty) + G^T(\infty)] = 0$ (i.e $\lim_{\omega \rightarrow \infty} \sigma_i(\omega) = 0 \forall i = 1 \dots p$), we can take $M = I$.⁵

2.3.3.2 Passivity Theorems

There are many theorems and lemmas relating to passivity. Here, we state a well-known formulation of the passivity theorem.

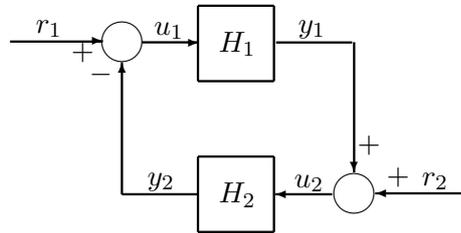


Figure 2.1: Negative Feedback System.

⁵In the scalar case ($p = 1$), these conditions reduces to $G(s)$ being Hurwitz, $\mathbf{Re}(G(j\omega)) > 0 \forall \omega \in \mathbb{R}$ and either $G(\infty) > 0$ or $G(\infty) = 0$ & $\lim_{\omega \rightarrow \infty} \omega^2 \mathbf{Re}[G(j\omega)] > 0$.

Theorem 2.3 (Passivity Theorem [50]). *Consider the well-posed system shown in Fig. 2.1 satisfying the dynamical equations (2.15) with $H_1, H_2 : \mathcal{L}_{2e}^m \rightarrow \mathcal{L}_{2e}^m$. Suppose there exists real constants ϵ_i, δ_i such that:*

$$\langle x, H_i x \rangle \geq \epsilon_i \|x\|_{T_1}^2 + \delta_i \|H_i x\|_{T_2}^2 \quad \forall T_i \geq 0, \forall x \in \mathcal{L}_{2e}^m, i = 1, 2. \quad (2.17)$$

Then the system is \mathcal{L}_2 -stable without bias if $\epsilon_1 + \delta_2 > 0, \epsilon_2 + \delta_1 > 0$.

2.3.4 Integral Quadratic Constraints (IQC) Theory

The unifying framework of integral quadratic constraints (IQC) gives useful input-output characterizations of the structure of an operator on an Hilbert space. Consider the well-posed system shown in Fig. 2.2 satisfying the dynamical equations (2.15) with $H_1, H_2 : \mathcal{L}_{2e}^m \rightarrow \mathcal{L}_{2e}^m$.

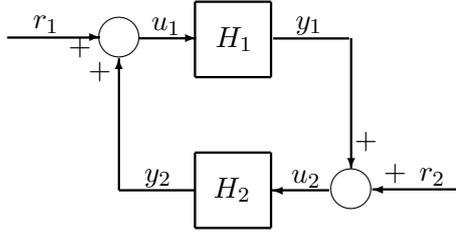


Figure 2.2: Positive Feedback System.

It has been shown in [51] that the internal signals of a feedback interconnection of two stable casual operators of H_1 and H_2 are unique and bounded under external disturbances if and only if the graph⁶ of inverse H_1 (i.e. $\mathcal{G}_{H_1}^I$) and the graph of H_2 (i.e. \mathcal{G}_{H_2}) intersect only at the origin i.e. $\mathcal{G}_{H_1}^I$ and \mathcal{G}_{H_2} are disjoint and topologically separated. It is also shown in [51], that a sufficient condition for topological separation is if for every $\tau \in [0, \infty)$ there exists a separating functional: $d_\tau : \mathcal{L}_{2e} \rightarrow \mathbb{R}$ and $\epsilon > 0$ such that

$$\mathcal{S}_1 : d_\tau(u, y) < -\epsilon(\|u\|^2 + \|y\|^2), \quad \forall (u, y) \in \mathcal{G}_{H_1}. \quad (2.18)$$

$$\mathcal{S}_2 : d_\tau(u, y) \geq 0, \quad \forall (u, y) \in \mathcal{G}_{H_2}^I. \quad (2.19)$$

⁶The graph of an operator is input-output signal pair of the operator.

In the passivity framework, this separating functional is the quadratic form $u^T y$. The use of cones (conic sectors) which are described by quadratic forms seems to be the simplest alternative means of describing \mathcal{S}_1 and \mathcal{S}_2 within spaces containing their operator's graph whilst ensuring topological separation and these conic descriptions has led to the integral quadratic constraint theory.

Remark 2.2. Consider \mathcal{S}_2 described by a cone defined by $\begin{pmatrix} y \\ u \end{pmatrix}^* \begin{pmatrix} \Pi \end{pmatrix} \begin{pmatrix} y \\ u \end{pmatrix} \geq 0$. Then a sufficient condition for \mathcal{L}_{2e} stability is that $\mathcal{G}_{H_1} \in \mathcal{S}_1$:

$$u^* \begin{pmatrix} H_1 \\ I_l \end{pmatrix}^* \begin{pmatrix} \Pi \end{pmatrix} \begin{pmatrix} H_1 \\ I_l \end{pmatrix} u < -\epsilon u^*(I_l)u.$$

In a more elegant fashion, the IQC theorem was stated in [46] with H_1 defined by a stable causal LTI operator $G_p(s)$ and H_2 defined by a stable causal (possibly time-varying nonlinear) operator Δ .

Definition 2.8 ([46]). A operator $\tilde{\Delta}$, with input-output pair $(u, y) \in \mathcal{L}_2^m[0, \infty)$, is said to satisfy the IQC defined by a measurable bounded Hermitian-valued $\Pi : j\mathbb{R} \rightarrow \mathbf{C}^{(2m) \times (2m)}$, if for all $y \in \mathcal{L}_2^m[0, \infty)$ then

$$\int_{-\infty}^{\infty} \begin{bmatrix} \hat{y}(j\omega) \\ \hat{u}(j\omega) \end{bmatrix}^* \Pi(j\omega) \begin{bmatrix} \hat{y}(j\omega) \\ \hat{u}(j\omega) \end{bmatrix} d\omega \geq 0, \quad (2.20)$$

where \hat{z} and \hat{w} are the Fourier transform of the signals y and u respectively.

Theorem 2.4 (IQC Theorem [46]). Let $H_1 = G_p(s) \in \mathcal{RH}_\infty$ and $H_2 = \tilde{\Delta}$ be a bounded casual operator. Assume that:

1. The interconnection between $G_p(s)$ and $\tau\tilde{\Delta}$ is well posed for any fixed $\tau \in [0, 1]$.
2. There exists a measurable Hermitian-valued Π such that the operator $\tau\tilde{\Delta}$ satisfies the IQC defined by Π for all $\tau \in [0, 1]$.

3. There exists $\epsilon > 0$ such that

$$\begin{bmatrix} G_p(j\omega) \\ I \end{bmatrix}^* \Pi(j\omega) \begin{bmatrix} G_p(j\omega) \\ I \end{bmatrix} \leq -\epsilon I, \quad \forall \omega \in \mathbb{R}. \quad (2.21)$$

Then, the feedback interconnection of $G_p(s)$ and $\tilde{\Delta}$ is \mathcal{L}_2 -stable.

Remark 2.3. If Π is congruent to $\begin{pmatrix} I_m & 0 \\ 0 & -I_l \end{pmatrix}$ (which is the case in many applications), then $\tau\tilde{\Delta}$ satisfies the IQC defined by Π for all $\tau \in [0, 1]$ if and only if $\tilde{\Delta}$ does so. This simplifies Assumption 2).

Unlike the passivity theory, implicit incorporation of linear transformations/multipliers and the ease in combining nonlinearities/uncertainties that satisfies conic sector restrictions makes IQC framework attractive for stability analysis of nonlinear systems. The following well-known lemma is also useful in the treatment of the subject matter discussed in this thesis.

Lemma 2.2 (Schur Complement Lemma [52]). Given $P \in \mathbb{F}^{n \times m}$, $S \in \mathbb{F}^{n \times m}$ and $Q \in \mathbb{F}^{n \times m}$. Then,

$$\begin{bmatrix} P & S \\ S^* & Q \end{bmatrix} > 0 \iff \begin{cases} (i) & Q > 0 \text{ and } P - SQ^{-1}S^* > 0 \\ (ii) & P > 0 \text{ and } Q - S^*P^{-1}S > 0. \end{cases} \quad (2.22)$$

2.4 Summary

In this chapter, the concept of stability has been discussed. The stability of a (feedback) system has been analyzed either from the perspective of internal stability (i.e. Lyapunov theory) and its input-output properties (i.e. passivity and IQC theory). In particular, the discrete Lyapunov theorem, the IQC theorem and the Schur Complement Lemma discussed in this chapter are key mathematical tools required for the contributions presented in Chapter 4 and Chapter 8 respectively.

Chapter 3

Introductory Background on Anti-Windup Control

This chapter provides background information on anti-windup control and justifies the use of the Lur'e structure as a basis of the proposed general framework for addressing the issue of robust preservation in anti-windup schemes.

All practical systems are inherently nonlinear. Most commonly, the nonlinearity is as a result of saturation constraints and switching modes. Examples include constraints on valve openings, throttle openings, flight control surface deflection, safety limits, engine-nozzle openings, fuel flows in aerospace systems and boiler regulators. Saturation constraints are ubiquitous and critical for practical applications, hence considerable research activity has been focused on constrained control systems design.

Saturation constraints cause a mismatch between the controller input and the controller output. Thus, the saturated controller output is unaware of the controller input and therefore makes the feedback loop to run as open loop [53]. As a result, the controller output "winds up". The presence of slow or unstable dynamics (e.g. integrators) will also cause the controller output to "windup" and hence require that the error has opposite sign for a long period before the feedback loop is active. As a result a significant transient/overshoot must decay before the system returns to the linear regime. This behaviour results in degradation of controller performance. The aforementioned

destabilizing effects have been cited as contributing factors in several mishaps involving high performance aircraft [1] where pilot-induced oscillations were partly caused by rate saturation of control surfaces inducing time-delay effects in the control loop. It has also been attributed to the Chernobyl disaster [54] where limits in the rate of change of the actuator pushing the control rods into the core, aggravated an already dangerous situation. See Chapter 1 for simulation responses of a simple illustrative example.

3.1 Anti-Windup Control

There are two approaches in design of control systems subject to saturation constraints.

1. The a priori design approach is a *one-step approach* in which a (possibly nonlinear) control design satisfies all nominal performance specifications whilst implicitly satisfying such saturation constraints. Typical examples include model predictive controllers (MPC) [55] and nonlinear output feedback dynamic compensators [56, 57].
2. The a posteriori design strategy is a two-step approach in which a linear control design satisfying all nominal performance specifications is performed first, then an additional (*anti-windup*) *compensator* to the linear controller is designed to minimize the undesirable effects of anti-windup which can occur during saturation [58].

3.1.1 Anti-Windup Compensators

The design requirements of an anti-windup compensator as formalized in [59–61] include

- Closed-loop system stability
- Linear Performance Recovery: Recovery of the linear performance in the absence of saturation.
- Graceful Performance Degradation: Smooth degradation of the linear performance in the presence of saturation.

The "controller windup" phenomenon was conceived as the inconsistency between the internal states of the controller and the saturated controller output \hat{u} [53,62,63]. Many different schemes have appeared in the literature since the anti-windup methodology of Fertik and Ross [64]. Using the difference between the saturated \hat{u} and unsaturated controller output u , the generic anti-windup framework [18] (see Fig. 3.1) seeks to restore the consistency between the saturated controller output \hat{u} and the controller's internal states. This is achieved by conditioning the linear controller with two signals $\begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} = \begin{bmatrix} \Lambda_1(s) \\ \Lambda_2(s) \end{bmatrix} (\hat{u} - u)$, affecting the controller output and the controller states respectively.

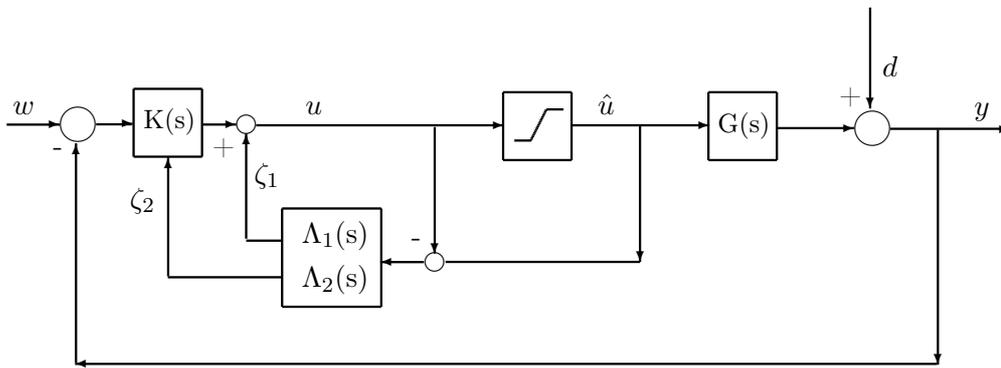


Figure 3.1: Generic Anti-Windup Framework.

A direct equivalence between generic anti-windup framework (Fig. 3.1) and classical feedback structure with anti-windup (Fig. 3.2) can be found in [9]. In particular,

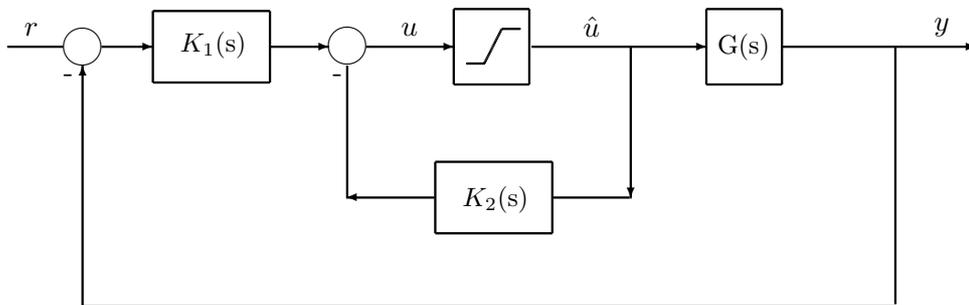


Figure 3.2: Classical Feedback Structure with Anti-Windup.

Kothare et. al. [5] considered the case when $\begin{bmatrix} \Lambda_1(s) \\ \Lambda_2(s) \end{bmatrix}$ are constant matrix parameters. In this case, $K_1(s) = U(s)$ and $K_2(s) = I - V(s)$ where $U(s), V(s)$ is a left coprime factorization of the controller $K(s)$ given by

$$[V(s) \quad U(s)] = \left[\begin{array}{c|cc} A - H_1C & -H_1 & B - H_1D \\ \hline H_2C & H_2 & H_2D \end{array} \right], \quad (3.1)$$

and

$$H_1 = \Lambda_1(1 + \Lambda_2)^{-1}, \quad (3.2)$$

$$H_2 = (1 + \Lambda_2)^{-1}. \quad (3.3)$$

If H_1 is chosen so that $A - H_1C$ is stable, then $U(s), V(s)$ is a stable left coprime factorization of the controller $K(s)$. It was also shown in [5] that with proper selection of $\begin{bmatrix} \Lambda_1(s) \\ \Lambda_2(s) \end{bmatrix}$ as constant matrix parameters, the generic framework in Fig. 3.1 (and therefore Fig. 3.2) unifies a large class of existing anti-windup control schemes.

The analysis and synthesis for the generic framework (Fig. 3.1) was extended in [18] to the more general case where $\begin{bmatrix} \Lambda_1(s) \\ \Lambda_2(s) \end{bmatrix}$ are dynamic transfer functions. Here, the authors showed that for stable plants, there always exists an anti-windup compensator of order greater than or equal to that of the plant which, in addition, satisfies an \mathcal{L}_2 performance objective.

However, in most engineering problems, it is preferable to inject the compensation signals directly into the controller input and controller output respectively. To this end, an alternative framework called external anti-windup framework was introduced in [17], where the conditioning signal $\begin{bmatrix} \Theta_1(s) \\ \Theta_2(s) \end{bmatrix} (\hat{u} - u)$ enters the controller input and controller

output respectively. It is instructive to note that with the transformation,

$$\begin{bmatrix} \Lambda_1(s) \\ \Lambda_2(s) \end{bmatrix} = \begin{bmatrix} B & 0 \\ D & I \end{bmatrix} \begin{bmatrix} \Theta_1(s) \\ \Theta_2(s) \end{bmatrix}, \quad (3.4)$$

the external anti-windup framework is equivalent to the generic framework (Fig. 3.1). Nevertheless, it is also easy to show the direct equivalence of the external anti-windup framework to classical feedback structure with anti-windup (Fig. 3.2). Furthermore, the classical feedback structure with anti-windup Fig. 3.2 can subsequently be recast as a Lur'e structure (Fig. 3.3),

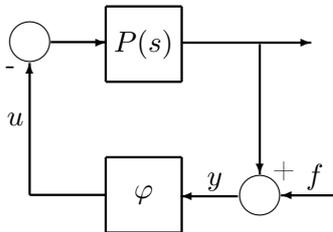


Figure 3.3: The Lur'e Structure.

where φ represents the saturation nonlinearity and $P(s) = K_1(s)G(s) + K_2(s)$.

3.1.2 Robust-Preserving Anti-windup

The robustness of the classical anti-reset anti-windup has been examined in [72]. The external anti-windup framework [17] has also been extensively studied (e.g. [65–69]) and robustness considered in [14, 15, 24, 70, 71]. An anti-windup scheme that preserves the robustness of the unconstrained closed-loop system is said to be "optimally robust" [24]. The IMC anti-windup was reported in [14] to be "optimally robust" with respect to the additive plant unstructured uncertainty. Furthermore, an important conclusion of [15] is that there need not exist an anti-windup scheme that preserves the robustness of the linear controller. In general, we are interested in knowing if a proposed anti-windup scheme is "optimally robust". If indeed "optimal robustness" holds, then the designer can easily estimate the amount of uncertainty a given anti-windup scheme can tolerate

by simply analyzing the linear counterpart.

3.2 Summary

This chapter has discussed some anti-windup frameworks available in the literature and has shown that the presented anti-windup schemes could be transformed into an equivalent Lur'e structure. In chapter 4, a fairly general and abstract framework (based on the Lur'e structure) is presented to study the issue of this robust preservation for first-order SISO plants.

Chapter 4

A Robust Kalman Conjecture

4.1 Introduction

This chapter presents a robust absolute stability conjecture that serves as a fairly general framework to study the issue of robust preservation in anti-windup systems. It has previously been noted in Chapter 3 that most anti-windup systems can be represented as a Lur'e structure (see Fig. 4.1).

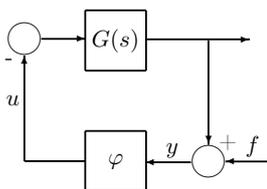


Figure 4.1: The Lur'e Structure.

Absolute stability theory studies the stability of the well-posed Lur'e structure for all nonlinearities φ from a given class of nonlinearities Φ . Determining the conditions for which the Lur'e structure loses its absolute stability has long attracted the interests of researchers since it was posed by Lur'e and Postnikov in 1944. The Lur'e problem represents a particular case of general nonlinear systems wherein the nonlinearity is separable and this includes feedback linear systems with saturation constraints.

An operator $\varphi : \mathcal{L}_e \rightarrow \mathcal{L}_e$ is static if $\exists N : \mathbb{R} \rightarrow \mathbb{R}$ such that $(\varphi y)(t) = N(y(t))$ and

it is monotone non-decreasing if

$$[N(y_2) - N(y_1)][y_2 - y_1] \geq 0 \quad \forall y_1, y_2 \in \mathbb{R}.$$

Typically, conic conditions are used to further describe the families of nonlinearities.

Given $\alpha, \beta \in \mathbb{R}$, a nonlinear operator is classified as follows:

- Sector-Bounded i.e. SB $[\alpha, \beta]$:

$$\alpha y_1^2 \leq y_1 N(y_1) \leq \beta y_1^2.$$

- Slope-Restricted i.e. SR $[\alpha, \beta]$:

$$\alpha(y_2 - y_1)^2 \leq [N(y_2) - N(y_1)][y_2 - y_1] \leq \beta(y_2 - y_1)^2.$$

Without loss of generality any SB $[\alpha, \beta]$ or SR $[\alpha, \beta]$ can be mapped to SB $[0, k]$ or SR $[0, k]$ respectively by loop transformations [7]. Two absolute stability conjectures have been proposed to answer the Lur'e problem. These two conjectures are stated as :

1. **Aizerman Conjecture [73]:** The feedback interconnection between a linear plant $G(s)$ with any sector-bounded SB $[\alpha, \beta]$ nonlinearity φ_k is stable if the feedback interconnection between $G(s)$ and any constant gain $K \in [\alpha, \beta]$ is stable.
2. **Kalman Conjecture [73]:** The feedback interconnection between a linear plant $G(s)$ with slope-restricted SR $[\alpha, \beta]$ nonlinearity φ_k is stable if the feedback interconnection between $G(s)$ and any constant gain $K \in [\alpha, \beta]$ is stable.



If the system in Fig. 4.2(a) is stable, then system in Fig. 4.2(b) is stable.

Figure 4.2: The Absolute Stability Conjecture.

Even though these conjectures have been shown to be false in general, they have played a vital role in rigorous development of modern absolute stability theory. For first and second-order plants, the Aizerman conjecture has been shown to be true using the circle

criterion and Popov criterion respectively [73](see the references therein). The Aizerman conjecture has been refuted for the generality of third-order plants by Pliss [73]. Thus, if stability is established via multipliers, the constant multipliers and Popov multipliers provide the needed stability multipliers for first-order and second-order strictly-proper stable plants respectively. The Kalman conjecture has been proved to be valid for third-order plants [74], [75] by constructing an allowed multiplier that can be interpreted as a first-order Zames-Falb multiplier [76]. It also has been shown in [74] that for n^{th} -order plants with $n \geq 4$, there exists a Lur'e system with a nontrivial periodic solution and therefore not satisfying the Kalman conjecture.

Plant Order	Aizerman	Kalman
1 st	True	True
2 nd	True	True
3 rd	False	True
$\geq 4^{\text{th}}$	False	False

The aim of this work is to study the Lur'e problem (see Fig. 4.1) when the stable linear plant is considered with (possibly nonlinear) uncertainty (see Fig. 4.3). We focus our attention on the family of *scalar* static nonlinearities slope-restricted in the interval $[0, k]$, henceforward, φ_k .

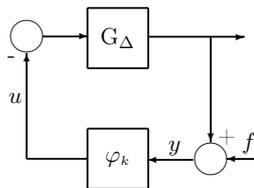


Figure 4.3: The Robust Lur'e problem.

Most of the literature on this problem has been devoted to parametric uncertainty [77] using Kharitonov's Theorem [78] or equivalent results. For norm-bounded LTI unstructured uncertainties, a robust circle criterion and robust Popov criterion for additive and multiplicative LTI uncertainties were presented in [79].

The integral quadratic constraint (IQC) theory [46] provides a unified framework

where the norm-bounded general uncertainties can easily be embedded with the nonlinear block. With the IQC framework, the results in [79] can be extended to other robust stability criteria for different (possibly nonlinear) uncertainty descriptions.

In this thesis, a conjecture is proposed for the robust Lur'e problem that may be true for low-order stable plants but will be false in general just as with the Aizerman and Kalman conjectures. In particular, this analysis investigates the appropriate stability multiplier needed to verify this conjecture for first-order stable plants with various norm-bounded unstructured certainties,

- Additive Uncertainties: $G_\Delta = G(s) + \bar{w}\Delta$,
- Input-Multiplicative Uncertainties: $G_\Delta = G(s) + G(s)\bar{w}\Delta$,
- Feedback Uncertainties: $G_\Delta = G(s)[1 + \bar{w}\Delta G(s)]^{-1}$,

where $\bar{w} \in \mathbb{R}_+$ and Δ satisfying $\|\Delta\|_{\mathcal{L}_2} \leq 1$.

4.2 Notation

Let \mathbb{R}_+ be the set of all non-negative real numbers and \mathbf{RH}_∞ denote real rational stable transfer functions. RKC means robust Kalman conjecture. LFT denotes Linear Fractional Transformation while $\mathcal{L}_p^n[0, \infty)$ denotes the space of p -integrable functions $f : [0, \infty) \rightarrow \mathbb{R}^n(-\infty, \infty)$. The \mathcal{L}_p norm is thus defined by $\|f\|_p^p = \int_0^\infty |f|^p dt$.

A truncation of the function f at T is given by $f_T(t) = f(t)$, $\forall t \leq T$ and $f_T(t) = 0$, $\forall t > T$. $\mathcal{L}_{2e}^n[0, \infty)$ represents a space of extended functions whose truncations at any finite time are square integrable.

Let the operator S be a map from $\mathcal{L}_{2e}^n[0, \infty)$ to $\mathcal{L}_{2e}^n[0, \infty)$, with input u and output Su . This operator S is causal if $Su(t) = S(u_T)(t)$ for all $t < T$. Moreover, S is \mathcal{L}_2 -stable if for all $u \in \mathcal{L}_2^n[0, \infty)$, then $Su \in \mathcal{L}_2^n[0, \infty)$. Furthermore, the operator S is bounded and finite-gain \mathcal{L}_2 -stable if there exists a constant γ such that $\|Su\|_2 \leq \gamma\|u\|_2$. The supremum of such constants defines $\|S\|_{\mathcal{L}_2}$.

Consider the feedback interconnection of a stable LTI SISO plant G and a bounded

operator φ , shown in Fig. 4.1:

$$\begin{cases} y = f - Gu, \\ u = \varphi y. \end{cases} \quad (4.1)$$

Since G is a stable LTI SISO plant, any exogenous input in this part of the loop can be taken as the zero signal without loss of generality while f denotes disturbances or nonzero initial conditions of the plant (see [80]). It is well-posed if the map $(y, u) \rightarrow (0, f)$ has a causal inverse on $\mathcal{L}_{2e}^{2n}[0, \infty)$. Furthermore, this interconnection is \mathcal{L}_2 -stable if for any $f \in \mathcal{L}_2^n[0, \infty)$, then $(Gu, \varphi y) \in \mathcal{L}_2^n[0, \infty)$. The transfer function of the linear plant G is denoted by $G(s)$. Subsequently, it is assumed that the interconnected system in Fig. 4.3 is well-posed by requiring that $G(s)$ be strictly-proper.

4.3 Robust Interval

With the nonlinearity φ_k in Fig. 4.3 replaced by a static gain K , we then have an uncertain system as shown in Fig. 4.4(a). G_Δ is an uncertain stable plant that belongs to a family of stable plants \mathcal{G}_Δ defined in terms of norm-bound (possibly nonlinear) uncertainties Δ , where $\|\Delta\|_{\mathcal{L}_2} \leq 1$.



Using LFT, the system in Fig. 4.4(a) is transformed into Fig. 4.4(b).

Figure 4.4: The Uncertain System.

Now, a well-known theorem is stated. The sufficiency and necessity of Theorem 4.1 follows from small-gain argument [50] and contradiction argument respectively.

Theorem 4.1. *Assume the feedback system in Fig. 4.4(b) is well-posed. Suppose that both $\tilde{G}_K(s) \in \mathbf{RH}_\infty$ and Δ are causal and finite-gain \mathcal{L}_2 -stable. Under these conditions,*

the feedback system in Fig. 4.4(b) is \mathcal{L}_2 -stable for all Δ with $\|\Delta\|_{\mathcal{L}_2} \leq 1$ if and only if $\|\tilde{G}_K\|_\infty < 1$.

The robust interval $\mathcal{I}_r = [0, K_r)$ is the largest interval such that the feedback interconnection of any plant $G_\Delta \in \mathcal{G}_\Delta$ with a constant gain $K \in \mathcal{I}_r$ is stable. Thus, K_r is the supremum of \mathcal{I}_r for which the feedback system shown in Fig. 4.4(a) is stable.

4.3.1 Graphical Interpretation

From the proof of Theorem 4.1, we find that there exists an LTI uncertainty Δ_{LTI} that renders the feedback system shown in Fig. 4.4(a) unstable. Thus in this section, we develop a graphical interpretation of the robust interval for such LTI uncertainties Δ_{LTI} .

When there is no uncertainty (i.e. $\Delta = 0$, $G_\Delta = G$) in Fig. 4.4(a), then the following theorem holds:

Theorem 4.2 (Nyquist Criterion [39]). *The feedback interconnection of a rational stable transfer function $G(s)$, i.e. $G(s) \in \mathbf{RH}_\infty$, and a constant gain K is stable if and only if*

- $\inf_{\omega \in \mathbb{R}} |1 + KG(j\omega)| \neq 0$,
- $KG(j\omega)$ does not encircle the $-1 + 0j$ point.

Now consider the case for which the uncertainty in Fig. 4.4(a) is linear (i.e. Δ_{LTI}). The uncertain plant $G_{\Delta_{LTI}}$ now belongs to a family of stable LTI plants $\mathcal{G}_{\Delta_{LTI}}$. Theorem 4.2 can then be extended to the (LTI) robust case as follows:

Corollary 4.1. *Let $\mathcal{G}_{\Delta_{LTI}}$ be a family of stable rational transfer functions. The feedback interconnection of a $G_{\Delta_{LTI}}(s) \in \mathcal{G}_{\Delta_{LTI}}$ and a constant gain K is stable if and only if*

$$\inf_{\omega \in \mathbb{R}} |1 + KG_{\Delta_{LTI}}(j\omega)| > 0. \quad (4.2)$$

Graphically, $\mathcal{G}_{\Delta_{LTI}}$ is represented by a “loose” region about the Nyquist plot of

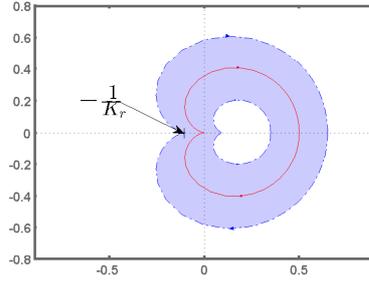


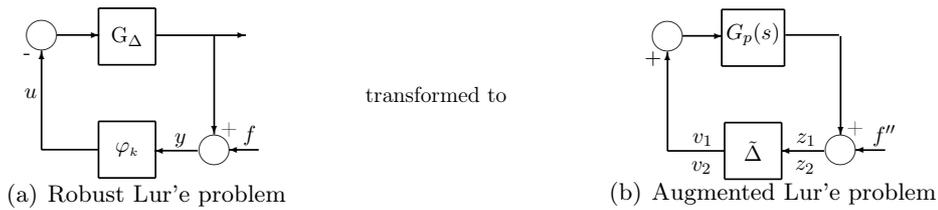
Figure 4.5: Robust Nyquist Plot.

This example shows the plot of a nominal transfer function of a 2nd-order plant. The blue-shaded region signifies the “loose” region of the norm-bounded uncertainty.

the nominal plant model (see Fig. 4.5). Corollary 4.1 provides the interval of gains for which elements of $\mathcal{G}_{\Delta_{LTI}}$ in feedback interconnection are stable. The supremum of such gains is denoted by bound K_r and parameterizes the robust interval. Thus the “loose” region cannot include the real interval $(-\infty, -1/K_r]$.

4.4 Robust Absolute Stability

In this section, the robust absolute stability of the robust Lur’e problem (Fig. 4.3) is analysed. The uncertainty Δ is combined with nonlinearity φ_k to form $\tilde{\Delta}$ defined in (4.5) and an augmented Lur’e problem is obtained, see Fig. 4.6.



Using LFT, the system in Fig. 4.6(a) is transformed into Fig. 4.6(b).

Figure 4.6: Augmented Lur’e problem.

The unifying framework of integral quadratic constraints (IQC) gives useful input-output characterizations of the structure of an operator on a Hilbert space. IQCs are defined by quadratic forms which are in turn defined in terms of bounded self-adjoint operators. With the IQC framework, the norm-bounded general uncertainties can easily

be embedded with the nonlinear block.

Definition 4.1 ([46]). A bounded operator $\tilde{\Delta} : \mathcal{L}_{2e}^n \rightarrow \mathcal{L}_{2e}^n$, with input z and output v , is said to satisfy the IQC defined by a measurable bounded Hermitian-valued $\Pi : j\mathbb{R} \rightarrow \mathbb{C}^{(2n) \times (2n)}$, if for all $z \in \mathcal{L}_2^n$,

$$\int_{-\infty}^{\infty} \begin{bmatrix} \hat{z}(j\omega) \\ \hat{v}(j\omega) \end{bmatrix}^* \Pi(j\omega) \begin{bmatrix} \hat{z}(j\omega) \\ \hat{v}(j\omega) \end{bmatrix} d\omega \geq 0, \quad (4.3)$$

where \hat{z} and \hat{v} are the Fourier transform of the signals z and w respectively.

Theorem 4.3 (IQC Theorem [46]). Consider the feedback interconnection in Fig. 4.6(b). Let $G_p(s) \in \mathbf{RH}_\infty$ and let $\tilde{\Delta}$ be a bounded causal operator. Assume that:

- (i) The feedback interconnection between $G_p(s)$ and $\tau\tilde{\Delta}$ is well posed for all $\tau \in [0, 1]$.
- (ii) There exists a measurable Hermitian-valued Π such that the operator $\tau\tilde{\Delta}$ satisfies the IQC defined by Π for all $\tau \in [0, 1]$.
- (iii) There exists $\epsilon > 0$ such that

$$\begin{bmatrix} G_p(j\omega) \\ I \end{bmatrix}^* \Pi(j\omega) \begin{bmatrix} G_p(j\omega) \\ I \end{bmatrix} \leq -\epsilon I \quad \forall \omega \in \mathbb{R}. \quad (4.4)$$

Then, the feedback system in Fig. 4.6(b) is \mathcal{L}_2 -stable.¹

For different uncertainty descriptions, the generalized plant $G_p(s)$ will have different forms whereas the augmented nonlinearity will preserve the diagonal structure

$$\tilde{\Delta} = \begin{bmatrix} \varphi_k & 0 \\ 0 & \Delta \end{bmatrix}, \quad (4.5)$$

where φ_k is a SR $[0, k]$ nonlinearity. Δ is a causal and bounded operator. Thus, $\Pi(j\omega)$

¹Condition (iii) is satisfied if the LHS of (4.4) is strictly negative (i.e less than 0).

will have the same structure

$$\Pi(j\omega) = \begin{bmatrix} 0 & 0 & kM(j\omega)^* & 0 \\ 0 & 1 & 0 & 0 \\ kM(j\omega) & 0 & -M(j\omega) - M(j\omega)^* & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad (4.6)$$

throughout this work, where the positive operator $M(j\omega)$ is a stability multiplier for φ_k [76]. Depending on the stability criteria, $M(j\omega)$ has different parametrization:

- Constant Multipliers: $M(j\omega) = \eta : \eta > 0$.
- Popov Multipliers: $M(j\omega) = \eta + j\omega\lambda : \eta > 0, \lambda \in \mathbb{R}$.
- Zames-Falb Multipliers: $M(j\omega) = \eta + H(j\omega) : \eta > 0$ and $\|h\|_1 < \eta$ where h is the impulse response of $H(j\omega)$.

Remark 4.1. *Due to the structure of the augmented nonlinearity $\tilde{\Delta}$, assumption (i) is guaranteed if it is well-posed for $\tau = 1$.*

Remark 4.2. *The existence of this stability multiplier $M(j\omega)$ ensures that $M(j\omega) \times (G_\Delta + \frac{1}{k})$ is strictly positive real (SPR) [81].*

4.5 Robust Kalman Conjecture

This section presents a key contribution towards addressing robust preservation in anti-windup schemes. As noted in Chapter 3, most anti-windup schemes can be recast as a Lur'e structure. This section provides a general and concise framework which seeks to ascertain if a Lur'e structure has the same measure of robustness as its linear counterpart.

Robust Kalman Conjecture (RKC).

Suppose G_Δ is an uncertain stable plant that belongs to a family of stable plants \mathcal{G}_Δ defined in terms of norm-bound (possibly nonlinear) uncertainties Δ , where $\|\Delta\|_{\mathcal{L}_2} \leq 1$. If the feedback interconnection between G_Δ and any constant gain $K \in [0, K_r)$ is

stable, then the feedback interconnection of G_Δ and any slope-restricted nonlinearity $\varphi_k : \text{SR}[0, K_r)$ is stable.



If the system in Fig. 4.7(a) is stable, then system in Fig. 4.7(b) is stable.

Figure 4.7: Robust Kalman Conjecture (RKC).

The RKC provides the condition for which the robust absolute stability of the robust Lur'e problem is exact. The key idea of verifying this Robust Kalman Conjecture is *determining if the robust interval of the uncertain plant coincides with the slope interval of φ_k for which the robust Lur'e structure is absolutely stable*. In subsequent sections, we investigate and study the class of stability multipliers required to verify this conjecture for first-order stable plants with various uncertainty descriptions.

This thesis consider the case where $G(s)$ is a first-order plant given by,

$$G(s) = \frac{a}{s + b}, \quad a > 0 \text{ and } b > 0. \quad (4.7)$$

4.5.1 ADDITIVE UNCERTAINTY

Consider the well-posed uncertain nonlinear system with $\bar{w} \in \mathbb{R}_+$ and Δ satisfying $\|\Delta\|_{\mathcal{L}_2} \leq 1$ as shown in Fig. 4.8. Here, $G_\Delta = G(s) + \bar{w}\Delta$.

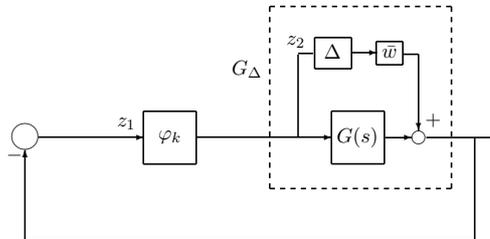


Figure 4.8: Robust Lur'e for Additive Uncertainty.

The generalized plant for the additive uncertainty which is obtained by transforming

the robust Lur'e problem in Fig. 4.8 into the augmented Lur'e problem in Fig. 4.6, is given by

$$G_p(s) = \begin{bmatrix} -G(s) & -\bar{w} \\ 1 & 0 \end{bmatrix}. \quad (4.8)$$

Remark 4.3. *It is assumed that the interconnected system in Fig. 4.6 is well-posed. Then, due to the structure of $\tilde{\Delta}$, well-posedness for any $\tau \in [0, 1]$ is guaranteed.*

Corollary 4.2. *Consider the nonlinear system with additive uncertainty as shown in Fig. 4.8. Then, the system is \mathcal{L}_2 -stable if there exist a stability multiplier $M(j\omega)$ such that*

$$kM(j\omega)G(j\omega) + kG(j\omega)^*M(j\omega)^* + M(j\omega) + M(j\omega)^* - 1 - \bar{w}^2k^2(j\omega)M(j\omega)^* > 0 \quad (4.9)$$

for all $\omega \in \mathbb{R}$.

Proof. If there exists $M(j\omega)$ that satisfies (4.9) for all $\omega \in \mathbb{R}$, then applying Schur complements, $M(j\omega)$ also satisfies

$$\begin{bmatrix} M(j\omega)(kG(j\omega) + I) + (kG(j\omega) + I)^*M(j\omega)^* - 1 & \bar{w}M(j\omega)k \\ \bar{w}M(j\omega)^*k & 1 \end{bmatrix} > 0 \quad (4.10)$$

for all $\omega \in \mathbb{R}$. Thus using $M(j\omega)$ in (4.6), the obtained $\Pi(j\omega)$ fulfils the conditions of Theorem 4.3 and therefore the feedback interconnection in Fig. 4.8 is \mathcal{L}_2 -stable. ■

Lemma 4.1. *Let $G_\Delta \in \mathcal{G}_\Delta$ be a first-order plant (4.7) with additive uncertainty i.e. $G_\Delta = G + \bar{w}\Delta$ as shown in Fig. 4.8, then the robust interval $\mathcal{I}_r [0, K_r]$ of G_Δ is $[0, 1/\bar{w}]$.*

Proof. The result is obtained by applying Theorem 4.1. The system in 4.8 is stable if and only if,

$$\left\| \frac{K\bar{w}(s+b)}{s+b+Ka} \right\|_{\mathcal{L}_2} \|\Delta\|_{\mathcal{L}_2} < 1. \quad (4.11)$$

It then follows that an equivalent condition is

$$\left\| \frac{K\bar{w}(s+b)}{s+b+Ka} \right\|_{\mathcal{L}_2} = \frac{K\bar{w}(s+b)}{s+b+Ka} \Big|_{s \rightarrow \infty} < 1. \quad (4.12)$$

Thus it follows from (4.12) that $K < \frac{1}{\bar{w}}$ and $K_r = \frac{1}{\bar{w}}$. ■

Once we have obtained the maximum interval for which the nonlinear system could be stable, IQC machinery is then used to study the absolute stability of the robust Lur'e problem. The generalized plant in (4.8) is given by

$$G_p(s) = \begin{bmatrix} -\frac{a}{s+b} & -\bar{w} \\ 1 & 0 \end{bmatrix}. \quad (4.13)$$

Result 4.1. *Let $G_\Delta \in \mathcal{G}_\Delta$ be a first-order plant (4.7) with additive uncertainty as shown in Fig. 4.8. By virtue of Lemma 4.1, the robust interval (\mathcal{I}_r) of G_Δ is defined as $[0, 1/\bar{w})$. Thus the system in Fig. 4.8 is \mathcal{L}_2 -stable for any $\varphi_k \in \text{SR}[0, \frac{1}{\bar{w}})$ nonlinearity. Moreover, the stability multiplier $M(j\omega) = \frac{1}{k^2\bar{w}^2}$ is sufficient to establish this result.*

Proof. It is shown that there exists an admissible constant multiplier $M(j\omega) = \eta$ with $\eta > 0$, such that (4.9) in Corollary 4.2 is satisfied.

Choose $M(j\omega) = \frac{1}{k^2\bar{w}^2}$, then the stability condition (4.9) reduces to

$$\frac{1}{k\bar{w}^2} \frac{2ab}{\omega^2 + b^2} + 2\left(\frac{1}{k^2\bar{w}^2} - 1\right) > 0 \quad \forall \omega \in \mathbb{R}. \quad (4.14)$$

The first term of (4.14) is always positive and the second term is positive provided $k < \frac{1}{\bar{w}}$.

Thus, $M(j\omega) = \frac{1}{k^2\bar{w}^2}$ satisfies the conditions in Theorem 4.3. As a result, the system in Fig. 4.8 is \mathcal{L}_2 -stable for any $\varphi_k \in \text{SR}[0, \frac{1}{\bar{w}})$ nonlinearity. ■

Remark 4.4. *Result 4.1 demonstrates that the robust Kalman conjecture is true for first-order plant with additive uncertainty. As indicated in Table 4.2, the robust circle criterion provides an appropriate constant multiplier for this purpose.*

Graphical Interpretation of Result 4.1

Given the linear uncertain plant as $G_{\Delta_{LTI}}(s) = G(s) + \bar{w}\Delta_{LTI}$, the “loose” region is defined by discs of radius $\bar{w} = 0.01$ centered at each frequency as shown in Fig. 4.9.

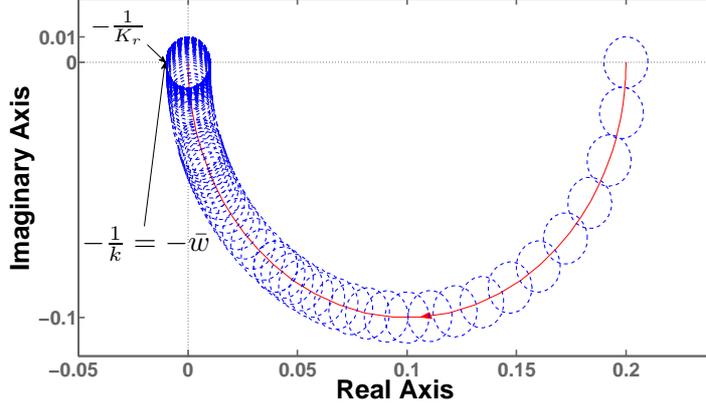


Figure 4.9: Robust Nyquist Plot.

In Fig. 4.9, the nominal plant is $G(s) = \frac{1}{s+5}$ and the uncertainty satisfies $\|\Delta\|_{\infty} = 1$. The value at K_r shown in Fig. 4.9, is the supremum of the robust interval \mathcal{I}_r . Moreover, by invoking Corollary 4.1, the obtained bound K_r satisfies :

$$\left|1 + K \frac{a}{j\omega + b}\right| > |K\bar{w}\Delta_{LTI}| \quad \forall \omega \in \mathbb{R}. \quad (4.15)$$

Remark 4.5. *An equivalent interpretation of (4.15) is that $G(j\omega)$ avoids and does not encircle the circle of radius $|\bar{w}\Delta_{LTI}|$ centered at $-1/K_r$ [82].*

The obtained K_r shown in Fig. 4.9 gives a robust interval $[0, K_r)$ that is the same as nonlinearity’s interval $\text{SR}[0, \frac{1}{\bar{w}})$. Thus, the analytical result of Result 4.1 corresponds with the graphical interpretation of the circle criterion for the “loose” region.

Table 4.2: RKC for Additive Uncertainty			
	Robust Interval (\mathcal{I}_r)	Multipliers :	$\frac{\varphi_k}{\text{Constant}}$ Slope Interval
$\bar{w} > 0$	$[0, 1/\bar{w})$		SR $[0, 1/\bar{w})$ -

4.5.2 INPUT-MULTIPLICATIVE UNCERTAINTY

Consider the well-posed uncertain nonlinear system with $\bar{w} \in \mathbb{R}_+$ and Δ satisfying $\|\Delta\|_{\mathcal{L}_2} \leq 1$ as shown in Fig. 4.10. Here, $G_\Delta = G(s) + G(s)\bar{w}\Delta$.

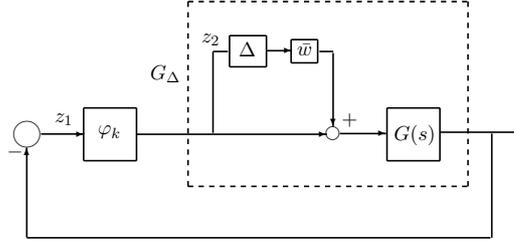


Figure 4.10: Robust Lur'e for Input-Multiplicative Uncertainty.

The generalized plant for the input-multiplicative uncertainty which is obtained by transforming the robust Lur'e problem in Fig. 4.10 into the augmented Lur'e problem in Fig. 4.6, is given by

$$\begin{bmatrix} -G(s) & -\bar{w}G(s) \\ 1 & 0 \end{bmatrix}. \quad (4.16)$$

Remark 4.6. *It is assumed that the interconnected system in Fig. 4.6 is well-posed. Then, due to the structure of $\tilde{\Delta}$, well-posedness for any $\tau \in [0, 1]$ is guaranteed.*

Corollary 4.3. *Consider the nonlinear system with input-multiplicative uncertainty as shown in Fig. 4.10. Then, the system is \mathcal{L}_2 -stable if there exist a stability multiplier $M(j\omega)$ such that*

$$kM(j\omega)G(j\omega) + kG(j\omega)^*M(j\omega)^* + M(j\omega) + M(j\omega)^* - 1 - \bar{w}^2k^2M(j\omega)G(j\omega)G(j\omega)^*M(j\omega)^* > 0 \quad (4.17)$$

for all $\omega \in \mathbb{R}$.

Proof. If there exists $M(j\omega)$ that satisfies (4.17) for all $\omega \in \mathbb{R}$, then applying Schur

complements, $M(j\omega)$ also satisfies

$$\begin{bmatrix} M(j\omega)(kG(j\omega) + I) + (kG(j\omega) + I)^*M(j\omega)^* - 1 & \bar{w}M(j\omega)G(j\omega)k \\ \bar{w}kG(j\omega)^*M(j\omega)^* & 1 \end{bmatrix} > 0 \quad (4.18)$$

for all $\omega \in \mathbb{R}$. Thus using $M(j\omega)$ in (4.6), the obtained $\Pi(j\omega)$ fulfills the conditions of Theorem 4.3 and therefore the feedback interconnection in Fig. 4.10 is \mathcal{L}_2 -stable. ■

Lemma 4.2. *Let $G_\Delta \in \mathcal{G}_\Delta$ be a first-order plant (4.7) with input-multiplicative uncertainty i.e. $G_\Delta = G[1 + \bar{w}\Delta]$ as shown in Fig. 4.10. The robust interval $\mathcal{I}_r [0, K_r)$ of G_Δ is:*

$$\mathcal{I}_r = \begin{cases} [0, \infty) & \text{if } \bar{w} \leq 1, \\ [0, \frac{b}{a(\bar{w}-1)}) & \text{if } \bar{w} > 1. \end{cases} \quad (4.19)$$

Proof. The result is obtained by applying Theorem 4.1. The system in 4.10 is stable if and only if,

$$\left\| \frac{\bar{w}Ka}{s + b + Ka} \right\|_{\mathcal{L}_2} \|\Delta\|_{\mathcal{L}_2} < 1. \quad (4.20)$$

It then follows that an equivalent condition is

$$\left\| \frac{\bar{w}Ka}{s + b + Ka} \right\|_{\mathcal{L}_2} = \left\| \frac{\bar{w}Ka}{s + b + Ka} \right\|_{\infty} < 1. \quad (4.21)$$

It follows from (4.21) that

$$(\bar{w} - 1)Ka \leq b. \quad (4.22)$$

Thus,

- If $\bar{w} \leq 1, K > 0$ and $K_r = \infty$.
- If $\bar{w} > 1, K < \frac{b}{a(\bar{w}-1)}$ and $K_r = \frac{b}{a(\bar{w}-1)}$. ■

Once we have obtained the maximum interval for which the nonlinear system could be stable, IQC machinery is then used to study the absolute stability of the robust Lur'e problem. The generalized plant in (4.16) is given by

$$G_p(s) = \begin{bmatrix} -\frac{a}{s+b} & -\bar{w}\frac{a}{s+b} \\ 1 & 0 \end{bmatrix}. \quad (4.23)$$

Result 4.2. *Let $G_\Delta \in \mathcal{G}_\Delta$ be a first-order plant (4.7) with input-multiplicative uncertainty as shown in Fig. 4.10. By virtue of Lemma 4.2, the robust interval (\mathcal{I}_r) of G_Δ is defined as (4.19). Thus the system in Fig. 4.10 is \mathcal{L}_2 -stable for any*

$$\varphi_k \in \begin{cases} \text{SR}[0, \infty) & \text{if } \bar{w} \leq 1, \\ \text{SR}[0, \frac{b}{a(\bar{w}-1)}) & \text{if } \bar{w} > 1. \end{cases} \quad (4.24)$$

Moreover, the stability multiplier $M(j\omega) = \frac{(ka+b)b^2}{k^2\bar{w}^2a^2} + j\omega\frac{(ka+b)b}{k^2\bar{w}^2a^2}$ is sufficient to establish this result.

Proof. It is shown that there exists an admissible Popov multiplier $M(j\omega) = \eta + j\omega\lambda$ with $\eta > 0$, $\lambda \in \mathbb{R}$ such that (4.17) in Corollary 4.3 is satisfied.

Set $M(j\omega) = \eta + j\omega\lambda$, substitute $G(j\omega)$ and let $x_\omega = b^2 + \omega^2$, then the stability condition (4.17) reduces to

$$\frac{2\eta kab}{x_\omega} + \frac{2\omega^2\lambda ka}{x_\omega} + 2\eta - 1 - k^2\bar{w}^2(\eta^2 + \omega^2\lambda^2)\frac{a^2}{x_\omega} > 0 \quad \forall \omega \in \mathbb{R}. \quad (4.25)$$

Choose $\eta = \lambda b$ with $\lambda > 0$, it then follows that an equivalent stability condition is

$$2\lambda ka + 2\lambda b - 1 - k^2\bar{w}^2a^2\lambda^2 > 0. \quad (4.26)$$

Choose $\lambda = \frac{kab+b^2}{k^2\bar{w}^2a^2}$ in (4.26) and the stability condition (4.26) reduces to

$$b^2 + 2kab + k^2a^2(1 - \bar{w}^2) > 0. \quad (4.27)$$

Thus,

- If $\bar{w} \leq 1, k > 0$.
- If $\bar{w} > 1, k < \frac{b}{a(\bar{w}-1)}$. ■

Remark 4.7. *Result 4.2 demonstrates that the robust Kalman conjecture is true for first-order plant with input-multiplicative uncertainty $\forall \bar{w} > 0$. As indicated in Table 4.3, the robust Popov criterion provides an appropriate Popov multiplier for this purpose. However, it can be shown that the robust circle criterion is only adequate when $\bar{w} \geq 2$ (see Appendix A.1).*

Graphical Interpretation of Result 4.2

Given the family of uncertain plants as $G_{\Delta_{LTI}}(s) = G(s) + \bar{w}G(s)\Delta_{LTI}$, then the “loose” region is defined by discs of radius $\bar{w}|G(j\omega)|$ centered at each frequency. The robust Nyquist and the corresponding robust Popov plots having ellipses with semi-axes of $\bar{w}|G(j\omega)|$ and $\omega\bar{w}|G(j\omega)|$ centered at each frequency are illustrated in Fig. 4.11 for $\bar{w}=1.3$.

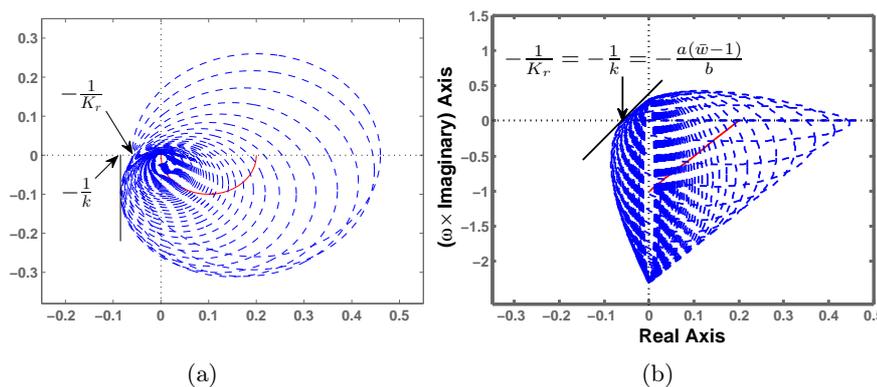


Figure 4.11: (a) Robust Nyquist and (b) Robust Popov Plots.

In Fig. 4.11, the nominal plant is $G(s) = \frac{1}{s+5}$ and the uncertainty satisfies $\|\Delta\|_\infty = 1$. The value at K_r shown in Fig. 4.11, is the supremum of the robust interval \mathcal{I}_r . Moreover,

by invoking Corollary 4.1, the bound K_r satisfies :

$$|1 + K \frac{a}{j\omega + b}| > |K\bar{w} \frac{a}{j\omega + b} \Delta_{LTI}| \quad \forall \omega \in \mathbb{R}. \quad (4.28)$$

Remark 4.8. *An equivalent interpretation of (4.28) is that $G_m = \frac{G(j\omega) + K^{-1}}{G(j\omega)} - K^{-1}$ avoids and does not encircle the circle of radius $|\bar{w}\Delta_{LTI}|$ centered at $-1/K_r$ [77].*

It can be observed from Fig. 4.11(a) that $-\frac{1}{K_r} \neq -\frac{1}{k}$ and therefore the robust Circle criterion is not enough to establish the veracity of the RKC. However, the obtained K_r from the Popov plot shown in Fig. 4.11(b) gives a robust interval $[0, K_r)$ that is the same as nonlinearity's interval *i.e.* $-\frac{1}{K_r} = -\frac{1}{k}$. Thus, the analytical result of Result 4.2 corresponds with the graphical interpretation of the Popov criterion for the “loose region”.

Table 4.3: RKC for Input-Multiplicative Uncertainty

	Robust Interval (\mathcal{I}_r)	φ_k Slope Interval	
		Constant Multi.	Popov Multi.
$\bar{w} \leq 1$	$[0, \infty)$	SR $[0, \frac{4b}{a\bar{w}^2})$	SR $[0, \infty)$
$1 < \bar{w} < 2$	$[0, \frac{b}{a(\bar{w}-1)})$	SR $[0, \frac{4b}{a\bar{w}^2})$	SR $[0, \frac{b}{a(\bar{w}-1)})$
$\bar{w} \geq 2$	$[0, \frac{b}{a(\bar{w}-1)})$	SR $[0, \frac{b}{a(\bar{w}-1)})$	SR $[0, \frac{b}{a(\bar{w}-1)})$

4.5.3 FEEDBACK UNCERTAINTY

Consider the well-posed uncertain nonlinear system with $\bar{w} \in \mathbb{R}_+$ and Δ satisfying $\|\Delta\|_{\mathcal{L}_2} \leq 1$ as shown in Fig. 4.12. Here, $G_\Delta = G(s)[1 + \bar{w}\Delta G(s)]^{-1}$.

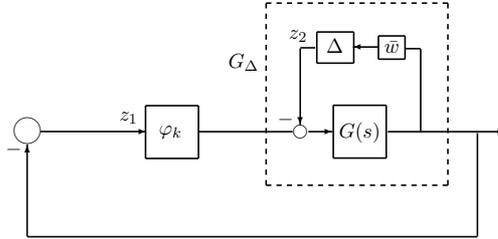


Figure 4.12: Robust Lur'e for Feedback Uncertainty.

The generalized plant for the feedback uncertainty which is obtained by transforming

the robust Lur'e problem in Fig. 4.12 into the augmented Lur'e problem in Fig. 4.6, is given by

$$\begin{bmatrix} -G(s) & \bar{w}G(s) \\ G(s) & -\bar{w}G(s) \end{bmatrix}. \quad (4.29)$$

Remark 4.9. *It is assumed that the interconnected system in Fig. 4.6 is well-posed. Then, due to the structure of $\tilde{\Delta}$, well-posedness for any $\tau \in [0, 1]$ is guaranteed.*

Corollary 4.4. *Consider the nonlinear system with feedback uncertainty as shown in Fig. 4.12. Let $\mathcal{T} = [\bar{w}G(j\omega)^*G(j\omega) - \bar{w}W(j\omega)kG(j\omega)]$, then the system is \mathcal{L}_2 -stable if there exist a stability multiplier $W(j\omega)$ such that $[1 - \bar{w}^2G(j\omega)^*G(j\omega)]^{-1} > 0$ and*

$$kM(j\omega)G(j\omega) + kG(j\omega)^*M(j\omega)^* + M(j\omega) + M(j\omega)^* - \\ G(j\omega)^*G(j\omega) - \mathcal{T}[1 - \bar{w}^2G(j\omega)^*G(j\omega)]^{-1}\mathcal{T}^* > 0 \quad (4.30)$$

for all $\omega \in \mathbb{R}$.

Proof. If there exists $M(j\omega)$ that satisfies (4.30) for all $\omega \in \mathbb{R}$, then applying Schur complements, $M(j\omega)$ also satisfies

$$\begin{bmatrix} M(j\omega)(kG(j\omega)+I)+(kG(j\omega)+I)^*M(j\omega)^*-G(j\omega)^*G(j\omega) & \bar{w}G(j\omega)^*G(j\omega)-\bar{w}M(j\omega)kG(j\omega) \\ \bar{w}G(j\omega)^*G(j\omega)-\bar{w}kG(j\omega)^*M(j\omega)^* & 1-\bar{w}^2G(j\omega)^*G(j\omega) \end{bmatrix} \\ > 0 \quad \forall \omega \in \mathbb{R}. \quad (4.31)$$

Thus using $M(j\omega)$ in (4.6), the obtained $\Pi(j\omega)$ fulfills the conditions of Theorem 4.3 and therefore the feedback interconnection in Fig. 4.12 is \mathcal{L}_2 -stable. \blacksquare

Remark 4.10. *The uncertain plant G_Δ is stable if $\bar{w} < \frac{b}{a}$.*

Lemma 4.3. *Let $G_\Delta \in \mathcal{G}_\Delta$ be a first-order plant (4.7) with an feedback uncertainty i.e. $G_\Delta = G[1 + \Delta G\bar{w}]^{-1}$ as shown in Fig. 4.12. For $\bar{w} < \frac{b}{a}$ the robust interval $\mathcal{I}_r [0, K_r)$ of G_Δ is $[0, \infty)$.*

Proof. The result is obtained by applying Theorem 4.1. The system in 4.12 is stable if and only if,

$$\left\| \frac{\bar{w}a}{s + b + Ka} \right\|_{\mathcal{L}_2} \|\Delta\|_{\mathcal{L}_2} < 1. \quad (4.32)$$

It then follows that an equivalent condition is

$$\left\| \frac{\bar{w}a}{s + b + Ka} \right\|_{\mathcal{L}_2} = \left\| \frac{\bar{w}a}{s + b + Ka} \right\|_{\infty} < 1. \quad (4.33)$$

It follows from (4.33) that

$$K > \bar{w} - \frac{b}{a}. \quad (4.34)$$

Thus, if $\bar{w} \leq \frac{b}{a}$, $K > 0$ and $K_r = \infty$. ■

Once we have obtained the maximum interval for which the nonlinear system could be stable, IQC machinery is then used to study the absolute stability of the robust Lur'e problem. The generalized plant in (4.29) is given by

$$G_p(s) = \begin{bmatrix} -\frac{a}{s+b} & -\bar{w} \\ \frac{a}{s+b} & 0 \end{bmatrix}. \quad (4.35)$$

Result 4.3. *Let $G_\Delta \in \mathcal{G}_\Delta$ be a first-order plant (4.7) with feedback uncertainty as shown in Fig. 4.12. By virtue of Lemma 4.3, for $\bar{w} < \frac{b}{a}$ the robust interval (\mathcal{I}_r) of G_Δ is defined as $[0, \infty)$. Thus, for $\bar{w} < \frac{b}{a}$ the system in Fig. 4.12 is \mathcal{L}_2 -stable for any $\varphi_k \in \text{SR}[0, \infty)$. Moreover, the stability multiplier $M(j\omega) = \frac{(ka+b)b - \bar{w}^2 a^2}{k^2 \bar{w}^2 a^2}$ is sufficient to establish this result.*

Proof. It is shown that there exists an admissible constant multiplier $M(j\omega) = \eta$ with $\eta > 0$ such that (4.30) in Corollary 4.4 is satisfied.

Set $M(j\omega) = \eta$, substitute $G(j\omega)$ and let $x_\omega = b^2 + \omega^2$, then for all $\omega \in \mathbb{R}$ the stability condition (4.30) reduces to

$$\frac{2\eta kab}{x_\omega} + 2\eta - \frac{a^2}{x_\omega} - \frac{x_\omega}{x_\omega - a^2\bar{w}^2} \left[\frac{\bar{w}^2 a^4}{x_\omega^2} - \frac{2\bar{w}^2 \eta k a^3 b}{x_\omega^2} + \frac{\bar{w}^2 \eta^2 k^2 a^2}{x_\omega} \right] > 0. \quad (4.36)$$

Factorizing and collecting like terms together, stability condition (4.36) is equivalent to

$$\frac{2\eta kab}{x_\omega} + 2\eta - \frac{a^2}{x_\omega} \left[\frac{x_\omega - 2\eta kab \bar{w}^2}{x_\omega - a^2 \bar{w}^2} \right] - \frac{\eta^2 k^2 \bar{w}^2 a^2}{x_\omega - a^2 \bar{w}^2} > 0. \quad (4.37)$$

Now, since $x_\omega \geq b^2$, it then follows that the stability condition (4.37) reduces to

$$2\eta kab + 2\eta(b^2 - \bar{w}^2 a^2) - \eta^2 k^2 \bar{w}^2 a^2 > 0. \quad (4.38)$$

Let $\tau = b^2 - \bar{w}^2 a^2$ and choose $\eta = \frac{kab + \tau}{k^2 \bar{w}^2 a^2}$, then the stability condition (4.38) is reduced to

$$k^2 a^2 (b^2 - \bar{w}^2 a^2) + 2kab\tau + \tau^2 > 0. \quad (4.39)$$

Thus, for $b < \frac{\bar{w}}{a}$, we have $k > 0$. ■

Remark 4.11. *Result 4.3 demonstrates that the robust Kalman conjecture is true for first-order plant with feedback uncertainty. As indicated in Table 4.4, the robust circle criterion provides an appropriate constant multiplier for this purpose.*

Graphical Interpretation of Result 4.3

In this case, the “loose” region is defined by discs of $G(j\omega)[1 + \bar{w}\Delta_{LTI}G(j\omega)]^{-1}$ as shown in Fig. 4.13 for $\bar{w} = 3$.

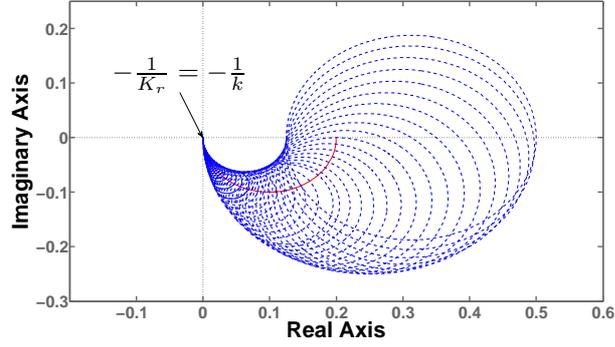


Figure 4.13: Robust Nyquist Plot.

In Fig. 4.13, the nominal plant is $G(s) = \frac{1}{s+5}$ and the uncertainty satisfies $\|\Delta\|_\infty = 1$. The value at K_r shown in Fig. 4.13, is the supremum of the robust interval \mathcal{I}_r . Moreover, by invoking Corollary 4.1, the obtained bound K_r satisfies :

$$\left| \frac{s+b}{a} + K \right| > |\bar{w}\Delta_{LTI}| \quad \forall \omega \in \mathbb{R}. \quad (4.40)$$

Remark 4.12. An equivalent interpretation of (4.40) is that $G_m = \frac{G(j\omega)+K^{-1}}{K^{-1}G(j\omega)} - K^{-1}$ avoids and does not encircle the circle of radius $|\bar{w}\Delta_{LTI}|$ centered at $-1/K_r$.

The obtained K_r shown in Fig. 4.13 gives a robust interval $[0, \infty)$ that is the same as nonlinearity's interval $\text{SR}[0, \infty)$. Thus, the analytical result of Result 4.3 corresponds with the graphical interpretation of the circle criterion for the “loose” region.

Table 4.4: RKC for Feedback Uncertainty			
	Robust Interval	φ_k Multipliers :	Slope Interval
	(\mathcal{I}_r)	Constant	-
$\bar{w} < \frac{b}{a}$	$[0, \infty)$	SR	$[0, \infty)$

4.6 Conclusion

This chapter has presented a new robust absolute stability conjecture called the robust Kalman conjecture. As indicated in Table 4.5, with the aid of different stability multipliers, it has been shown that this robust Kalman conjecture is true for first-order stable

plants with additive, input-multiplicative and feedback uncertainties. A graphical interpretation of the result is given for each case when the norm-bound perturbations includes LTI uncertainties. The practical significance of this conjecture, albeit for higher -order systems, is highlighted by recent considerations of robust preservation in anti-windup control systems [24, 83, 84]. Thus, in general, robust control of Lur'e-type nonlinear systems satisfying this novel conjecture can therefore be designed using linear robust control methods.

Table 4.5: Robust Kalman Conjecture For First-Order Plants

	Uncertainty Descriptions		
	Additive	Input-Multiplicative	Feedback
Sufficient Multiplier	Constant	Popov	Constant

Summary of PART I

The concept of stability in the sense of Lyapunov and input-output properties has been discussed in Chapter 2. In particular, the discrete Lyapunov theorem, the IQC theorem and the Schur Complement Lemma discussed in this chapter are key mathematical tools required for the contributions presented in Chapter 4 and Chapter 8 respectively. The issue of robust preservation in anti-windup schemes is briefly highlighted in Chapter 3 and it is also concluded therein that most notable anti-windup schemes could be transformed into an equivalent Lur'e structure. In Chapter 4, a fairly general and abstract framework (based on the Lur'e structure) known as the "Robust Kalman Conjecture" is presented to study the issue of this robust preservation for first-order SISO plants. It is shown that this robust Kalman conjecture is true for first-order stable plants with additive, input-multiplicative and feedback uncertainties. The practical significance of this conjecture, albeit for higher-order systems, is highlighted by recent considerations of robust preservation in anti-windup control systems. Thus, in general, robust control of Lur'e-type nonlinear systems satisfying this novel conjecture can be designed using linear robust control methods.

PART II:
Optimization in Anti-windup Control

Chapter 5

Introductory Background on Optimization

Online-optimizing controllers such as MPC and the two-stage IMC anti-windup [11] require some sort of optimization routine. A number of optimization approaches have been reported in literature for MPC applications. These approaches can be easily applied to other classes of emerging online-optimizing anti-windup (e.g. [11]). The suitability of these approaches for high-speed applications depends on factors such as (a) ability to meet hard real-time constraints on the solution-time (b) tight practical complexity certificates (c) Ease of implementation. At this point, the various approaches are briefly discussed, with emphasis on their suitability or otherwise for online-optimizing anti-windup. These approaches as reported in the literature generally fall into three main categories as follows:

1. One approach is multi-parametric programming, which allows one to pre-compute the solution for every state offline [94, 95]. The explicit solution is a piece-wise affine map over a polyhedral partition of the state-space and can be stored efficiently such that fast online look-up is ensured. However, as the control action implemented online is in the form of a lookup table, the explicit MPC cannot deal with applications whose dynamics, cost function and/or constraints are

time-varying problems [96]. Moreover, the number of entries in the table can grow exponentially with the size of the optimization problem limiting the applicability of explicit MPC to very small problems. Recent work on reducing the complexity of explicit MPC while maintaining stability and feasibility of the obtained control law can be found in [97–99].

2. Online optimization methods are generally used for large-scale problems, the two main proponents being interior point [100,101] and active set methods [102]. The authors of [96,103] report a fast implementation of an interior point method where a significant speed-up is gained by exploiting the structure of the involved matrices as well as by early stopping and warm-starting from a solution obtained at the previous time-step. Successful implementation of both active-set and interior-point methods to fast real-time application can be found in [104]. One of the main issues with the aforementioned online optimization methods is the inability to guarantee ϵ -suboptimality within the hard real-time constraints (e.g. sampling interval) of a given application. Such certificates can be obtained for interior point methods but the computed bounds are far off from the practically observed ones [105].
3. Recently, there has been significant interest in using first-order methods [20,106,107] for the online optimization of linear-quadratic MPC problems. Compared to other solution methods for quadratic programs, first-order methods do not require the solution of a linear system of equations at every iteration, which is often a limiting factor for embedded platforms with modest computational capability. This feature, coupled with the observation that medium-accuracy solutions are often sufficient for good control performance [96], make first-order methods promising candidates for efficient online optimization. In particular, complexity certificates for fast gradient methods reported in [21,108] allows one to derive practically less-conservative bounds on the computational effort.

More often than not, we are interested in the optimization of convex formulation of the anti-windup problem. Thus, this chapter subsequently discusses a special category

of optimization problems called convex optimization. Section 5.2 explores the unconstrained case while Section 5.3 looks at the constrained counterpart.

5.1 What is Convex Optimization?

A set \mathcal{X} is convex if for any 2 points $x_0, x_1 \in \mathcal{X}$ we have that $\lambda x_1 + (1 - \lambda)x_0 \in \mathcal{X}$ for all $0 \leq \lambda \leq 1$. An objective function $f : \mathcal{X} \rightarrow \mathbb{R}$ defined on a set \mathcal{X} is convex if and only if the epigraph of f is a convex set. Furthermore, the epigraph of f is a convex set if and only if it satisfies Jensen's inequality,

$$f(\lambda x_1 + (1 - \lambda)x_0) \leq \lambda f(x_1) + (1 - \lambda)f(x_0); \quad \forall x_0, x_1 \in \mathcal{X}, \quad \forall 0 \leq \lambda \leq 1.$$

Convex optimization is a class of optimization wherein the objective function is well-defined, convex and its domain also defines a feasible convex set.

5.2 Unconstrained Optimization

Consider the following unconstrained optimization of a convex function f :

$$\text{UOP} : \quad \min_{x \in \mathbb{R}^n} f(x), \tag{5.1}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous differentiable convex function and $x \in \mathbb{R}^n$ is a real vector. If the domain of f , $\mathbf{dom} f$, is a convex set (e.g. \mathbb{R}^n), then UOP is a convex optimization problem. In this case, a necessary and sufficient condition for point x^* to be a minimizer of f is $\nabla f(x^*) = 0$ [85]. Furthermore, if f is bounded below, then there exists a unique $x^* : f(x^*) = f^*$ with $f^* \leq f(x) \quad \forall x \in \mathbb{R}^n$ [86].

In a few special cases, an analytic solution of (5.1) can be obtained but usually the problem must be solved by an iterative algorithm. An iterative algorithm computes a sequence of iterates $x_0, x_1, \dots \in \mathbf{dom} f$ with $f(x_k) \rightarrow f^*$ as $k \rightarrow \infty$. Such a sequence of points is called a minimizing sequence for the problem (5.1) if $f(x_{k+1}) < f(x_k), \forall k > 0$.

The algorithm is terminated when $f(x_k) - f^* \leq \epsilon$, where $\epsilon > 0$ is some specified tolerance. Iterative algorithms are of the form

$$x_{k+1} = x_k + \alpha_k d_k, \quad (5.2)$$

where α_k , d_k denotes the step length and the search direction at the k^{th} iteration respectively. In this section, the general descent method for unconstrained optimization is discussed. Other types of algorithms also applicable to unconstrained optimization are explored under constrained optimization.

5.2.1 Descent Methods

Monotone iterative algorithms require that $f(x_{k+1}) < f(x_k)$ for all $k \geq 0$ and they are also known as descent methods [86].

Algorithm 5.1 ([86]). The outline of a general descent method is as follows:

Algorithm 5.1 Descent method.

Given a starting point $x_0 \in \mathbf{dom} f$.

repeat until stopping criterion is satisfied

1. Descent direction: Compute d_k .
2. Line-Search: Choose a step-size $\alpha_k > 0$.
3. Update: $x_{k+1} = x_k + \alpha_k d_k$.

end (repeat)

Descent Direction. The search direction in a descent method must satisfy $\nabla f(x_k)^T d_k < 0$ i.e. the descent direction must make an acute angle θ_k with the negative gradient. $\nabla f(x_k)^T d_k$ is known as the directional derivative of $f(x)$ at x_k in the direction

of d_k . The directional derivative gives the decrease in f along the direction of d_k . A natural choice for the search direction is the unit-norm negative gradient,

$$d_k = -\nabla f(x_k) / \|\nabla f(x_k)\|,$$

which minimizes the first-order Taylor approximation of $f(x+d)$ at x_k subject to $\|d\| \leq 1$. The resulting algorithms are called gradient-descent methods if $f(x_{k+1}) < f(x_k)$ or simply called gradient methods if otherwise. Provided the Hessian $\nabla^2 f(x) > 0$, another good choice is the Newton direction $d_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$ which minimizes the second-order Taylor approximation of $f(x+d)$ at x . Similarly, this choice results in the Newton-descent methods if $f(x_{k+1}) < f(x_k)$ or simply Newton methods if otherwise.

Furthermore, any descent direction d_k must satisfy the Zoutendijk condition [86] i.e.

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|\nabla f(x_k)\|^2 < \infty, \quad \text{where} \quad \cos \theta_k = \frac{-\nabla f(x_k)^T d_k}{\|\nabla f(x_k)\| \|d_k\|}. \quad (5.3)$$

The Zoutendijk condition dictates how far the descent direction d_k is allowed to deviate away from the negative gradient direction $-\nabla f(x_k)$.

Line-Search. The descent condition $f(x_{k+1}) < f(x_k)$ is not sufficient to guarantee that iterates converge to a minimizer of f . It is, however, sufficient to select a suitable step-size α_k such that $f(x_{k+1})$ is minimized along the descent direction d_k . The computation of α_k is a line-search whose solution can be obtained either by:

- 1.) Exact line-search computes the step-length α_k^* that globally minimizes

$$\phi(\alpha) = f(x_k + \alpha d_k), \quad \alpha > 0. \quad (5.4)$$

The step-size obtained by the exact line-search is called the exact step-size or Cauchy step-size. The corresponding descent method is known as the classical steepest descent (SD) method i.e. $\alpha_k^{SD} = \alpha_k^*$. In particular, for the quadratic convex functions, the

exact step-length can be computed as

$$\alpha_k^{SD} = \alpha_k^* = \frac{d_k^T d_k}{d_k^T H d_k}, \quad (5.5)$$

where H is the Hessian matrix. However, in general an exact line-search is impossible or too expensive. For all convex $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$, it can be shown that the fixed step-length $\alpha = 1/L$ guarantees a decrease that results in optimal convergence rate for this class of functions [23].

2.) Inexact line-search attempts to identify a step-length that achieves adequate reductions in f at minimal cost. One of the well-known and useful inexact line-search was proposed by Armijo (see [86]). The Armijo condition ensures that α_k should give a sufficient decrease in the objective function as measured by the inequality:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \alpha_k c_1 \nabla f(x_k)^T d_k. \quad (5.6)$$

Using the so-called backtracking approach to enforce (5.6) rules out unacceptable short step-sizes. Alternatively, the Goldstein conditions and the Wolfe conditions (see [86]) impose an additional condition to the Armijo condition in order to ensure that the step-lengths α_k are not too short.

Algorithm 5.2 ([86]). The outline of the backtracking line-search is as follows:

Algorithm 5.2 (Backtracking Line-Search).

Given $\bar{\alpha} > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$. Set $\alpha \leftarrow \bar{\alpha}$.

repeat until $f(x_k + \alpha \Delta x_k) \leq f(x_k) + \alpha c \nabla f(x_k)^T \Delta x_k$

1. Set $\alpha \leftarrow \rho \alpha$.

end (repeat)

Terminate with $\alpha_k = \alpha$.

Performing back-tracking ensures that the inexact line-search terminates finitely [87].

5.3 Constrained Optimization

The mathematical formulation for constrained optimization problem is expressed as follows:

$$\text{COP : } \quad \min_{x \in \mathbb{R}^n} f(x) \quad (5.7a)$$

$$\text{s.t } \quad g_i(x) = 0, \quad \text{for } i = 1, \dots, r, \quad (5.7b)$$

$$h_i(x) \leq 0, \quad \text{for } i = 1, \dots, m, \quad (5.7c)$$

where $x \in \mathbb{R}^n$ is a real vector and f, h_i, g_i are continuous functions. Moreover, the constrained optimization problem (COP) is convex if the objective function f is convex, constraint function h_i is convex and constraint function g_i is affine. Examples include linear programming and convex quadratic programming. Briefly, linear programming is a constrained optimization problem wherein both the objective function and constraints are linear, while the convex quadratic programming refers to the case involving a quadratic objective function and linear constraints. The simplex algorithm, popularized by Dantzig in 1947 (see [88]), is a very efficient method for solving linear programming. However, Karmarkar [89] introduced in 1984 a new class of methods called interior-point methods. Most of the ideas underlying this new class of methods originate from the nonlinear optimization domain. Alternative methods include the active-set methods and variants of projected gradient methods. These methods are both theoretically and practically efficient and can be generalized to other types of constrained optimization problems specifically convex quadratic programming.

Optimality Conditions

The KKT conditions generalizes the classical Langrange multiplier method to include inequality constraints. Karush-Kuhn-Tucker (KKT) conditions are necessary optimality conditions pertaining to nonlinear constrained optimization with a differentiable objective. The COP has a primal optimal (x^*) and dual optimal (β^*, λ^*) with zero duality gap if the points x^*, β^*, λ^* satisfy the KKT conditions [85, pg. 243]:

$$\text{Dual Feasibility} \quad \nabla f_i(x^*) + \sum_{i=1}^r \beta_i^* \nabla g_i(x^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(x^*) = 0, \quad (5.8a)$$

$$\text{Primal Feasibility} \quad \begin{cases} g_i(x^*) = 0 & i = 1 \cdots r, \\ h_i(x^*) \leq 0 & i = 1 \cdots m, \end{cases} \quad (5.8b)$$

$$\text{Dual Feasibility} \quad \begin{cases} \lambda_i^* \geq 0 & i = 1 \cdots m, \\ \beta_i^* \in \mathbb{R} & i = 1 \cdots r, \end{cases} \quad (5.8c)$$

$$\text{Complementary Slackness} \quad \begin{cases} \lambda_i^* h_i(x^*) = 0 & i = 1 \cdots m. \end{cases} \quad (5.8d)$$

Moreover, the KKT conditions are also sufficient for "optimality with zero duality-gap" when the COP is convex (i.e f, h_i are convex and g_i is affine) [85, pg. 244], which is the case for linear programming and convex quadratic programming. Subsequently a special case of the convex COP is considered.

$$\min_{x \in \mathbb{R}^n} \quad \frac{1}{2} x^T H x + c^T x \quad (5.9a)$$

$$\text{s.t.} \quad A x \leq b, \quad \text{for } i = 1, \dots, m, \quad (5.9b)$$

where $H \geq 0$ and $A \in \mathbb{R}^{m \times n}$ with full row rank. The COP (5.9) is known as convex quadratic programming. In this case, the KKT conditions for (5.9) reduce to

$$\text{Dual Feasibility} \quad H x + c + A^T \lambda = 0, \quad (5.10a)$$

$$\text{Primal Feasibility} \quad \begin{cases} A x - b \leq 0, \end{cases} \quad (5.10b)$$

$$\text{Dual Feasibility} \quad \begin{cases} \lambda_i \geq 0 & i = 1 \cdots m, \end{cases} \quad (5.10c)$$

$$\text{Complementary Slackness} \quad \begin{cases} \lambda_i (a_i^T x - b_i) = 0 & i = 1 \cdots m, \end{cases} \quad (5.10d)$$

where a_i^T is the i^{th} row of matrix A and b_i is the i^{th} row of column vector b .

5.3.1 Active-Set Method

Primal active-set methods generate iterates that remain feasible with respect to the primal problem while steadily decreasing the objective function $f(x)$. The active set $\mathcal{A}(x^*)$ consists of the indices of the constraints for which equality holds at x^* : $\mathcal{A}(x^*) = \{i \mid a_i^T x = b_i\}$. This method finds a step from one iterate to the next by solving a quadratic sub-problem on a subset defined by the current estimate of the active set $\mathcal{A}(x^*)$. This subset is referred to as the working-set and is denoted at the k^{th} iterate by \mathcal{W}_k .

Algorithm 5.3 ([86]). The outline of a basic active-set method is as follows:

Algorithm 5.3 Basic active-set algorithm.

Given a feasible point x_0 and a working-set \mathcal{W}_0 .¹

repeat until stopping criterion is satisfied

1. Determine if x_k minimizes (5.9a) subject to the constraints in \mathcal{W}_k . This is done by reformulating as follows. Let $\Delta x_k = x - x_k$. By substituting for x into $f(x)$ and ignoring constant terms, then sub-QP problem can be expressed as

$$\begin{cases} \min_{\Delta x_k} & \frac{1}{2} \Delta x_k^T H \Delta x_k + z_k^T \Delta x_k \\ & a_i^T \Delta x_k = b_i - a_i^T x_k = 0, \quad i \in \mathcal{W}_k. \end{cases} \quad (5.11)$$

The KKT conditions of this sub-QP problem (5.11) is solved for $\Delta x_k, \lambda_k$:

$$\begin{bmatrix} H & A_w^T \\ A_w & 0 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \lambda_k \end{bmatrix} = \begin{bmatrix} -z_k \\ 0 \end{bmatrix}, \quad (5.12)$$

where $z_k = Hx_k + c$ and A_w is the matrix of $a_i \in \mathcal{W}_k$. If $\Delta x_k = 0$, go to step 3, otherwise continue with the next step.

¹ \mathcal{W}_0 comprises all the equality constraints (none in this case) and some of the inequality constraints imposed as equalities

2. If $\Delta x_k \neq 0$, then the current feasible set x_k does not minimize $f(x)$ with respect to the working-set \mathcal{W}_k . Then Δx_k is used as a search direction and the next feasible point is found as $x_{k+1} = x_k + (\alpha_k \Delta x_k)$. An explicit definition of step-size parameter α_k can be derived by considering only what happens to the constraints $i \notin \mathcal{W}_k$, since the constraints $i \in \mathcal{W}_k$ will certainly be satisfied regardless of the choice of α_k (see [86]).

$$\alpha_k = \min_{a_i \Delta x_k > 0} \left\{ 1, \frac{b_i - a_i x_k}{a_i \Delta x_k} \right\} \quad i \notin \mathcal{W}_k. \quad (5.13)$$

For $\alpha_k < 1$, the vector(s) of $a_i \notin \mathcal{W}_k$ corresponding to the blocking constraints is adjoined to \mathcal{W}_k to form a next working-set. $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k + A_{nw}$, where A_{nw} is the matrix of $a_i \notin \mathcal{W}_k$.

3. Here, $f(x_k)$ is optimal with respect to the working-set \mathcal{W}_k . Examine KKT multipliers λ_k of (5.12).
- (a) If $\lambda_k \geq 0$, then optimal solution obtained.
 - (b) If $\lambda_{i,k} < 0$, then the constraint with the most-negative λ_k is removed from the working set so as to decrease the cost further while remaining in the feasible space. $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k - A_w$.

end (repeat)

Updating of both $\{a_i, b_i\} \in \mathcal{W}_k$ and $\{a_i, b_i\} \notin \mathcal{W}_k$ is done at each iteration. It is noted that when the working-set \mathcal{W}_k is empty, the next iteration starts by solving an unconstrained sub-QP problem which yields $\Delta x_k = -H^{-1}z_k$ and the corresponding KKT multipliers are set to 0.

5.3.2 Interior-Point Method

The interior-point approach has proved to be an attractive alternative when the problems are large and convex. In addition, this approach has the advantage that the system

of linear equations to be solved at each iterate has the same dimension and structure throughout the algorithm, making it possible to exploit any structure inherent in the problem.

Iterate Space:

Primal-Dual interior point methods compute a sequence of strictly feasible primal-dual iterates by applying Newton's method to solve the KKT conditions, modifying the search directions and step-size so that the iterates remain strictly feasible. Introduce dual slack variable s in constraints (5.10) and the KKT conditions become:

$$\text{Dual Feasibility} \quad Hx + c + A^T \lambda = 0, \quad (5.14a)$$

$$\text{Primal Feasibility} \quad \begin{cases} Ax - b + s = 0, \\ s_i \geq 0, \end{cases} \quad (5.14b)$$

$$\text{Dual Feasibility} \quad \begin{cases} \lambda_i \geq 0 & i = 1 \dots m, \end{cases} \quad (5.14c)$$

$$\text{Complementary Slackness} \quad \begin{cases} \lambda_i s_i = 0 & i = 1 \dots m, \end{cases} \quad (5.14d)$$

Using the standard form notation, we define the feasible set \mathcal{F} to be the set satisfying

$$\mathcal{F} = \{(x, \lambda, s) \mid Hx + c + A^T \lambda = 0, Ax - b + s = 0, s \geq 0, \lambda \geq 0\} \quad (5.15)$$

and the associated strictly feasible set \mathcal{F}^+ to be the subset of \mathcal{F} satisfying

$$\mathcal{F}^+ = \{(x, \lambda, s) \mid Hx + c + A^T \lambda = 0, Ax - b + s = 0, s > 0, \lambda > 0\}. \quad (5.16)$$

The convex quadratic program (5.9) can be solved by finding solutions of the set of equations (5.14).

Central Path-Following:

Given a current iterate $(x, s, \lambda) \in \mathcal{F}^+$ that satisfies $(s > 0, \lambda > 0)$. Define a comple-

mentarity measure

$$\nu = \frac{\lambda^T s}{m}$$

and define $F(x, s, \lambda, \sigma\nu)$

$$F(x, s, \lambda, \sigma\mu) = \begin{bmatrix} Hx + c + A^T \lambda \\ Ax - b + s \\ \Lambda S e - \sigma\mu e \end{bmatrix}, \quad \lambda, s \geq 0, \quad (5.17)$$

where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$, $S = \text{diag}(s_1, s_2, \dots, s_m)$, $e = (1, \dots, 1)^T$ and $\sigma \in [0, 1]$.

The central path is defined as the set of points (x_k, λ_k, s_k) such that

$$F(x_k, s_k, \lambda_k, \sigma\mu) = 0. \quad (5.18)$$

The solutions of the perturbed KKT conditions (5.18) for all values of $\sigma \in [0, 1]$ and $\nu \geq 0$, define the central path, which is a trajectory that leads to the solution of the original KKT conditions (5.14) as ν tends to zero. A step with $\sigma = 1$ is referred to as a centering step and a step with $\sigma = 0$ is referred to as an affine-scaling step. The choice of the centering parameter $\sigma \in [0, 1]$ provides a trade-off between moving towards the central path and moving towards the optimal solution of the optimization problem.

Descent Direction:

The perturbed KKT conditions (5.18) are now solved using first-order Newton's method to obtain the descent direction. Given strictly feasible iterate $(x_k, \lambda_k, s_k) \in \mathcal{F}^+$, the Newton's step is then defined by the linear system of equations,

$$\begin{bmatrix} H & A & 0 \\ A^T & 0 & I \\ 0 & S_k & \Lambda_k \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta \lambda_k \\ \Delta s_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \Lambda_k S_k e - \sigma\mu_k e \end{bmatrix}, \quad (5.19)$$

where $\Delta x_k, \Delta \lambda_k, \Delta s_k$ are the descent directions. The descent directions can now be obtained by Cholesky factorization and two back-solves.

Step-Size

Here, the next strictly feasible iterate is obtained with

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \\ s_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \\ s_k \end{bmatrix} + \alpha_k \begin{bmatrix} \Delta x_k \\ \Delta \lambda_k \\ \Delta s_k \end{bmatrix}, \quad (5.20)$$

where $\alpha_k \in [0, 1]$ is computed such that the inequality $\lambda_{k+1} > 0, s_{k+1} > 0$ is retained. As such all subsequent iterates also belong \mathcal{F}^+ .

Algorithm 5.4 ([86]). The outline of a basic interior-point algorithm is as follows:

Algorithm 5.4 Basic interior-point algorithm.

Given $0 < \sigma < 1$ and a feasible point $(x_0, \lambda_0, s_0) \in \mathcal{F}^+$.

repeat until stopping criterion is satisfied

1. Solve (5.19) and determine Newton's step $\begin{bmatrix} \Delta x_k \\ \Delta \lambda_k \\ \Delta s_k \end{bmatrix}$.
2. Compute the next strictly feasible iterate according to (5.20).
3. $\mu_{k+1} \leftarrow \sigma \mu_k$.

end (repeat)

5.3.2.1 Infeasible-Interior-Point Method

Unlike the interior-point method, the infeasible-interior-point method admits iterates $(x, \lambda, s) \notin \mathcal{F}^+$ and only requires $\lambda > 0, s > 0$. The algorithm is similar to the interior-point method except that two new conditions are imposed on the choice of the step-size α_k [90]. Firstly, the neighbourhood of the infeasible iterates is restricted to the amount by which the violated constraints ($Hx + A^T y + f = 0, Ax - b + s = 0$) are required to

decrease. Secondly, Armijo-like condition is enforced to ensure that μ decrease by at least some fraction of the predicted decrease at each step. Alternative implementation of exterior-point methods include [91] and Mehrotra predictor-corrector algorithm [92].

5.3.3 Projected Gradient Method

Active-set methods tend to move along edges and faces of boundary of feasible set. As a result, an active-set method changes its working set of constraints slowly, usually by a single index at each iteration. A second class of methods known as the projected gradient method, allow more substantial rapid changes to the working set by choosing a descent direction Δx and searching along the piecewise linear path $\mathcal{P}(x - \alpha\Delta x)$, where $\alpha > 0$ and \mathcal{P} is the projection onto a feasible set. It is most efficient when the constraints are simple in form in particular, when there are only bounds on the variables.

Consider the COP problem rewritten as,

$$\min_x f(x) \quad \text{s.t. } x \in \mathcal{X}. \quad (5.21)$$

The projection of a point y onto \mathcal{X} is the mapping $\mathcal{P} : \mathbb{R}^n \rightarrow \mathcal{Z}$ defined by

$$\mathcal{P}(y) = \operatorname{argmin}_x \|x - y\| \quad \text{s.t. } x \in \mathcal{X}. \quad (5.22)$$

Each iteration of the gradient projection algorithm consists of two stages. In the first stage, we search along the steepest descent direction from the current point x_k i.e.

$$x_{k+1} = x_k + \alpha_k \Delta x_k, \quad (5.23)$$

and then project the possible iterate x_{k+1} on \mathcal{X} to obtain a feasible iterate \bar{x}_{k+1} . The resulting feasible direction ($\bar{\Delta}x_k = \bar{x}_{k+1} - x_k$) is also known as the projected gradient. In the the second stage, we take a step along the feasible direction using stepsize ρ_k i.e

$$x_{k+1}^c = x_k + \rho_k \bar{\Delta}x_k, \quad (5.24)$$

There are a number of choices for determining the parameter α_k and ρ_k to ensure convergence(see [93]). A popular choice is the "Armijo Rule Along Feasible Direction",

which sets $\alpha_k = \alpha$ and uses the Armijo rule to find the local minimizer $f(x)$ along the resulting feasible direction. This first local minimizer is called the Cauchy point x_{k+1}^c . Furthermore, an improvement step of the Cauchy point is computed for better convergence rate by approximately solving subproblem which explores the face of the feasible box on which the Cauchy point lies (see [86] for details). Here, an outline for a projected gradient method for simple constraint set $\mathcal{X} = \{x : l < x < u\}$ is presented.

Algorithm 5.5 ([86]). The outline of a projected gradient method is as follows:

Algorithm 5.5 Basic projected gradient algorithm.

Given a starting point $x_0 \in \mathbf{dom} f$, $\alpha > 0$, $\alpha_k \leftarrow \alpha \forall k$ and $l_i < \mu_i$,

where l_i , u_i and $x_{i,k}$ are the i^{th} row of l, μ, x respectively at the k^{th} iterate.

repeat until stopping criterion is satisfied

1. Compute possible iterate along descent direction $x_{k+1} = x_k + \alpha_k \Delta x_k$.
2. Obtain $\bar{x}_{k+1} = \mathcal{P}(x_{k+1})$. In this case it is given by,

$$\mathcal{P}(x_{i,k+1}) = \begin{cases} l_i & \text{if } x_{i,k+1} < l_i, \\ x_{i,k+1} & \text{if } l_i \leq x_{i,k+1} \leq u_i, \\ u_i & \text{if } x_{i,k+1} > u_i. \end{cases} \quad (5.25)$$

3. Compute projected gradient, $\bar{\Delta}x_k = \bar{x}_{k+1} - x_k$.
4. Compute $x_{k+1}^c : x_{k+1}^c = x_k + \rho_k \bar{\Delta}x_k$, ρ_k satisfies Armijo condition.
5. Update $x_{k+1} : x_{k+1} = x_{k+1}^c$, No improvement step.

end (repeat)

5.3.4 Projected Fast Gradient Method

A very important extension of projected gradient method is discussed here. It is a mix of the projected gradient method (see § 5.3.3) and the Nesterov gradient method. [23].

Algorithm 5.6 ([86]). The outline of a projected fast gradient method is as follows:

Algorithm 5.6 Simplified fast projected gradient algorithm [23].

Given a starting point $x_0 \in \mathbf{dom} f$, $\beta \in (0, 1)$, $y_0 = x_0$ and $q = \frac{\mu}{L}$

repeat until stopping criterion is satisfied

1. Compute $\beta_k \in (0, 1)$ from $\beta_{k+1}^2 = (1 - \beta_{k+1})\beta_k^2 + q\beta_{k+1}$.
2. Set $\theta_k : \theta_k = \frac{\beta_k(1 - \beta_k)}{\beta_{k+1} + \beta_k^2}$.
3. Compute $y_{k+1} = x_{k+1} + \theta_k(x_{k+1} - x_k)$.
4. Compute possible Nesterov iterate $x_{k+1} = y_k - \alpha_k \nabla f(y_k)$, with $\alpha_k = \frac{1}{L}$.
5. Obtain $\bar{x}_{k+1} = \mathcal{P}(x_{k+1})$.
6. Compute projected gradient, $\bar{\Delta}x_k = \bar{x}_{k+1} - x_k$.
7. Compute $x_{k+1}^c : x_{k+1}^c = x_k + \rho_k \bar{\Delta}x_k$, ρ_k satisfies Armijo condition.
8. Update $x_{k+1} : x_{k+1} = x_{k+1}^c$, No improvement step.

end (repeat)

5.4 Summary

This chapter has presented a detailed description of some optimization approaches that can be applied to the implementation of online-optimizing anti-windup schemes. In particular, the fast gradient method has been noted to be more suitable in this regard. In Chapter 7, the efficiency of fast gradient methods of [23] are further improved using the secant properties of convex functions. The work in Chapter 8 improves on the fast gradient methods of [26, 109] by using nonmonotonic discrete Lyapunov functions.

Chapter 6

Algorithmic Analysis

This chapter provides the mathematical tools that are used in analyzing the convergence behaviour and numerical results of the algorithmic contributions presented in Chapter 7 and Chapter 8.

An algorithm is a set of ordered well-defined instructions for solving an instance of a class of problems in a finite number of steps. Algorithmic analysis is a tool that explains how an algorithm behaves from its initialization stage to its termination stage. Algorithm analysis provides theoretical estimates for the resources needed by an algorithm for solving a given computational problem. Moreover, these estimates provide an insight into reasonable directions of improving the efficiency of the considered algorithm. Algorithmic analysis serve as a tool for predicting performance, classifying and comparing algorithms.

In theoretical analysis of algorithms, it is common to estimate their complexity in the asymptotic sense i.e. for arbitrarily large input. Furthermore, in asymptotic algorithmic analysis, worst-case complexity certificates are preferred because they guarantee that the algorithm will never take more than this time [110]. Such guarantees can be quite important in time-critical applications such as missile and aircraft guidance systems, air-traffic control and control of a nuclear power plant. In addition, the worst-case running time of an algorithm is often found to occur frequently in practical applications. Notations such as the Big Oh notation are used to this end, to describe worst-case

asymptotic complexities of algorithms.

The Big Oh notation is presented in Section 6.1 while classification of algorithmic performance based on convergence measures, complexity measures and performance profiles are discussed in Section 6.2.

6.1 The Big Oh (\mathcal{O}) Notation

The big Oh (\mathcal{O}) notation characterizes functions according to their growth/decay rates and is thus useful for characterizing the effectiveness of algorithms.

Definition 6.1 ([111]). *Let $f(k)$ and $g(k)$ be two functions on some subset of real numbers. Then*

$$f(k) \in \mathcal{O}(g(k)) \quad \text{as } k \rightarrow \infty \quad (6.1)$$

if and only if there is a positive constant M such that

$$\|f(k)\| \leq M\|g(k)\| \quad \text{for all } k > k_o. \quad (6.2)$$

6.2 Algorithmic Performance

Algorithmic performance of a solver \mathcal{S} on problem \mathcal{P} is the total amount of computational effort, which is required by the solver \mathcal{S} to solve the problem \mathcal{P} . Algorithmic performance can be classified based on

1. Convergence measures
2. Complexity measures
 - (a) space-complexity
 - (b) time-complexity
 - (c) oracle-complexity
3. Performance profiles

6.2.1 Convergence Measures

Iterative methods create a sequence $\{x_k\}_{k \geq 0} \subset \mathbb{R}^n$ that converge to some desired point x^* (often the local minimizer). The fundamental question is how fast is this convergence?

Definition 6.2. *Given a sequence $\{x_k\}_{k \geq 0}$. If for any $\epsilon > 0$ there exists an N such that $\|x_k - x^*\| < \epsilon$ for all $k > N$, then the sequence $\{x_k\}_{k \geq 0}$ converges to a limit point x^* .*

Given a sequence $\{x_k\}_{k \geq 0} \rightarrow x^*$. Define the error $e_k = \|x_k - x^*\|$ in a suitable norm. The task then becomes one of measuring the convergence properties of the positive sequence $\{e_k\}_{k \geq 0}$.

- Q-convergence measure [112–114] : Assume that for some α , the condition

$$\lim_{k \rightarrow \infty} \frac{e_{k+1}}{e_k^\alpha} = \mu, \quad \text{where } \mu \geq 0, \quad (6.3)$$

is satisfied. Then the order of convergence of $\{e_k\}_{k \geq 0}$ is α and the limit value μ is the rate of convergence or asymptotic constant. Larger values of α , and smaller values of μ for a given α , corresponds to faster convergence.

- If $\alpha = 1$ and $0 < \mu < 1$, then the sequence $\{e_k\}_{k \geq 0}$ converges linearly.
- If $\alpha = 1$ and $\mu = 1$ and the sequence is known to converge (since $\mu = 1$ does not tell us if it converges or diverges), then the sequence $\{e_k\}_{k \geq 0}$ is said to converge sublinearly i.e. $\alpha < 1$. If in addition

$$\lim_{k \rightarrow \infty} \frac{e_{k+2}}{e_{k+1}} = 1, \quad (6.4)$$

then the sequence converges logarithmically.

- If $\alpha = 1$ and $\mu = 0$ OR $\alpha = 1$ and $\mu = \mu_k : \lim_{k \rightarrow \infty} \mu_k = 0$, then the sequence converges superlinearly i.e. $1 < \alpha < 2$.
- If $\alpha = 2$, the sequence has a quadratic convergence.
- If $\alpha = 3$, the sequence has a cubic convergence.

- R-convergence measure: The order/rate scale is sometimes not sufficiently fine to differentiate between sequences. The R-convergence measure captures sequences that possess variable speed of convergence. The R-convergence measure is a weaker notion than the Q-convergence measure and thus useful for characterizing the behaviour of sequences that possess variable speed of convergence.

Definition 6.3 ([111]). *Suppose $\{\zeta_k\}_{k \geq 0}$ is a sequence known to converge to 0 with an order of α and a rate of μ in the sense of (6.3). Then the positive sequence $\{e_k\}_{k \geq 0}$ has an order of at least α and a rate of at most μ if there exists a nonnegative constant M such that*

$$e_k \leq M\zeta_k \quad \text{for all } k. \quad (6.5)$$

Using the big Oh notation, inequality (6.5) can then be written as

$$e_k \in \mathcal{O}(\zeta_k). \quad (6.6)$$

Even though Definition 6.3 permits $\{\zeta_k\}_{k \geq 0}$ to be an arbitrary sequence, usually ζ_k is taken to be $\frac{1}{k^n}$, for some $n > 0$. In essence, $\{e_k\}_{k \geq 0}$ does not decay worse than $\{\frac{1}{k^n}\}_{k \geq 0}$ and thus the sequence $\{x_k\}_{k \geq 0}$ converges to x^* such that

$$x_k = x^* + \mathcal{O}\left(\frac{1}{k^n}\right). \quad (6.7)$$

6.2.2 Complexity Measures

The total complexity of an optimization algorithm is amount of the computational effort required to reach the optimal solution of a given problem. This depends on the convergence rate of the optimization algorithm as well as the computational complexity per iteration of the algorithm. As earlier noted on page 89, it is frequently important to know how much of a particular resource (execution time and memory space) is theoretically required for a given algorithm. Methods have been developed to obtain such

quantitative answers (time-complexity and space-complexity) to be explicitly expressed as a function of the problem size.

Space-Complexity Cost : This refers to the amount of memory locations utilized by the algorithm to store variables or reference data (e.g. cache usage), in order for the algorithm to reach the optimal solution of a given problem.

Time-Complexity Cost : Suppose a particular algorithm is composed of various components, (e.g search direction updates, gradient evaluations), each of which has complexity C_i and each component is executed R_i times. Then the total complexity cost, F_T , of using such an algorithm is given by

$$F_T = \sum_{i=1} C_i \times R_i. \quad (6.8)$$

R_i is the no of iterations required by each component before the optimal solution is reached. An important parameter in determining C_i is the number of floating point operations (flops). The flops refer to elementary mathematical operations of addition, subtraction, multiplication and division of two floating-point numbers. The following give examples of flops commonly encountered in optimization algorithms.

1. addition or subtraction or scalar multiplication of n-vectors : n flops.
2. inner product of two n-vectors : $2n - 1$ flops.
3. outer product of an n-vector and m-vector : $2nm$ flops.
4. matrix-vector product with $m \times n$ - matrix A : $m(2n - 1)$ flops.

The total-complexity cost, F_T is independent of the machine on which the algorithm is run. A more indicative measure of the time-complexity is the running-time (RT). The run-time of the algorithm depends on the word-size, cache and main memory sizes, processor and bus speeds of the machine platform on which the algorithm is run [115, § 1.7]. Though the running-time (RT) is machine-dependent, it still serve as a very useful tool for comparative evaluation of algorithms if all the algorithms are run

on the same machine, moreso if its implementation is single -threaded and sequential computed on a single-CPU unit

Oracle-Complexity Cost [23] : The oracle \mathcal{O} is an information unit, which answers the successive questions of the algorithm. The three main types of oracle used in optimization are

- Zero-order oracle: function value $f(x)$,
- First-order oracle: function value $f(x)$ and gradient $\nabla f(x)$,
- Second-order oracle: function value $f(x)$, gradient $\nabla f(x)$ and Hessian $\nabla^2 f(x)$.

The oracle-complexity cost is the number of calls to the oracle, which is required by the solver \mathcal{S} to solve the problem \mathcal{P} with the accuracy $\epsilon > 0$.

6.2.3 Performance Profiles

The interpretation and analysis of data generated by algorithms usually involve tables displaying the performance of each solver on each problem for a set of metrics such as CPU time, number of function evaluation, iteration counts. The performance profile as introduced in [116] is used in evaluating and comparing the overall performance of an optimization software over a set of problems. The performance profile for a solver is the (cumulative) distribution function of a performance metric. This thesis uses this concept of performance profiling to evaluate and compare the performance of a set of solvers \mathcal{S} on a problem set \mathcal{P} . Assume there are n_s solvers and n_p problems, and the computation run-time is to be used as the performance metric. For each problem p and solver s , define

$$t_{p,s} = \text{computing time required to solve problem } p \text{ by solver } s. \quad (6.9)$$

The performance ratio compares the performance on a problem $p \in \mathcal{P}$ by solver $s \in \mathcal{S}$

to the best performance by any solver on this problem.

$$r_{p,s} = \frac{t_{p,s}}{\min t_{p,s} : s \in \mathcal{S}}. \quad (6.10)$$

Thus the performance ratio $r_{p,s} \in [1, r_M]$, where $r_M \geq r_{p,s}$ is for all p, s . This performance ratio $r_{p,s}$ is indicative of how slow a solver performs on a given problem. Subsequently, we define the performance profile which is indicative of the overall assessment of the performance of a solver $s \in \mathcal{S}$ on a set of problems \mathcal{P} .

Definition 6.4 ([116]). *The performance profile $p_s(\tau) : \mathbb{R} \mapsto [0, 1]$ is a continuous nondecreasing piecewise-constant function which denotes the (cumulative) distribution function of the performance ratio of a solver $s \in \mathcal{S}$ on a set of problems \mathcal{P} . It is given by*

$$p_s(\tau) = \frac{1}{n_p} \text{size} \{p \in \mathcal{P} : r_{p,s} \leq \tau\}. \quad (6.11)$$

$p_s(\tau)$ is the probability for a solver $s \in \mathcal{S}$ that a performance ratio $r_{p,s}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio. This alternatively means that a point $(\tau, p_s(\tau))$ in the performance profile of a solver can solve $100p_s(\tau)\%$ of the tested problems τ times slower than the best competing solver. As a result of this convention, $p_s(r_M) = 1$ and in addition $p_s(1)$ of a particular solver is the probability that the solver will win over all the others. At any given τ , the sum of $p_s(\tau)$ may be greater than 1 because more than one solver may have a performance ratio within τ multiples of the best possible ratio. It is shown in [116], that the performance profile $p_s(\tau)$ is relatively insensitive to small changes in the performance data used for analysis and a very useful tool for comparing how different solvers perform relative to one another.

6.3 Examples of Algorithm Analytics

In this section, the above discussed analysis tools are used to characterize the algorithmic performance of the steepest descent method and Nesterov gradient method. Consider

the minimization problem of a convex quadratic function (CQ1)

$$\text{CQ1 : } \min_{x \in \mathbb{R}^n} f(x) : \frac{1}{2} x^T H x + c^T x, \quad (6.12)$$

with positive-definite $H \in \mathbb{R}^{n \times n}$, $c \in \mathbb{R}^{n \times 1}$ and where μ and L correspond to the minimum and maximum eigenvalue of H . The condition number of H is then, $q = \frac{L}{\mu}$.

Algorithm analysis of steepest descent method

(a.) Time-Complexity Cost

Here, the total algorithmic time-complexity of steepest descent method is obtained for the unconstrained optimization problem CQ1. It can be shown that the gradient method for CQ1 converges with $\mathcal{O}(\rho^k)$ [23]. For a given starting point $x_0 \in \mathbf{dom} f$, the gradient method terminates with

$$e_k = f_k - f^* \leq M \rho^k, \quad \text{where } M = \frac{L}{2} \|x_0 - x^*\|_2^2 \text{ and } \rho = \left(\frac{q-1}{q+1}\right)^2.$$

Now, we need to find the number of iterations to reach ϵ -suboptimal solution. The phrase ϵ -suboptimal solution means that, the solution \hat{x} is such that $f(\hat{x})$ is within a distance of at most ϵ from the optimal solution $f(x^*)$ i.e. $\|f(\hat{x}) - f(x^*)\| \leq \epsilon$, where $\epsilon > 0$. Suppose solution \hat{x} is not within ϵ -suboptimality, then $\epsilon < \|f(\hat{x}) - f(x^*)\|$ holds. Hence,

$$\epsilon < f(\hat{x}) - f(x^*) \leq M \rho^k. \quad (6.13)$$

Using the power inequality $e^x \geq 1 + x$ in (6.13), we then have that,

$$\epsilon < f(\hat{x}) - f(x^*) \leq M e^{-\left(\frac{2}{q+1}\right)2k}. \quad (6.14)$$

It follows from (6.14) that,

$$k \leq \frac{q}{4} (\log(1/\epsilon) + \log M) \leq \frac{q}{4} \log(1/\epsilon). \quad (6.15)$$

Thus, the steepest descent method attains ϵ -suboptimality in $\mathcal{O}(q \log(1/\epsilon))$ iterations.. Each iteration of the steepest descent method (see pp. 76) has complexities C_i as given

below:

$$d_k = \nabla f(x_k) = c - Hx_k. \quad n(2n - 1) + n \quad \text{flops} \quad (6.16)$$

$$\alpha_k = \frac{d_k^T d_k}{d_k^T H d_k}. \quad n(2n - 1) + 2(2n - 1) \quad \text{flops} \quad (6.17)$$

$$x_{k+1} = x_k + \alpha_k d_k. \quad 2n \quad \text{flops} \quad (6.18)$$

The main computational burden is dominated by the matrix-vector multiplication and as a whole the gradient method has a complexity of $\mathcal{O}(n^2)$ per iteration. Therefore, the total complexity, F_T , in obtaining an ϵ -approximation to the optimal solution of CQ1 is $\mathcal{O}(q \log(1/\epsilon)n^2)$.

(b.) Space-Complexity Cost

The steepest descent algorithm needs to store the present iterate x_k , present search direction, d_k and the present scalar step-size α_k at any given time. Ignoring the intermediate variables required to obtain d_k , the steepest descent method therefore needs $2n + 1$ memory locations for implementation.

Algorithm analysis of Nesterov gradient method

(a.) Time-Complexity Cost

Here, the total algorithmic time-complexity of steepest descent method is obtained for the unconstrained optimization problem CQ1. It can be shown that the gradient method for CQ1 converges with $\mathcal{O}(\rho^k)$ [23]. For a given starting point $x_0, y_0 \in \mathbf{dom} f$, the Nesterov gradient method terminates with

$$e_k = f_k - f^* \leq M\rho^k, \quad \text{where} \quad M = f(x_0) - f(x^*) + \frac{\gamma_0}{2}\|x^* - x_0\|^2 \quad \text{and} \quad \rho = \left(\frac{\sqrt{q} - 1}{\sqrt{q}}\right).$$

Similarly using the power inequality $e^x \geq 1 + x$, it then follows that the Nesterov gradient method attains ϵ -suboptimality in $\mathcal{O}(\sqrt{q} \log(1/\epsilon))$ iterations. Each iteration of the Nesterov gradient method (see 108, $\beta_k = \sqrt{\mu/\bar{L}}$) has complexities C_i as follows,

$$x_{k+1} = y_k - \frac{1}{L} \nabla f(y_k). \quad n(2n - 1) + n + 2n \quad \text{flops} \quad (6.19)$$

$$y_{k+1} = x_{k+1} + \frac{\sqrt{q} - 1}{\sqrt{q} + 1} (x_{k+1} - x_k). \quad n + 2n \quad \text{flops} \quad (6.20)$$

Thus, the Nesterov gradient method has a complexity of $\mathcal{O}(n^2)$ per iteration. Therefore, the total complexity, F_T , in obtaining an ϵ -approximation to the optimal solution of CQ1 is $\mathcal{O}(\sqrt{q} \log(1/\epsilon)n^2)$.

(b.) Space-Complexity Cost

The Nesterov gradient method needs to store the present iterate y_k , present gradient vector, $\nabla f(y_k)$, present iterate x_k , next iterate x_{k+1} , and the scalar step-size $1/L$ at any given time. Ignoring the intermediate variables required to obtain $\nabla f(y_k)$, the Nesterov gradient method therefore needs $4n + 1$ memory locations for implementation.

6.4 Summary

This chapter has briefly discussed how the performance of an algorithm can be classified based on convergence measures, complexity measures and performance profiling. Some of these presented performance-indices are used in analyzing the algorithmic contributions presented in Chapter 7 and Chapter 8.

Chapter 7

A Secant-Based Nesterov Gradient Method

7.1 Introduction

This chapter considers the unconstrained optimization of convex function f :

$$\text{UOP} : \quad \min_{x \in \mathbb{R}^n} f(x) \quad (7.1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous and differentiable convex function. The domain of f , $\text{dom } f$, is the convex set \mathbb{R}^n and x is a real vector. A necessary and sufficient condition for a point x^* to be a minimizer of f is $\nabla f(x^*) = 0$ [85]. Furthermore, if f is bounded below, then there exists a unique $x^* : f(x^*) = f^*$ with $f^* \leq f(x) \forall x \in \mathbb{R}^n$ [86].

The idea of enhancing or accelerating gradient methods directly has been intensively researched [20, 22, 23, 117–119] since the pioneering works of Shah et. al. [120] and Polyak [121]. Accelerated gradient methods are easy to implement and offer much lower memory requirement as compared to higher-order methods such as Newton's method. Accelerated gradient methods rely only on the local gradient and a history of past iterates when computing future ones. Accelerated gradient schemes can be thought of as momentum methods, in that the step taken at the current iteration depends on the

previous iterations, where the momentum grows from one iteration to the next [25]. Accelerated gradient methods, unlike gradient-descent methods, are not guaranteed to monotonically decrease the objective value. In other words, accelerated gradient methods are nonmonotone gradient methods that utilize the momentum from the previous iterates.

Particular schemes include the Barzilai-Bowein gradient method [109], the back-propagation method with momentum [122, 123] - a well-known algorithm in the neural network community - and a fast gradient method developed by Nesterov [23]. All the aforementioned accelerated gradient methods use only the previous iterate and as such they can be considered special cases of two-step iterative algorithms,

$$x_{k+1} = x_k - \alpha_k \nabla f(y_k) + \eta_k (x_k - x_{k-1}) \quad : \quad \alpha_k, \eta_k > 0; \quad y_k = \sum_{i=0}^{i=k} \tau_i x_i, \quad \tau_i \in \mathbb{R},$$

with appropriate choice of α_k , η_k and y_k . The nonmonotonicity of the accelerated gradient methods is beneficial and contribute to their increased convergence rate [35, 124, 125]. However they are susceptible to severe ripples or bumps in the objective values that may be detrimental and lead to wasted iterations as noted in [25]. This is the case in the Nesterov gradient method when the momentum factor has exceeded a critical value. This can happen when the condition number (i.e. $q^{-1} = \frac{L}{\mu}$, where L , μ are as defined in Section 7.2) is underestimated [25]. Moreover, accelerating the gradient method with the precise q in a well-conditioned region can also lead to wasted iterations [25]. The Lipschitz constant L can be estimated in a straightforward manner using backtracking (e.g. [126, pg. 162-163], [127, pg. 195]); however obtaining a nontrivial lower bound for the strong-convexity parameter μ is much more challenging. In [128], a backtracking approach is taken to estimate a nontrivial strong-convexity parameter. The use of fixed restart proportional to the condition number has also been considered (see [127, 129, 130]). A heuristic adaptive restart technique was recently introduced in [25] based on the idea of restarting the momentum factor to zero when a heuristic gradient condition is satisfied. The origin of momentum restart can in fact be traced

back to the late 80's (see e.g [131]). O'Donoghue and Candes [25] demonstrated dramatic speed up in the convergence rate of accelerated gradient methods by adaptively restarting the momentum factor with zero when a heuristic gradient condition is satisfied. They show that their restart scheme recovers the optimal complexity $\mathcal{O}(\sqrt{q} \ln \frac{1}{\epsilon})$ for strongly-convex quadratic functions. A significant improved performance of accelerated gradient methods combined with this heuristic adaptive restart of [25] has also been reported in [132–134].

This chapter provides a theoretical justification for the heuristic restart condition of [25] by extending the Nesterov gradient method to utilize available secant information. The proposed algorithm (**Secant-Based-NGM**) is based on updating the estimate-sequence parameter with secant information whenever possible. Furthermore, the proposed **Secant-Based-NGM** embodies an "update rule with reset" that parallels the restart rule suggested in [25].

The rest of this chapter is organized as follows: In sections 7.3 and 7.4, the Nesterov gradient method and the quasi-Newton method are discussed. The proposed **Secant-Based-NGM** is described in section 7.5. The global convergence for all convex functions will also be established in section 7.5 and numerical results are reported in section 7.6.

7.2 Notation

The function f denotes a continuous and differentiable convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. The optimal values of $f(x)$ and $\Phi_k(x)$ are denoted by f^* and Φ_k^* respectively. ∇ denotes the gradient operator and it is defined by $\nabla f(x) = \left[\frac{df(x)}{dx_1}, \dots, \frac{df(x)}{dx_n} \right]^T$. A differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ has a Lipschitz continuous gradient on \mathbb{R}^n with constant L if there exist a constant $L > 0$ such that $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$, $\forall x, y \in \mathbb{R}^n$. A continuously differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex with parameter μ if there exists a constant $\mu \geq 0$ such that $f(x) \geq f(y) + \nabla f(y)(x-y) + \frac{\mu}{2}\|x-y\|^2$, $\forall x, y \in \mathbb{R}^n$. If $\mu > 0$, then the continuously differentiable function f is strongly-convex. Subsequently,

the term trivial convexity parameter means a lower bound of convexity parameter while nontrivial convexity parameter refers to the greatest lower bound of the convexity parameter. Denote $\mathcal{F}_L^{i,p}(\mathbb{R}^n)$ as the class of convex functions that are at least i times continuously differentiable on \mathbb{R}^n and the p^{th} derivative is Lipschitz continuous on \mathbb{R}^n with the constant L . Let $\mathcal{S}_{\mu,L}^{1,1}(\mathbb{R}^n)$ be the class of convex functions with strong-convexity parameter μ and Lipschitz continuous gradient L , i.e. $\mathcal{S}_{\mu,L}^{i,p}(\mathbb{R}^n) \subseteq \mathcal{F}_L^{k,p}(\mathbb{R}^n)$.

Assumption 7.1. *A trivial convexity parameter $\mu \geq 0$ and the gradient's Lipschitz constant L are known.*

7.3 Nesterov Gradient Method

This section reviews the fast gradient method due to Nesterov [23]. Consider the following approximations of $f(x)$ at x_k :

$$\phi_k^1(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2\alpha} \|x - x_k\|^2. \quad (7.2)$$

$$\phi_k^2(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k). \quad (7.3)$$

The Nesterov gradient method [23] attempts to use approximations $\Phi_k(x)$ which are better than $\phi_k^1(x)$ but less expensive than $\phi_k^2(x)$ by defining an estimate-sequence (see Definition 7.1). Provided this estimate-sequence $\Phi_k(x)$ satisfies Nesterov's Principle (see below), then convergence to f^* is guaranteed (see Lemma 7.1).

Definition 7.1 ([23]). *A pair of sequences $\{\Phi_k(x)\}_{k=0}^\infty$, $\{\lambda_k\}_{k=0}^\infty$ is called an estimate-sequence of a function $f(x)$ if $\lambda_k \rightarrow 0$ and for any $x \in \mathbb{R}^n$ and for all $k \geq 0$, we have:*

$$\Phi_k(x) \leq (1 - \lambda_k)f(x) + \lambda_k\Phi_0(x), \quad (7.4)$$

where $\Phi_k(x)$ is some local function. ■

The Nesterov gradient method is based on the principle of utilizing a sequence of local functions $\Phi_k(x)$ whose limit approaches the greatest global lower bound of $f(x)$.

Nesterov's Principle: This principle requires that the estimate-sequence (see Definition 7.1) defined by the local functions $\Phi_k(x)$ is constructed such that

$$f(x_k) \leq \Phi_k^*, \quad \Phi_k^* = \min_x \Phi_k(x). \quad \blacksquare \quad (7.5)$$

As graphically illustrated in Fig. 7.1, Nesterov's principle ensures that the local functions $\Phi_k(x)$ constituting the estimate-sequence have a continuum of minima that approaches the minimum of $f(x)$ as $\lambda_k \rightarrow 0$. This convergence property of Nesterov's principle is made precise in Lemma 7.1.

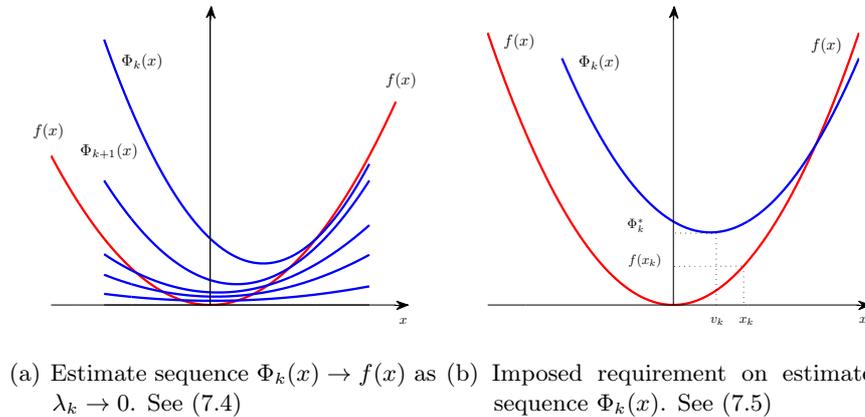


Figure 7.1: Nesterov's optimal concept.

An illustration of Nesterov's principle ($f(x_k) \leq \Phi_k^*$ and $\Phi_k(x) \rightarrow f(x)$ as $\lambda_k \rightarrow 0$).

Lemma 7.1 ([23]). *If a local function $\Phi_k(x)$ is chosen such that (7.4) and Nesterov's principle are both satisfied, then*

$$f(x_k) - f(x^*) \leq \lambda_k[\Phi_0(x^*) - f(x^*)], \quad \forall k > 0. \quad \blacksquare$$

Thus, for any a scheme that satisfies Nesterov's principle [i.e. (7.4), (7.5)], the convergence rate of its minimization process is directly related to the rate of convergence of the λ_k sequence.

The following Lemma 7.2, gives a recursive rule that satisfies Definition 7.1.

Lemma 7.2 ([23]). *Let scalars $\lambda_0 = 1$, $\beta_k \in (0, 1)$, $\sum_{k=1}^{\infty} \beta_k = \infty$. The following recursive rules,*

$$\lambda_{k+1} = (1 - \beta_k)\lambda_k, \quad (7.6)$$

$$\Phi_{k+1}(x) \leq (1 - \beta_k)\Phi_k(x) + \beta_k f(x), \quad (7.7)$$

are sufficient to constitute an estimate-sequence $\{\Phi_k(x)\}_{k=0}^{\infty}$, $\{\lambda_k\}_{k=0}^{\infty}$ in the sense of Definition 7.1. ■

Remark 7.1. *The conditions on β_k ensures that the recursion is guaranteed to be finite.*

The Nesterov scheme is one approach that ensures satisfaction of both Lemma 7.2 and Nesterov's principle. The Nesterov scheme uses Lemma 7.2 to construct the estimate-sequence defined in Lemma 7.3. Thereafter, the acceleration parameter β_k and search point y_k are carefully chosen such that (7.5) is satisfied.

7.3.1 NESTEROV'S CHOICE OF RECURSIVE RULE

Any chosen recursive rule for the local function $\Phi_k(x)$ must satisfy the requirements of Lemma 7.2. In this section, the recursive rule of the local function $\Phi_k(x)$ used in the Nesterov scheme is given and illustrated in Fig. 7.2.

Definition 7.2 ([23]). *Define $\overline{\Phi_{k+1}}(x)$ as*

$$\overline{\Phi_{k+1}}(x) = (1 - \beta_k)\Phi_k(x) + \beta_k[f(y_k) + \nabla f(y_k)^T(x - y_k) + \frac{\mu}{2}\|x - y_k\|^2] \quad (7.8)$$

for a given sequence $\{y_k\}_{k=0}^{\infty}$. ■

Remark 7.2. *In particular, the choice $\Phi_{k+1}(x) = \overline{\Phi_{k+1}}(x)$ obeys (7.7) and this corresponds to Nesterov's choice of recursive rule.*

The next local function $\Phi_{k+1}(x)$ is then a convex combination of the previous local function $\Phi_k(x)$ and the greatest global lower bound of $f(x)$. Thus, Nesterov's choice of recursive rule is graphically illustrated in Fig. 7.2.

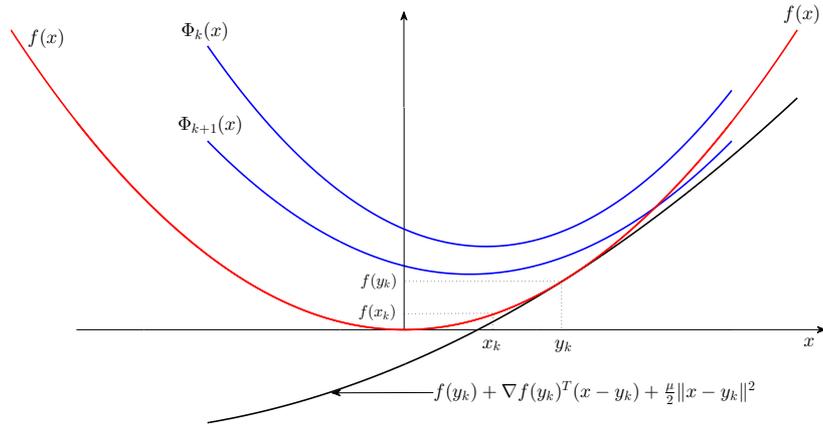


Figure 7.2: Updating the local function $\Phi_k(x)$ for some y_k .
The estimate sequence $\Phi_{k+1}(x)$ is obtained as a convex combination of $\Phi_k(x)$ and the global lower bound of $f(x)$ at y_k . (see Remark 7.2).

7.3.2 NESTEROV'S CHOICE OF ESTIMATE-SEQUENCE

A simple quadratic form is chosen as the initial local function $\Phi_0(x)$. This choice allows the requirements of Lemma 7.2 to be satisfied easily. Thus the recursion of the sequences defined in Lemma 7.3 defines an estimate-sequence that satisfies Definition 7.1.

Lemma 7.3 ([23]). *Let scalars $\beta_k \in (0, 1)$, $\gamma_k > 0$, $\mu \geq 0$, and $v_k, y_k \in \mathbb{R}^n$. Let $\Phi_0(x) = \Phi_0^* + \frac{\gamma_0}{2} \|x - v_0\|^2$. The recursive rules in Lemma 7.2 [i.e. (7.6), (7.7)] hold for*

$$\Phi_k(x) = \Phi_k^* + \frac{\gamma_k}{2} \|x - v_k\|^2 \quad (7.9)$$

provided the sequences $\{\gamma_k, v_k, \Phi_k^*\}_{k=0}^\infty$ are defined as

$$\gamma_{k+1} = (1 - \beta_k)\gamma_k + \beta_k\mu, \quad (7.10)$$

$$v_{k+1} = \frac{1}{\gamma_{k+1}} [(1 - \beta_k)\gamma_k v_k + \beta_k\mu y_k - \alpha_k \nabla f(y_k)], \quad (7.11)$$

$$\begin{aligned} \Phi_{k+1}^* = & (1 - \beta_k)\Phi_k^* + \beta_k f(y_k) - \frac{\beta_k^2}{2\gamma_{k+1}} \|\nabla f(y_k)\|^2 + \\ & \frac{\beta_k(1 - \beta_k)\gamma_k}{\gamma_{k+1}} \left[\frac{\mu}{2} \|y_k - v_k\|^2 + \nabla f(y_k)^T (v_k - y_k) \right]. \end{aligned} \quad (7.12)$$

The variables β_k and γ_k shall subsequently be referred to as the acceleration parameter

and estimate-sequence parameter respectively. Now, one has the estimate-sequence as desired but still the local condition (7.5) at the next iterate, $f(x_{k+1}) \leq \Phi_{k+1}^*$, needs to be ensured. This is subsequently achieved in §7.3.3 by carefully choosing the accelerating parameter β_k and the search point y_k such that (7.5) is satisfied.

7.3.3 NESTEROV'S CHOICE OF β_k AND SEARCH POINT y_k

Suppose that $f(x_k) \leq \Phi_k^*$. Denote $\zeta(\beta_k) = \left[f(x_k) - f(y_k) + \frac{\beta_k \gamma_k}{\gamma_{k+1}} \left[\frac{\mu}{2} \|y_k - v_k\|^2 + \nabla f(y_k)^T (v_k - y_k) \right] \right]$. Then Φ_{k+1}^* (7.12) can be written as

$$\Phi_{k+1}^* = f(y_k) - \frac{\beta_k^2}{2\gamma_{k+1}} \|\nabla f(y_k)\|^2 + (1 - \beta_k)\zeta(\beta_k). \quad (7.13)$$

Hence, the choice of y_k and β_k that satisfies Nesterov's Principle (i.e Lemma 7.3 and $f(x_{k+1}) \leq \Phi_{k+1}^*$) are obtained as follows:

Take $x_{k+1} = y_k - \alpha_k \nabla f(y_k)$, $\alpha_k = \frac{1}{L}$. Then, we have $f(x_{k+1}) \leq f(y_k) - \frac{1}{2L} \|\nabla f(y_k)\|^2$.

It then follows that (7.13) reduces to

$$\Phi_{k+1}^* \geq f(x_{k+1}) + \frac{1}{2L} \|\nabla f(y_k)\|^2 - \frac{\beta_k^2}{2\gamma_{k+1}} \|\nabla f(y_k)\|^2 + (1 - \beta_k)\zeta(\beta_k). \quad (7.14)$$

Hence, to satisfy $f(x_{k+1}) \leq \Phi_{k+1}^*$, we choose y_k as

$$y_k = \frac{\beta_k \gamma_k v_k + \gamma_{k+1} x_k}{\gamma_k + \beta_k \mu} : \zeta(\beta_k) = \left[\frac{\beta_k \gamma_k}{\gamma_{k+1}} \left[\frac{\mu}{2} \|y_k - v_k\|^2 \right] \right] \quad (7.15)$$

and compute β_k :

$$\beta_k^2 L = \gamma_{k+1} = (1 - \beta_k)\gamma_k + \beta_k \mu. \quad (7.16)$$

Remark 7.3. The search point y_k (7.15) can be written as

$$y_k = x_k - \rho_k \nabla \Phi_k(x), \quad \rho_k = \frac{\beta_k}{\gamma_k + \beta_k \mu}, \quad \nabla \Phi_k(x) = \gamma_k (x_k - v_k). \quad (7.17)$$

The Nesterov scheme is graphically illustrated as shown below in Fig. 7.3.

Algorithm 7.1b ([23]). The simplified Nesterov gradient method is outlined as follows:

Algorithm 7.1b Simplified Nesterov gradient method.

Given a starting point $x_0 \in \mathbf{dom} f$, $\beta_0 \in (0, 1)$, $y_0 = x_0$ and $q = \frac{\mu}{L}$.

repeat until stopping criterion is satisfied

1. Compute the Nesterov iterate: $x_{k+1} = y_k - \alpha_k \nabla f(y_k)$, with $\alpha_k = \frac{1}{L}$.
2. Compute $\beta_{k+1} \in (0, 1)$ from $\beta_{k+1}^2 = (1 - \beta_{k+1})\beta_k^2 + q\beta_{k+1}$
3. Compute θ_{k+1} : $\theta_{k+1} = \frac{\beta_k(1 - \beta_k)}{\beta_{k+1} + \beta_k^2}$.
4. Compute $y_{k+1} = x_{k+1} + \theta_{k+1}(x_{k+1} - x_k)$.

end (repeat)

Remark 7.4. The choice of $\beta_0 = \sqrt{\frac{\mu}{L}}$ corresponds to $\gamma_0 = \mu$ while the corresponding β_0 for the case of $\gamma_0 = L$ can be obtained from (7.16). It is important to emphasize that $\beta_0 \neq 1$ since (7.16) cannot hold when $\beta_0 = 1$. Hence the choice $\beta_0 \in (0, 1)$. Were β_k be chosen as 0 for all $k \geq 0$, **Algorithm 7.1** would reduce to a fixed-step gradient-descent method.

Theorem 7.1 ([23]). Let $\Phi_0(x) = \Phi_0^* + \frac{\gamma_0}{2}\|x - v_0\|^2$. Suppose $v_0 = x_0$. If a scheme satisfies Lemma 7.2 and Nesterov's principle, then

$$f(x_k) - f(x^*) \leq \lambda_k \left[f(x_0) - f(x^*) + \frac{\gamma_0}{2}\|x - x_0\|^2 \right], \quad \forall k > 0,$$

where $\lambda_0 = 1$ and $\lambda_k = \prod_{i=0}^{k-1} (1 - \beta_i)$. ■

Remark 7.5. Take $\gamma_0 = L$ in **Algorithm 7.1a** (or the corresponding β_0 in **Algorithm 7.1b**). Let $v_0 = x_0$. Then the Nesterov scheme satisfies the premises of Theorem 7.1. If $\gamma_k > 0$, then the Nesterov gradient method **Algorithm 7.1** generates

a sequence $\{x_k\}_{k=0}^{\infty}$ such that

$$f(x_k) - f^* \leq \frac{4L}{(k+2)^2} \times \|x_0 - x^*\|^2. \quad (7.18)$$

Furthermore, if $\gamma_k \geq \mu$ for all k , then

$$f(x_k) - f^* \leq \min \left\{ L \left(1 - \sqrt{\frac{\mu}{L}}\right)^k, \frac{4L}{(k+2)^2} \right\} \times \|x_0 - x^*\|^2. \quad (7.19)$$

7.4 Quasi-Newton Method

The NGM uses a $\Phi_k(x)$ as an inexpensive approximation to $\phi_k^2(x)$ and imposes the Nesterov's principle for convergence. The quasi-Newton [86] method, on the other hand, imposes a different requirement on the local quadratic model $\Psi_k(x)$ about x_k ,

$$\Psi_k(x) = f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2} (x - x_k)^T B_k (x - x_k), \quad (7.20)$$

where B_k is a Hessian-approximate. Lets define the point where $\Psi_k(x_k) = f(x_k)$ and

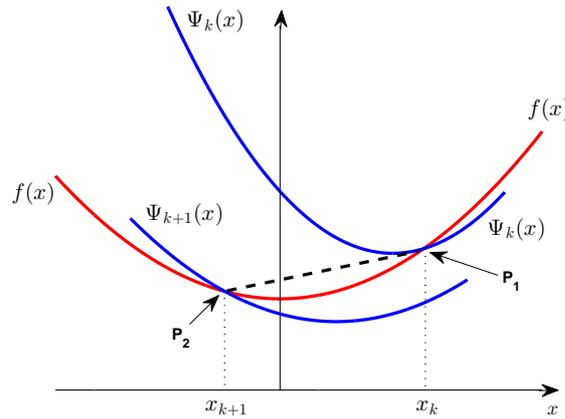


Figure 7.4: Quasi-Newton method, showing secant line P_1 - P_2 .
Illustrating the secant line P_1 - P_2 , where P_1 and P_2 are defined below. The imposed requirement on B_{k+1} is that $\nabla \Psi_{k+1}(x_k) = \nabla f(x_k)$.

$\nabla \Psi_k(x_k) = \nabla f(x_k)$ as point P_1 (see Fig. 7.4). Suppose the new iterate x_{k+1} has been

generated by minimizing $\Psi_k(x)$ i.e.

$$x_{k+1} = x_k + \alpha_k d_k = x_k - B_k^{-1} \nabla f(x_k) = x_k + s_k. \quad (7.21)$$

We wish to construct $\Psi_{k+1}(x)$ of the form

$$\Psi_{k+1}(x) = f(x_{k+1}) + \nabla f(x_{k+1})^T (x - x_{k+1}) + \frac{1}{2} (x - x_{k+1})^T B_{k+1} (x - x_{k+1}). \quad (7.22)$$

Similarly, define the point where $\Psi_{k+1}(x_{k+1}) = f(x_{k+1})$ and $\nabla \Psi_{k+1}(x_{k+1}) = \nabla f(x_{k+1})$ as P_2 (see Fig. 7.4). Requirements can be imposed on B_{k+1} based on our knowledge of the previous step. Provided the Hessian does not vary significantly, it is reasonable to expect that in addition to $\nabla \Psi_{k+1}(x_{k+1}) = \nabla f(x_{k+1})$, that it is desired to have $\nabla \Psi_{k+1}(x_k) = \nabla f(x_k)$. Therefore, it follows that

$$B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k). \quad (7.23)$$

This imposed condition (7.23) is known as the secant condition and can be written as:

$$B_{k+1} s_k = y_k. \quad (7.24)$$

Alternatively, the secant condition can be derived from the mean-value theorem for vector-valued functions which implies that (7.24) is satisfied by the mean Hessian in the interval (x_k, x_{k+1}) [27]. The pair (s_k, y_k) is said to be the secant pair associated with the secant condition (7.24). The matrix B_k is updated (see [86]) using symmetric rank-one updates (SR1) or symmetric rank-two updates (e.g Powell-Symmetric-Broyden(PSB) and Davidon-Fletcher-Powell(DFP) updates).

7.5 A Secant-Based Nesterov Gradient Method

In this section, a new accelerated gradient method (**Secant-Based-NGM**) is proposed by extending the classical Nesterov gradient method to utilize available secant information whenever possible. The **Secant-Based-NGM** is based on updating the

estimate-sequence parameter γ_k by imposing a secant condition on the choice of search point y_k . The global convergence of the proposed **Secant-Based-NGM** is also established for all convex functions.

7.5.1 RECURSIVE RULE REVISITED

Recall from Lemma 7.2 that the recursive rule for $\Phi_k(x)$ must satisfy,

$$\Phi_{k+1}(x) \leq (1 - \beta_k)\Phi_k(x) + \beta_k f(x). \quad (7.25)$$

Also recall from Definition 7.2 that

$$\overline{\Phi_{k+1}}(x) = (1 - \beta_k)\Phi_k(x) + \beta_k [f(y_k) + \nabla f(y_k)^T(x - y_k) + \frac{\mu}{2}\|x - y_k\|^2] \quad (7.26)$$

Definition 7.3. Let $l_k(x) : l_k(x) = f(y_k) + \nabla f(y_k)^T(x - y_k) + \frac{\mu_k}{2}\|x - y_k\|^2$ for a given sequence $\{y_k\}_{k=0}^{\infty}$ with $\mu_k \in (0, \mu]$.

Subsequently, we do not explore the flexibility in μ_k and take that $\mu_k = \mu$. Therefore, the equality (7.26) can be written as

$$\overline{\Phi_{k+1}}(x) = (1 - \beta_k)\Phi_k(x) + \beta_k l_k(x). \quad (7.27)$$

Thus, Nesterov's choice of recursive rule (see Remark 7.2) corresponds to $\Phi_{k+1}(x) = \overline{\Phi_{k+1}}(x)$.

In the case $\mu \neq 0$, it then follows from (7.27) that

$$\overline{\Phi_{k+1}}(x) = (1 - \beta_k) \left[\Phi_k^* + \frac{\gamma_k}{2}\|x - v_k\|^2 \right] + \beta_k \left[l_k^* + \frac{\mu}{2}\|x - z_k\|^2 \right], \quad (7.28)$$

where $z_k = y_k - \frac{1}{u}\nabla f(y_k)$, $l_k^* = f(y_k) - \frac{1}{2\mu}\|\nabla f(y_k)\|^2$.

Furthermore, the proposed algorithm chooses $\Phi_{k+1}(x) \leq \overline{\Phi_{k+1}}(x)$ such that,

$$\Phi_{k+1}(x) \leq (1 - \beta_k)\Phi_k(x) + \beta_k [f(y_k) + \nabla f(y_k)^T(x - y_k) + \frac{\mu}{2}\|x - y_k\|^2]. \quad (7.29)$$

7.5.2 CONSTRUCTION OF Secant-Based-NGM

The **Secant-Based-NGM** scheme extends the classical Nesterov scheme by utilizing secant information in updating the estimate-sequence parameter γ_k . The subsequent two lemmas are used to arrive at an inequality (7.37) that gives an upper bound to γ_k .

Lemma 7.4. *Given $a, b > 0$ and $x_1, x_2 \in \mathbb{R}^n$, there exists $x_3 \in \mathbb{R}^n$ and $d \geq 0$ such that $a\|x - x_1\|^2 + b\|x - x_2\|^2 = (a + b)\|x - x_3\|^2 + d$ for all $x \in \mathbb{R}^n$.*

Proof:

Take

$$x_3 = \frac{ax_1 + bx_2}{a + b} \quad \text{and} \quad d = \frac{ab(x_1 - x_2)^T(x_1 - x_2)}{a + b}. \quad (7.30)$$

It then follows that

$$a\|x - x_1\|^2 + b\|x - x_2\|^2 = (a + b)\|x - x_3\|^2 + d. \quad (7.31)$$

Hence, $d = 0$ if and only if $x_1 = x_2$. Thus $d \geq 0$ if $a, b > 0$. \blacksquare

Lemma 7.5. *Given $a, b > 0$ and $x_1, x_2 \in \mathbb{R}^n$, there exists $x_3 \in \mathbb{R}^n$ and $\hat{d} \in \mathbb{R}$ such that $a\|x - x_1\|^2 + bx_2^T x = a\|x - x_3\|^2 + \hat{d}$ for all $x \in \mathbb{R}^n$.*

Proof:

Take

$$x_3 = \frac{2ax_1 - bx_2}{2a} \quad \text{and} \quad \hat{d} = \frac{4abx_1^T x_2 - bx_2^T x_2}{4a^2}. \quad (7.32)$$

It then follows that $a\|x - x_1\|^2 + bx_2^T x = a\|x - x_3\|^2 + \hat{d}$, where $\hat{d} \in \mathbb{R}$ if $a, b > 0$. \blacksquare

The subsequent Lemma 7.6 gives the needed freedom in updating the γ_k sequence and also gives the upper bound to the acceptable value of γ_k .

Lemma 7.6. *Let scalars $\beta_k \in (0, 1)$, $\gamma_k > 0$, $\mu \geq 0$, and $v_k, y_k \in \mathbb{R}^n$. Let $\Phi_0(x) = \Phi_0^* + \frac{\gamma_0}{2}\|x - v_0\|^2$. The recursive rules in Lemma 7.2 [i.e. (7.6), (7.7)] hold for*

$$\Phi_k(x) = \Phi_k^* + \frac{\gamma_k}{2}\|x - v_k\|^2 \quad (7.33)$$

provided the sequences $\{\gamma_k, v_k, \Phi_k^*\}_{k=0}^\infty$ are defined as

$$\gamma_{k+1}^F = (1 - \beta_k)\gamma_k + \beta_k\mu, \quad (7.34)$$

$$v_{k+1} = \frac{1}{\gamma_{k+1}^F}[(1 - \beta_k)\gamma_k v_k + \beta_k\mu y_k - \alpha_k \nabla f(y_k)], \quad (7.35)$$

$$\begin{aligned} \Phi_{k+1}^* &= (1 - \beta_k)\Phi_k^* + \beta_k f(y_k) - \frac{\beta_k^2}{2\gamma_{k+1}^F} \|\nabla f(y_k)\|^2 + \\ &\quad \frac{\beta_k(1 - \beta_k)\gamma_k}{\gamma_{k+1}^F} \left[\frac{\mu}{2} \|y_k - v_k\|^2 + \nabla f(y_k)^T (v_k - y_k) \right], \end{aligned} \quad (7.36)$$

$$\gamma_{k+1} \leq \gamma_{k+1}^F. \quad (7.37)$$

Proof:

Given that $\Phi_k(x) = \Phi_k^* + \frac{\gamma_k}{2} \|x - v_k\|^2$. Define $\gamma_{k+1}^F = (1 - \beta_k)\gamma_k + \beta_k\mu$. From (7.27), $\overline{\Phi_{k+1}}(x)$ is defined as $\overline{\Phi_{k+1}}(x) = (1 - \beta_k)\Phi_k(x) + \beta_k l_k(x)$. Furthermore, choose v_{k+1} as the unconstrained minimum of $\overline{\Phi_{k+1}}(x)$. Thus,

$$v_{k+1} = \frac{1}{\gamma_{k+1}^F} [(1 - \beta_k)\gamma_k v_k + \beta_k\mu y_k - \beta_k \nabla f(y_k)]. \quad (7.38)$$

Moreover by virtue of Lemma 7.4, it follows from (7.28) that $\overline{\Phi_{k+1}}(x) = (1 - \beta_k)\Phi_k^* + \beta_k l_k^* + \frac{\gamma_{k+1}^F}{2} \|x - v_{k+1}\|^2 + d$ for some $d \geq 0$. The case of $d = 0$ occurs if and only if we have coincident minimizers i.e $v_{k+1} = v_k = z_k$ (see [119]).

Nesterov's choice corresponds to $\Phi_{k+1}(x) = \overline{\Phi_{k+1}}(x)$ with $\gamma_{k+1} = \gamma_{k+1}^F$. However, we choose

$$\gamma_{k+1} \leq \gamma_{k+1}^F. \quad (7.39)$$

We then have $\Phi_{k+1}(x) \leq \overline{\Phi_{k+1}}(x)$ which still satisfies (7.25) and more importantly $\Phi_{k+1}(v_{k+1}) = \overline{\Phi_{k+1}}(v_{k+1})$ also as a consequence of Lemma 7.4. This inequality (7.39) still holds for the case of $\mu = 0$ using similar arguments in conjunction with Lemma 7.5. This inequality (7.39) is crucial in the sense that it gives an upper bound to γ_k for all k and allows the use of the secant information in updating the γ_k sequence whenever

possible.

Let us compute Φ_{k+1}^* as follows. It follows from (7.33) that at $x = v_{k+1}$, we have that $\Phi_{k+1}^* = \Phi_{k+1}(v_{k+1})$. Since $\Phi_{k+1}(v_{k+1}) = \overline{\Phi_{k+1}}(v_{k+1})$, then by (7.27) it then follows that Φ_{k+1}^* can be computed as

$$\Phi_{k+1}^* = (1 - \beta_k)\Phi_k(v_{k+1}) + \beta_k l_k(v_{k+1}). \quad (7.40)$$

Substitute for $\Phi_k(v_{k+1})$ in (7.40) using (7.33), then (7.40) becomes

$$\begin{aligned} \Phi_{k+1}^* &= (1 - \beta_k)\Phi_k^* + \frac{(1 - \beta_k)\gamma_k}{2}\|v_{k+1} - v_k\|^2 + \\ &\quad \beta_k f(y_k) + \beta_k \nabla f(y_k)(v_{k+1} - y_k) + \frac{\beta_k \mu}{2}\|v_{k+1} - y_k\|^2. \end{aligned} \quad (7.41)$$

It follows from (7.38) that

$$v_{k+1} - y_k = \frac{\gamma_k}{\gamma_{k+1}^F}(1 - \beta_k)(v_k - y_k) - \frac{\beta_k}{\gamma_{k+1}^F}\nabla f(y_k), \quad (7.42)$$

and that

$$v_{k+1} - v_k = \frac{\beta_k \mu}{\gamma_{k+1}^F}(y_k - v_k) - \frac{\beta_k}{\gamma_{k+1}^F}\nabla f(y_k). \quad (7.43)$$

Substituting (7.42) and (7.43) into (7.41), then the result for Φ_{k+1}^* follows,

$$\begin{aligned} \Phi_{k+1}^* &= (1 - \beta_k)\Phi_k^* + \beta_k f(y_k) - \frac{\beta_k^2}{2\gamma_{k+1}^F}\|\nabla f(y_k)\|^2 + \\ &\quad \frac{\beta_k(1 - \beta_k)\gamma_k}{\gamma_{k+1}^F} \left[\frac{\mu}{2}\|y_k - v_k\|^2 + \nabla f(y_k)^T(v_k - y_k) \right]. \quad \blacksquare \end{aligned} \quad (7.44)$$

Remark 7.6. Lemma 7.3 is a special case of Lemma 7.6 if γ_{k+1} is chosen as $\gamma_{k+1} = \gamma_{k+1}^F$.

The secant information is used in updating γ_k by requiring that $\nabla \Phi_{k+1}(y_k) = \nabla f(y_k)$.

It then follows that $\hat{\gamma}_{k+1}(y_k - v_{k+1}) = \nabla f(y_k)$ where $\hat{\gamma}_{k+1}$ is a possible update of γ_k .

Using a symmetric rank-1 update, it then follows that :

$$\hat{\gamma}_{k+1} = \frac{\nabla f(y_k)(y_k - v_{k+1})}{(y_k - v_{k+1})^T(y_k - v_{k+1})}. \quad (7.45)$$

Updating γ_k with (7.45) (i.e $\gamma_{k+1} = \hat{\gamma}_{k+1}$) ensures that the computed search point y_k (7.17) utilizes available secant information at the k^{th} iterate. However, this updating is subject to the constraint (7.37). This computation of $\hat{\gamma}_{k+1}$ comes at an extra cost of 2 vector-vector multiplication. However, as shown in the simulation results, the benefits of computing γ_{k+1} outweigh the extra cost of its computation. Thus, the update γ_{k+1} can be appended to the classical Nesterov gradient method in a straight-forward manner as shown below in step 6 of the **Basic Secant-Based-NGM** below.

Algorithm 7.2a (Basic Secant-Based-NGM). The outline is as follows:

Basic Secant-Based-NGM.

Given a starting point $x_0 \in \mathbf{dom} f$, $\gamma_0 > 0$, $v_0 = x_0$, $\epsilon > 0$ and $\epsilon \approx 0$

repeat until stopping criterion is satisfied

1. Compute $\beta_k \in (0, 1)$ from $\beta_k^2 L = (1 - \beta_k)\gamma_k + \beta_k\mu$.
2. Compute γ_{k+1}^F : $\gamma_{k+1}^F = (1 - \beta_k)\gamma_k + \beta_k\mu$. (7.34)
3. Compute search point: $y_k = x_k - \rho_k\gamma_k(x_k - v_k)$.
4. Compute the Nesterov iterate: $x_{k+1} = y_k - \alpha_k\nabla f(y_k)$, with $\alpha_k = \frac{1}{L}$.
5. Compute v_{k+1} : $v_{k+1} = \frac{1}{\gamma_{k+1}^F}[(1 - \beta_k)\gamma_k v_k + \beta_k\mu y_k - \beta_k\nabla f(y_k)]$. (7.45)
6. Compute $\hat{\gamma}_{k+1}$: $\hat{\gamma}_{k+1} = \frac{\nabla f(y_k)(y_k - v_{k+1})}{(y_k - v_{k+1})^T(y_k - v_{k+1})}$. (7.35)
7. Compute γ_{k+1} : $\gamma_{k+1} = \widehat{\min}_\mu(\hat{\gamma}_{k+1}, \gamma_{k+1}^F)$ update rule
8. If $\hat{\gamma}_{k+1} < 0$, then set $\gamma_{k+1} = \min\{\max\{\epsilon, \beta_k\mu\}, \gamma_{k+1}^F\}$ reset rule

end (repeat)

Remark 7.7. The $\widehat{\min}_\mu$ operator rule in step 7 is given by

$$c = \widehat{\min}_\mu(a, b) : \begin{cases} c = \min(a, b) & \text{if } a > \mu, \\ c = b & \text{if } a < \mu. \end{cases}$$

Remark 7.8. The reset rule in step 8 is equivalent to $\gamma_{k+1} = \beta_k \mu$ if $\mu \neq 0$ and $\gamma_{k+1} = \min\{\epsilon, \gamma_{k+1}^F\}$ if otherwise.

Just as with the classical Nesterov gradient method, $v_{k+1} = x_{k+1} + \frac{1 - \beta_k}{\beta_k}(x_{k+1} - x_k)$ and the variable v_k can therefore be eliminated. With this elimination of v_{k+1} , the **Basic Secant-Based-NGM** simplifies to the **Simplified Secant-Based-NGM**.

Algorithm 7.2b (Simplified Secant-Based-NGM). The outline is as follows:

Simplified Secant-Based-NGM.

Given a starting point $x_0 \in \mathbf{dom} f$, $\beta_0 \in (0, 1)$, $y_0 = x_0$, $\epsilon > 0$ and $\epsilon \approx 0$
repeat until stopping criterion is satisfied

1. Compute Nesterov iterate: $x_{k+1} = y_k - \alpha_k \nabla f(y_k)$, with $\alpha_k = \frac{1}{L}$.
2. Compute $\gamma_{k+1}^F = \beta_k^2 L$; $\tau_k = \frac{1 - \beta_k}{\beta_k}$.
3. Compute $y_v = [\alpha_k \nabla f(y_k) - \tau_k(x_{k+1} - x_k)]$; $\hat{\gamma}_{k+1} = \frac{y_v^T \nabla f(y_k)}{y_v^T y_v}$.
4. Compute $\gamma_{k+1} : \gamma_{k+1} = \widehat{\min}_\mu(\hat{\gamma}_{k+1}, \gamma_{k+1}^F)$ update rule
5. If $\hat{\gamma}_{k+1} < 0$, then set $\gamma_{k+1} = \min\{\max\{\epsilon, \beta_k \mu\}, \gamma_{k+1}^F\}$ reset rule
6. Compute $\beta_{k+1} \in (0, 1)$ from $\beta_{k+1}^2 L = \gamma_{k+1} - \beta_{k+1}(\gamma_{k+1} - \mu)$.
7. Compute $\theta_{k+1} : \theta_{k+1} = \rho_{k+1} \gamma_{k+1} \tau_k$, where $\rho_{k+1} = \frac{\beta_{k+1}}{\gamma_{k+1} + \beta_{k+1} \mu}$.
8. Compute $y_{k+1} = x_{k+1} + \theta_{k+1}(x_{k+1} - x_k)$.

end (repeat)

In what follows, the gradient restart condition and restart rule of [25] is contrasted with a gradient condition (7.46) and the proposed "update rule with reset" respectively. With the substitution of v_{k+1} in the **Basic Secant-Based-NGM**, we can see that the reset condition $\gamma_{k+1} < 0$ in step 5 of the **Simplified Secant-Based-NGM** is equivalent to

$$\alpha_k \|\nabla f(y_k)\|^2 - \tau_k \nabla f(y_k)^T (x_{k+1} - x_k) < 0. \quad (7.46)$$

This gradient condition (7.46) is more conservative than the gradient-scheme restart condition suggested in [25] especially when the iterates are far away from the optimum point. Thus the gradient condition (7.46) is less frequently satisfied. The advantage of the conservativeness of (7.46) is reinforced by the observation in [25] that "*... restarting far from the optimum can slow down the early convergence slightly, until the quadratic region is reached and the algorithm enters the rapid linear convergence phase.*".

The restart rule of [25] is given as

1. setting β_{k+1} as 1.
2. setting the momentum factor θ_{k+1} as 0.

Firstly, it should be noted that the reset β_{k+1} should be in the interval $(0, 1)$ since $\beta_k \in (0, 1)$ for all $k \geq 0$ (see Remark 7.4). Moreover, an arbitrary choice of $\beta_{k+1} \in (0, 1)$ may correspond to a γ_{k+1} that violates the inequality $\gamma_{k+1} \leq \gamma_{k+1}^F$ (7.39). Furthermore, the proposed **Simplified Secant-Based-NGM** proceeds with the computed β_{k+1} unlike the restart rule 1 of [25]. Also the proposed **Simplified Secant-Based-NGM** does not reset the momentum factor to zero (i.e. $\theta_{k+1} \neq 0$) unlike the restart rule 2 of [25]. Thus **Secant-Based-NGM** is not a momentum-restart algorithm, and the proposed "update rule with reset" (see Remarks 7.7, 7.8) in the **Secant-Based-NGM** satisfies inequality (7.39).

7.5.3 GLOBAL CONVERGENCE OF PROPOSED SCHEME

The scheme construction of the proposed algorithm **Secant-Based-NGM** satisfies Lemma 7.2 and Nesterov's principle. Thus the proposed scheme satisfies the premises

of Theorem 7.1 and therefore **Secant-Based-NGM** is globally convergent with

$$f(x_k) - f(x^*) \leq \prod_{i=0}^{k-1} (1 - \beta_i) \left[f(x_0) - f(x^*) + \frac{\gamma_0}{2} \|x - x_0\|^2 \right], \quad \forall k \geq 1, \quad (7.47)$$

where $\beta_i \in (0, 1)$ and $\gamma_0 > 0$.

7.6 Simulation Examples

Consider two test examples of the form

$$\text{UOP : } \min_{x \in \mathbb{R}^n} f(x), \quad (7.48)$$

where $x \in \mathbb{R}^n$. The numerical tests investigate the effects of increasing the dimension and condition number respectively on the performance of the proposed algorithm.

7.6.1 Simulation Setup

All numerical tests were coded in 64-bit MATLAB on a Dell-Optilex-780 PC with Intel dual-core CPU of 2.93 GHz, RAM of 16GB and a 150GB-free Hard-Disk. All Matlab sessions were single-threaded, `feature('accel','off')` and process-Priority set to "High". All Matlab sessions were executed on the PC running Windows 7 in "Safe Mode". The stopping criterion was $\|\nabla f(x)\| \leq 10^{-9}$ and $\|\nabla f(x)\| \leq 10^{-6}$ for Ex. 1 and 2 respectively. All matrices are square and random such that the Hessian has eigenvalues in the interval $[1, L]$. This was achieved using singular value decomposition to allocate the desired eigenvalues. All matrices and vectors were randomly generated in MATLAB with seed `rng(1234, 'twister')`. The simulations were also repeated for seed `rng(5678, 'twister')`. In the case of Ex. 1, the average run-time of 100 random simulations is also reported.

7.6.2 Test Functions and Solvers

The test function in Ex. 1 is a well-known convex quadratic function used for benchmarking convex solvers while the test function in Ex. 2 is a convex non-quadratic

function usually encountered in machine-learning literature. The set of solvers \mathcal{S} considered is

- Classical Nesterov gradient method(NGM)¹,
- Adaptive restart [25],
- Fixed restart,² after $k = \sqrt{\frac{8L}{\mu}}$
- Proposed algorithm **Simplified Secant-Based-NGM**.

In all cases, $\beta_0 = \sqrt{\mu/L}$ if $\mu \neq 0$ and $\beta_0 = \frac{\sqrt{5}-1}{2}$ if otherwise. These chosen β_0 corresponding to $\gamma_0 = \mu$ and $\gamma_0 = L$ respectively. The value of ϵ in **Simplified Secant-Based-NGM** is taken as 10^{-16} .

The performance profile in the sense of Dolan and Moré [116] is adopted to analyze the performance data of the above set of solvers \mathcal{S} on a problem set \mathcal{P} (e.g. Ex. 1). The percentage of the test problems for which a method is the fastest is given on the left axis of the plot. The right-hand side of the plot gives the percentage of the test problems that were successfully solved by each of the methods. In essence, the right side of the performance profile plot is a measure of an algorithm's robustness.

7.6.3 Numerical Results

Ex. 1: Ridge regression problem [135]

This is a linear least squares problem with Tikhonov regularization. Given $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^n$ and $\mu = 0.1$.

$$f(x) = \frac{\mu}{2} \|x\|_2^2 + \frac{1}{2} \|Ax - b\|_2^2. \quad (7.49)$$

The objective function $f(x) \in \mathcal{S}_{\mu, L}^{1,1}$ is a positive-definite quadratic convex function with Lipschitz gradient of $L = \lambda_{\max}(A^T A) + \mu$ and global convexity parameter of $\mu = 0.1$. All algorithms uses the global convexity parameter except algorithm NGM1 which used

¹The simplified Nesterov gradient method **Algorithm 7.1b** is used for simulation.

²In the case of example 2 ($\mu = 0$), fixed restart is done after $k = \min\{N, \sqrt{L}\}$ iterations.

a nontrivial convexity parameter of $\mu = 1.1$.

The plots in Fig. 7.5(a) and Fig. 7.5(b) show the effect of increasing the problem size (N) and condition number (L/μ) respectively. As expected the NGM with global convexity

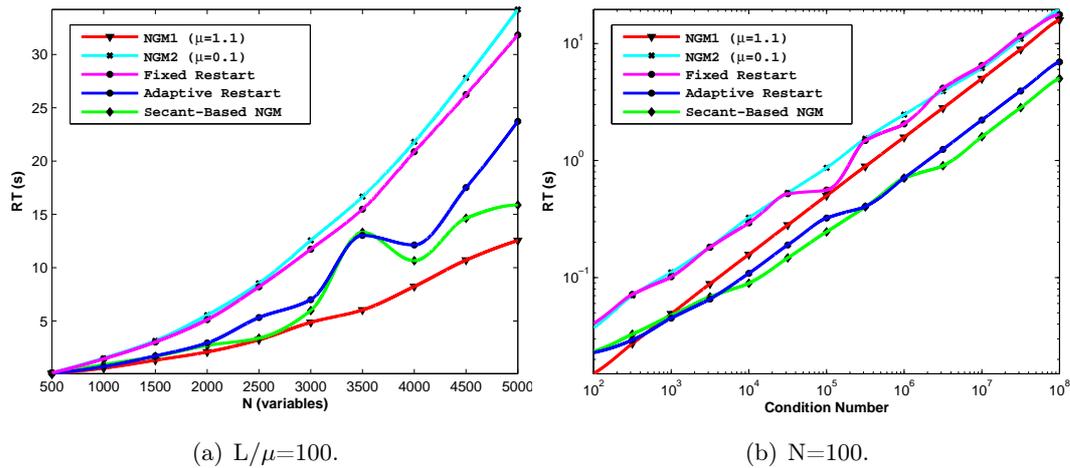


Figure 7.5: Effect of problem size (N) and condition number (L/μ) on the run-time. NGM1 - NGM with nontrivial μ ; NGM2 - NGM with global μ .

parameter $\mu = 0.1$ (NGM2) is slowest with the largest run-time (RT) as observed in Fig. 7.5(a). Using a fixed restart shows slightly improved performance with increasing dimension. However, the adaptive restart and **Simplified Secant-Based-NGM** perform significantly better than NGM2 and perform comparable with NGM1 as the dimension number increases. It can be noted in Fig. 7.5(b) that the adaptive restart and **Simplified Secant-Based-NGM** perform better than even the NGM1. Moreover, **Simplified Secant-Based-NGM** outperforms the adaptive restart as the condition number becomes high.

The simulations were repeated for randomly-generated matrices and vectors with seed `rng(5678, 'twister')`. The plots in Fig. 7.6(a) and Fig. 7.6(b) show the effect of increasing the problem size (N) and condition number (L/μ) respectively. Similar conclusions can

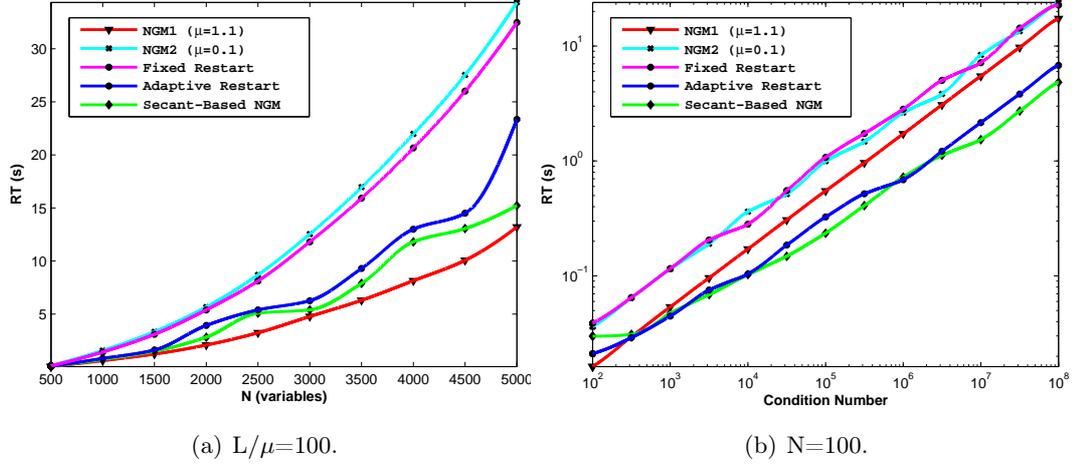


Figure 7.6: Effect of problem size (N) and condition number (L/μ) on the run-time. NGM1 - NGM with nontrivial μ ; NGM2 - NGM with global μ .

be drawn from the plots in Fig. 7.6(a) and Fig. 7.6(b). The performance profile [116] for all problem instances (i.e. with seed $rng(1234, 'twister')$ and seed $rng(5678, 'twister')$) is shown in Fig. 7.7(a). It is clear from Fig. 7.7(a) that the NGM with nontrivial convexity

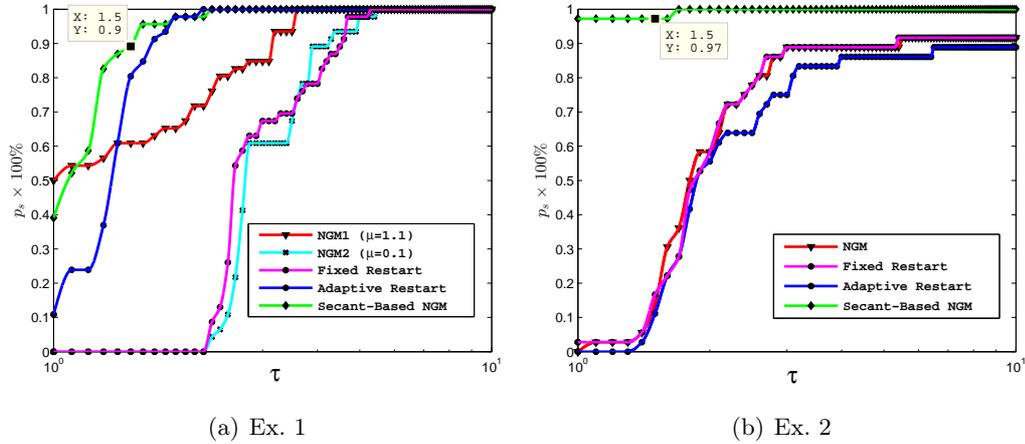


Figure 7.7: The performance profiles of the set of solvers \mathcal{S} for Ex. 1 and Ex. 2.

parameter has the most wins (i.e the highest probability of being the optimal solver). However, it can also be observed from Fig. 7.7(a) that the performance of NGM with a trivial convexity parameter is improved when a fixed restart or adaptive restart [25] is used. Moreover, **Simplified Secant-Based-NGM** performs better than when restarts are used with NGM. In general, the proposed **Simplified Secant-Based-NGM** has

the highest probability ($p_s(\tau)=0.9$) of being the fastest solver within a factor $\tau = 1.5$ of the best solver for that particular problem instance.

Due to the closeness of the run-times of some of the considered algorithms, some statistical analysis is required to account for the inevitable variability in timing. However, the statistical consideration is restricted to averaging the run-time over multiple-simulations. Fig. 7.8(a) and Fig. 7.8(b) show the mean run-time of 100 random simulations with increasing problem size (N) and condition number (L/μ) respectively.

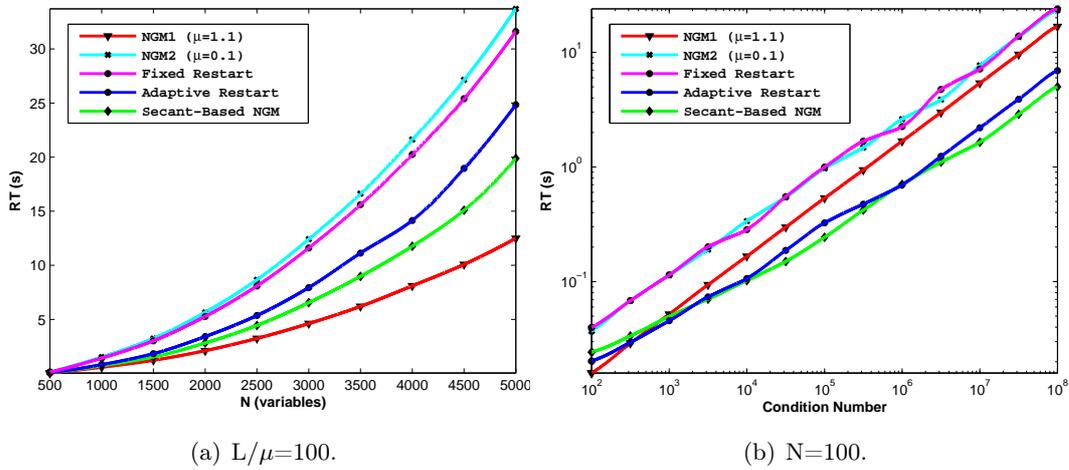


Figure 7.8: Effect of problem size (N) and condition number (L/μ) on the run-time. NGM1 - NGM with nontrivial μ ; NGM2 - NGM with global μ .

Ex. 2: Binary classification problem [136,137]

This is a logistic regression problem with l_2 -regularization. Given $z_i \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$, $\mu = 0$ and $N \geq 1$.

$$f(x) = \frac{\mu}{2} \|x\|_2^2 + \sum_{i=1}^N \log(1 + e^{-y_i z_i^T x}). \quad (7.50)$$

The objective function $f(x) \in \mathcal{S}_{\mu,L}^{1,1}$ is a non-quadratic convex function with Lipschitz gradient of $L = 0.25\lambda_{max}(F^T F)$ and global convexity parameter of $\mu = 0$, where $F = [z_1 \cdots z_N]^T \in \mathbb{R}^{N \times n}$. The design matrix F^T and the explained variable $y = \text{sign}(w^T F^T)$ were generated as described in [137,138] except that $w = [1; 1; 1 \cdots 1]$. The choice of w

means that each feature has an equal effect on the explained variable y .

The plots in Fig. 7.9(a) and Fig. 7.9(b) show the effect of increasing the problem size (N) and condition number (L/μ) respectively. The simulations were repeated for

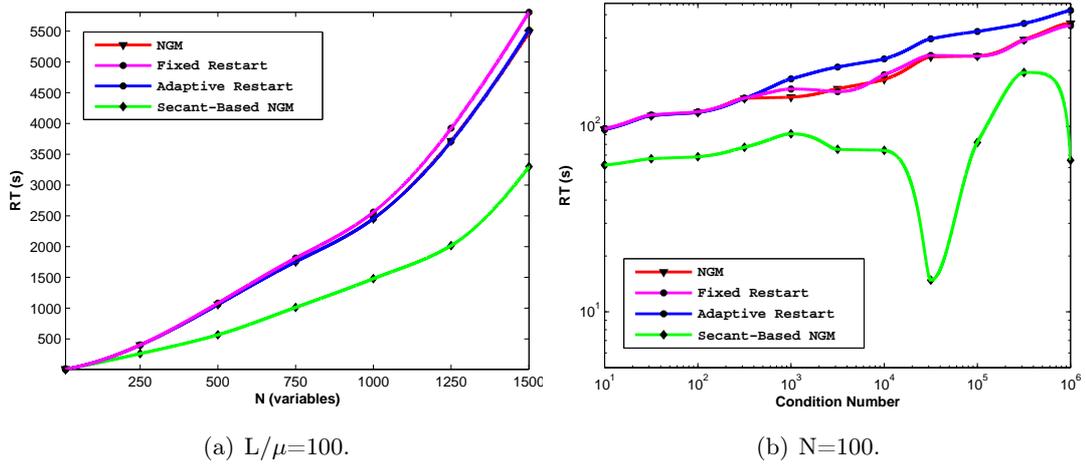


Figure 7.9: Effect of problem size (N) and condition number (L/μ) on the run-time.

randomly-generated matrices and vectors with seed `rng(5678, 'twister')`. The plots in Fig. 7.10(a) and Fig. 7.10(b) show the effect of increasing the problem size (N) and condition number (L/μ) respectively. Same conclusions can be drawn from the plots

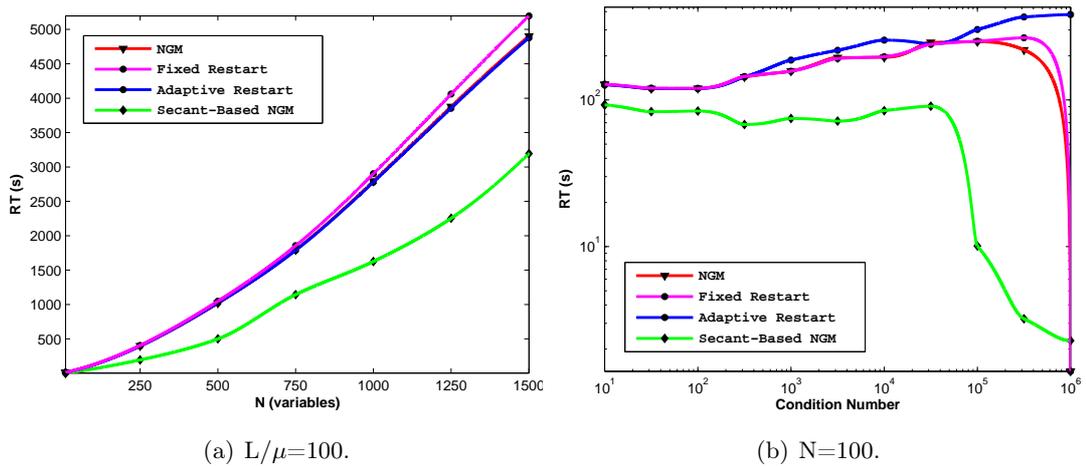


Figure 7.10: Effect of problem size (N) and condition number (L/μ) on the run-time.

in both Fig. 7.10(a) and Fig. 7.10(b). It can be observed in these figures that all the

algorithms have similar performance except for **Simplified Secant-Based-NGM** that clearly outperforms the rest as the dimension number increases. In Fig. 7.10(b), the adaptive restart performs worse than the NGM while the **Simplified Secant-Based-NGM** clearly significantly outperforms all other algorithms as the condition number increases.

The performance profile [116] for all problem instances is shown in Fig. 7.7(b). It is clear from Fig. 7.7(b) that **Simplified Secant-Based-NGM** is the most efficient of the considered solvers. It solved 97% of the problems significantly faster than the other solvers. It can also be observed from Fig. 7.7(b) that the performance of adaptive restart [25] is worse than the fixed restart or the classical NGM. In general, the proposed **Simplified Secant-Based-NGM** significantly improves over and above the classical NGM and the adaptive restart suggested in [25].

7.7 Conclusion

The Nesterov gradient method needs to be restarted (e.g. [25, 130]) when a nontrivial convexity parameter is not available. This chapter introduces a new secant-based Nesterov gradient method (**Secant-Based-NGM**) and also establishes that it is globally convergent for all convex functions. The algorithm only requires a trivial lower bound of the convexity parameter $\mu \geq 0$ and the gradient's Lipschitz constant L . The efficiency of the proposed algorithm derives from updating the estimate-sequence parameter γ_k by imposing a secant condition on the choice of search point y_k . Furthermore, the proposed **Secant-Based-NGM** embodies an "update rule with reset" that parallels the restart rule suggested in [25]. The effectiveness of the proposed algorithm is confirmed in numerical simulation of a few test functions with varying dimension and condition number. The proposed **Secant-Based-NGM** significantly improves the adaptive restart suggested in [25] and the classical NGM. Furthermore, Chapter 8, presents two new secant-based algorithms that exploit the discrete Lyapunov theorem to establish their global convergence.

Chapter 8

A Scaled Barzilai-Borwein Step-Size

8.1 Introduction

This chapter deals with unconstrained convex optimization from the control theory point of view. In particular, this chapter explores another subcategory of first-order methods called the Barzilai-Borwein gradient method. Let's consider the following unconstrained optimization of a general function f :

$$\text{UOP} : \quad \min_{x \in \mathbb{R}^n} f(x) \quad (8.1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous differentiable function. The domain of f , $\text{dom } f$, is the convex set \mathbb{R}^n . In general, we have the following subclasses of f as shown in Fig 8.1:

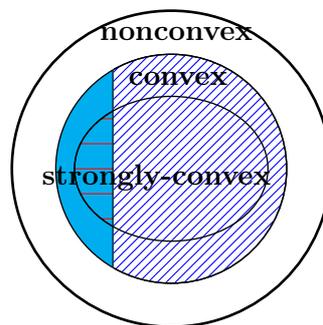


Figure 8.1: Subclasses of continuous differentiable function f .
▨ strongly-convex quadratic ■ convex quadratic ▩ convex non-quadratic □ non-convex

The above subclasses of general function f are defined precisely in section 8.2. If f is

convex, then a necessary and sufficient condition for a point x^* to be a minimizer of f is $\nabla f(x^*) = 0$ [85]. Furthermore, if f is bounded below, then there exists a unique minimum [86]. The UOP problem (8.1) is to be solved by an iterative algorithm which computes a sequence of iterates $x_0, x_1, \dots \in \mathbf{dom} f$ with $f(x_k) \rightarrow f^*$ as $k \rightarrow \infty$. The algorithm is terminated when $f(x_k) - f^* \leq \epsilon$, where $\epsilon > 0$ is some specified tolerance.

First-order methods are iterative algorithms that use only function evaluations and gradient information. They have received much attention due to their computational cheapness and low memory requirement. The Barzilai-Borwein (BB) gradient method is a subcategory of such first-order methods and does not guarantee a monotone decrease in the objective function.

The BB gradient method [109] has been shown to be R-linear globally convergent for strongly-convex quadratic functions  [139] and globally convergent for convex quadratic functions  [140]. However, the BB gradient method is not globally convergent for general function f . Goh et al [33], Leong et al [34] addressed the non-monotonicity "drawback" of the BB method by enforcing monotonic descent in order to ensure global convergence for general function f . Nevertheless, it has been experimentally observed by [124, 141] that the potential effectiveness of the BB method was related to the relationship between the step-size and the eigenvalues of the Hessian rather than to the decrease of the function value.

Moreso, several researchers [29, 31, 35, 125, 142–144] have also pointed out that non-monotone line-search schemes may considerably improve the rate of convergence, in particular, cases where a monotone line-search scheme is forced to creep along the bottom of steep-sided valleys (e.g. ill-conditioned functions). Nonmonotone techniques have been distinguished by the fact that they do not enforce strict monotonicity to the objective function values at successive iterates. Most nonmonotonic line-searches to date are still a form of relaxed-descent conditions that satisfy an Armijo-like inequality (e.g. [29, 31, 35, 125, 142, 144]).

Dai et al [32, 145] have proposed an adaptation of the nonmonotone line-search [143] to

the BB gradient method to establish global convergence for general function f . Raydan [26] incorporated the nonmonotone line-search of [142] in the BB gradient method to ensure global convergence for general function f . Modified-BB gradient algorithms that incorporate a nonmonotone line-search in order to guarantee global convergence of general function f are called Globalized BB gradient methods (GGB). The variants of the globalized BB gradient method introduced by [26] have appeared in subsequent works of [27, 28, 30, 35]. These globalized BB gradient methods as reported in the literature still require relaxed-descent conditions that satisfy an Armijo-like inequality,

$$f(x_k + \alpha_k d_k) \leq R_k + \alpha_k c_1 \nabla f(x_k)^T d_k. \quad (8.2)$$

where R_k represent some reference scalar value. There are several proposals for R_k in the literature (see Section 8.3.1). As argued by Fletcher [124], "*... the use of the line search technique of Grippo, Lampariello and Lucidi [142] in the manner proposed by Raydan [26] may not be the best way of globalizing the BB method, ...*".

In this chapter, new modified-BB gradient methods that do not involve any inexact line-search are proposed and they are shown to be globally convergent for all convex functions. In other words, the algorithms are globally convergent for functions included in the hatched region  of Fig. 8.1. Moreover, the algorithms do not enforce monotonicity to the objective function values at successive iterates. The **Scaled-BBGM** only requires a single gradient evaluation per iteration. Furthermore, the **Hybrid-BBGM** accelerates the **Scaled-BBGM** by interleaving scaled-BB step-sizes with BB step-sizes. Both proposed algorithms **Scaled-BBGM** and **Hybrid-BBGM** are shown to be globally convergent for all convex functions.

The rest of this chapter is organized as follows: In sections 8.3 and 8.4, the proposed modified-BB methods are described. The global convergence of the proposed methods for all convex functions will also be established therein and numerical results are reported in sections 8.5 and 8.8 respectively.

8.2 Notation

The notation is same as used in Chapter 7. Furthermore, the notation $h \in C[\mathbb{R}^+, \mathbb{R}^+]$ denotes a function h continuous on $\mathbb{R}^+ = [0, \infty)$ and $h : \mathbb{R}^+ \rightarrow \mathbb{R}^+$.

8.3 The Barzilai-Borwein step-size

The iterative algorithm

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k). \quad (8.3)$$

computes a sequence of iterates $x_0, x_1, \dots \in \mathbf{dom} f$. This algorithm converges with $f(x_k) \rightarrow f^*$ as $k \rightarrow \infty$ if and only if $\{\nabla f(x_k)\} \rightarrow 0$. The choice of α_k corresponds to the subcategory of gradient method. The BB step-size corresponds to the BB gradient method [109]. The BB gradient method can be arrived at in various ways. This is can be done as follows:

1. One can interpret the BB step-size as scalar Quasi-newton update. Recall from Chapter 7 (pg. 110) that the secant condition is given by,

$$B_{k+1} s_k = y_k. \quad (8.4)$$

Take B_k to be a scalar, b_k , and update using a scalar (symmetric rank-one update) as follows

$$b_{k+1} = b_k + u^T u. \quad (8.5)$$

Multiply both sides of (8.5) by $s_k^T s_k$ and use the secant condition (8.4), we have that

$$u^T u = \frac{y_k^T s_k - b_k s_k^T s_k}{s_k^T s_k}. \quad (8.6)$$

Substitute $u^T u$ in (8.5), we then have

$$\alpha_{k+1}^{BB1} = b_{k+1}^{-1} = \frac{s_k^T s_k}{y_k^T s_k}. \quad (8.7)$$

Similarly, one could obtain

$$\alpha_{k+1}^{BB2} = b_{k+1}^{-1} = \frac{y_k^T s_k}{y_k^T y_k}, \quad (8.8)$$

by multiplying both sides of (8.5) with $y_k^T y_k$.

2. Alternatively, the BB step-sizes [i.e. (8.7), (8.8)] are equivalent to $\min_{\alpha > 0} \{\|\alpha^{-1} s_k - y_k\|\}$ and $\min_{\alpha > 0} \{\|s_k - \alpha y_k\|\}$ respectively [146, 147].
3. Moreover for quadratic functions, the BB step-sizes [i.e. (8.7), (8.8)] can be derived from interpolation of the objective function as reported in [148]. Using quadratic and conic models respectively, the following step-sizes were obtained in [148],

$$\alpha_{k+1}^{DYY1} = \frac{s_k^T s_k}{2 \left(f(x_k) - f(x_{k+1}) + s_k^T \nabla f(x_{k+1}) \right)}, \quad (8.9)$$

$$\alpha_{k+1}^{DYY2} = \frac{s_k^T s_k}{6 \left(f(x_k) - f(x_{k+1}) \right) + 4s_k^T \nabla f(x_{k+1}) + 2s_k^T \nabla f(x_k)}. \quad (8.10)$$

It is straightforward to verify that (8.9) and (8.10) are equivalent to (8.7) if f is quadratic in the line segment between x_k and x_{k+1} [148].

In summary, the step-size of the BB gradient method is a scalar approximation of the Hessian B_{k+1} computed from the accumulated curvature information. It can also be shown that for convex functions $\frac{1}{L} \leq \alpha_k^{BB} \leq \frac{1}{\mu}$ where μ and L correspond to the minimum and maximum eigenvalues of the averaged Hessian matrix respectively [149]. Nevertheless, the Proposition 8.1 uses the fundamental definitions of μ and L to prove these bounds of the BB step-size for convex functions.

Proposition 8.1. *If f is convex with convexity parameter $\mu \geq 0$ and the ∇f is Lipschitz continuous with constant L , then the the BB step size is bounded with $\frac{1}{L} \leq \alpha_k^{BB} \leq \frac{1}{\mu}$.*

Proof. Let's consider the BB step-size (8.7). Using the identity $a^T b \leq \|a\| \|b\|$, we have that

$$(x_{k+1} - x_k)^T (\nabla f(x_{k+1}) - \nabla f(x_k)) \leq \|x_{k+1} - x_k\| \|\nabla f(x_{k+1}) - \nabla f(x_k)\|. \quad (8.11)$$

Since ∇f is Lipschitz continuous with constant L , i.e $\|\nabla f(x_{k+1}) - \nabla f(x_k)\| \leq L\|x_{k+1} - x_k\|$, it follows from (8.11)

$$(x_{k+1} - x_k)^T (\nabla f(x_{k+1}) - \nabla f(x_k)) \leq L\|x_{k+1} - x_k\|^2. \quad (8.12)$$

Also it follows from the definition of strong-convexity that

$$(x_{k+1} - x_k)^T (\nabla f(x_{k+1}) - \nabla f(x_k)) \geq \mu\|x_{k+1} - x_k\|^2. \quad (8.13)$$

Thus, using (8.12) and (8.13), we obtain that

$$\frac{1}{L} \leq \frac{s_k^T s_k}{y_k^T s_k} \leq \frac{1}{\mu}. \quad \blacksquare$$

It is interesting to note that for convex quadratic functions, the BB step-sizes (8.7) and (8.8) are equivalent to the previous Cauchy and minimum-gradient step-sizes respectively.

$$\alpha_k^{BB1} = \alpha_{k-1}^{SD} = \frac{d_{k-1}^T d_{k-1}}{d_{k-1}^T H d_{k-1}}, \quad (8.14)$$

$$\alpha_k^{BB2} = \alpha_{k-1}^{MG} = \frac{d_{k-1}^T H d_{k-1}}{d_{k-1}^T H^T H d_{k-1}}. \quad (8.15)$$

The alternate step (AS) method [150] is a particular case of j -cyclic steepest descent method introduced in [151] for convex quadratic functions. It is a hybrid gradient method which combines the Cauchy step and the BB step as follows:

$$\alpha_k^{AS} = \begin{cases} \alpha_k^{SD} & \text{if } k \text{ is odd,} \\ \alpha_k^{BB} & \text{if } k \text{ is even.} \end{cases} \quad (8.16)$$

8.3.1 GLOBALIZED BARZILAI-BOWEIN GRADIENT METHOD

As already noted in the introduction of this chapter, the BB gradient method (8.7), (8.8) is globally convergent for convex quadratic functions. The globalized BB-gradient method ensures convergence for convex non-quadratic functions by satisfying a non-monotonic line-search condition. Most nonmonotonic line-searches to date are still a form of relaxed-descent conditions that satisfy an Armijo-like inequality:

$$f(x_k + \alpha_k d_k) \leq R_k + \alpha_k c_1 \nabla f(x_k)^T d_k. \quad (8.17)$$

where R_k represent some reference scalar value. There are several proposals for R_k in literature: :

1. GLL: $R_k = \max_{0 \leq j \leq m_k} f(x_{k-j})$: maximum of a finite recent function values decreases as reported in [142],
2. ZHL: $R_{k+1} = \kappa_k R_k + (1 - \kappa_k) f(x_{k+1})$: average of successive function values decreases as reported in [29],
3. ANL: $R_k = \kappa_k \max_{0 \leq j \leq m_k} f(x_{k-j}) + (1 - \kappa_k) f(x_k)$: a convex combination variant reported in [35] and m_k is a non-decreasing integer, bounded by a fixed integer M .

Also $c_1, \kappa_k \in (0, 1)$.

One major drawback of this kind of nonmonotonic line-searches is that for a particular choice of M , the acceptable $\{\alpha_k\}_{k \in K}$ may be so small that the globalized BB gradient method produces a stagnating subsequence $\{x_k\}_{k \in K} \subset \{x_k\}$. In addition, the numerical performance of globalized BB gradient methods are sensitive to the choice of M and it has been reported in [26, 27, 152] to perform very well for values of $M \in (5, 20)$. In general the heuristic choice of M is problem-dependent [26, 29, 153]. To overcome these drawbacks, a modified-BB gradient method is presented in the next section. The modified-BB gradient method is globally convergent for all convex functions. Moreover, the modified gradient method is independent of any line-search and does not enforce monotonicity in the objective function values.

8.4 A scaled-BB step-size: Scaled-BBGM

Consider the iterative algorithm

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \quad (8.18)$$

where $\nabla f(x_k)$ is the gradient of general function f . It has been shown that the algorithm (8.18) can be cast in a control framework [154, 155] where α_k is a control parameter to be chosen dynamically in a feedback form. From a control-theoretic perspective, (8.18) is equivalent to Fig. 8.2 below.

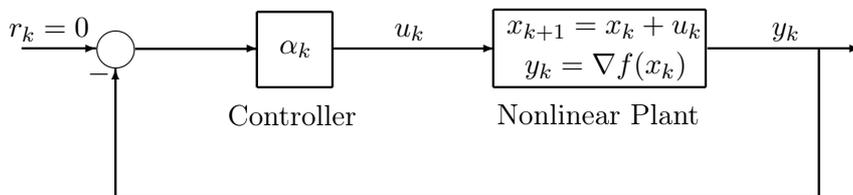


Figure 8.2: The Gradient Method.

In the case of a convex quadratic function, the closed-loop system in Fig. 8.2 becomes

$$x_{k+1} = (I - \alpha_k H)x_k. \quad (8.19)$$

Suppose that $\alpha_k = \alpha$, then the closed-loop system in Fig. 8.2 is stable if all the eigenvalues of $(I - \alpha H)$ lie inside the open unit circle on the z -plane. This implies that $\alpha < 2/L$ for the gradient method to converge to the optimal solution. However, stability of Fig. 8.2 in the general case (i.e. 8.18) can be established using the following corollary of the discrete Lyapunov theorem [43, 44] (see Chapter 2):

Corollary 8.1. *Let x^* be the equilibrium point of the closed-loop system (8.18). Assume that $\nabla f(x_k)$ is locally Lipschitz and let $V : \mathbb{R}^n \rightarrow \mathbb{R}$ be a scalar-valued continuous differentiable function defined on \mathbb{R}^n . Define $\Delta V(x_k) := V(x_{k+1}) - V(x_k)$. Suppose there exists V such that*

$$(i) \quad V(x^*) = 0,$$

(ii) $V(x_k) > 0$ for all $x_k \neq x^*$ in \mathbb{R}^n ,

(iii) $V(x) \rightarrow \infty$ as $\|x\| \rightarrow \infty$,

then the equilibrium point x^* of the closed-loop system (8.18) is

- globally stable if $\Delta V(x_k) \leq 0$ for all x_k in \mathbb{R}^n .
- globally asymptotically stable if $\Delta V(x_k) < 0$ for all $x_k \neq x^*$ in \mathbb{R}^n . ■

In the case of convex quadratic functions, the use of $V(x) = g_k^T g_k$ and $V(x) = g_k^T H^{-1} g_k$ can be used to establish the global convergence of the minimum gradient method and the steepest gradient-descent method respectively [156]. For the general case of convex functions, Corollary 8.1 is subsequently used to establish global convergence of the gradient method (8.18) for a particular interesting choice of step-size α_k .

Theorem 8.1. *Given a convex continuous differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Suppose $g_k = \nabla f(x_k)$ is locally Lipschitz with constant L . Let $\alpha_{k+1} = \alpha_k \frac{\|g_k\|_2^2}{\|g_{k+1}\|_2^2 - g_k^T \Delta g_k}$ and α_0 be such that $\alpha_1 \not\approx 0$. Then the closed-loop system (8.18) with this choice of α_{k+1} is globally asymptotically stable with the $\lim_{k \rightarrow \infty} \alpha_k \|g_k\|_2^2 \rightarrow 0$ at an exponential rate of $\eta_k \in (0, 1)$. Furthermore, we have that the $\lim_{k \rightarrow \infty} \|g_k\|_2 \rightarrow 0$.*

Proof:

Choose the Lyapunov function candidate be such that $V(x_k) = \alpha_k g_k^T g_k$. Then conditions (i), (ii) and (iii) of Corollary 8.1 are satisfied. Subsequently, it is shown that $\Delta V(x_k) < 0$ for all $x_k \neq x^*$ in \mathbb{R}^n . Note that $-g_k^T \Delta g_k > 0$ for convex functions. Suppose

$$0 < \alpha_{k+1} \leq \alpha_{k+1}^{BB1} \quad \text{i.e.} \quad 0 < \alpha_{k+1} \leq \alpha_k \frac{g_k^T g_k}{-g_k^T \Delta g_k}. \quad (8.20)$$

Then $\Delta V(x_k)$ is given by $\Delta V(x_k) = \alpha_{k+1} g_{k+1}^T g_{k+1} - \alpha_k g_k^T g_k$. This can be written as

$$\Delta V(x_k) = \alpha_{k+1} (g_k + \Delta g_k)^T (g_k + \Delta g_k) - \alpha_k g_k^T g_k, \quad (8.21)$$

where $\Delta g_k = g_{k+1} - g_k$. Expanding (8.21), $\Delta V(x_k)$ becomes

$$\Delta V(x_k) = \alpha_{k+1}(g_k^T g_k + g_k^T \Delta g_k + \Delta g_k^T \Delta g_k) + \alpha_{k+1}g_k^T \Delta g_k - \alpha_k g_k^T g_k. \quad (8.22)$$

Using $\Delta g_k = g_{k+1} - g_k$, (8.22) reduces to

$$\Delta V(x_k) = \underbrace{\alpha_{k+1}(g_{k+1}^T g_{k+1} - g_k^T \Delta g_k)}_A + \underbrace{\alpha_{k+1}g_k^T \Delta g_k - \alpha_k g_k^T g_k}_B. \quad (8.23)$$

Since $0 < \alpha_{k+1} \leq \alpha_k \frac{g_k^T g_k}{-g_k^T \Delta g_k}$ by supposition, then $-2\alpha_k g_k^T g_k < B < -\alpha_k g_k^T g_k$ and it follows that

$$\Delta V(x_k) = \alpha_{k+1}(g_{k+1}^T g_{k+1} - g_k^T \Delta g_k) - \alpha_k g_k^T g_k - \eta_k \alpha_k g_k^T g_k, \quad (8.24)$$

for some $\eta_k \in (0, 1]$. The obtained $\Delta V(x_k)$ (8.24) is demonstrated by the graphical interpretation of (8.23) in Fig. 8.3. The lines in Fig. 8.3 corresponds to A and B designated in (8.23).

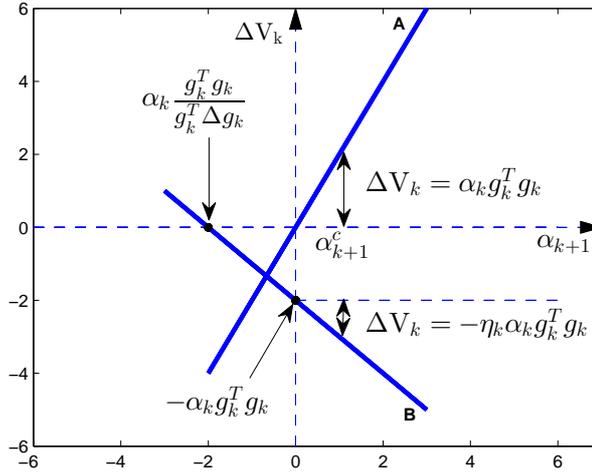


Figure 8.3: Graph of $\Delta V(x_k)$ against α_{k+1} .

ΔV_k denotes $\Delta V(x_k)$ and α_{k+1}^c corresponds to the proposed step-size. At α_{k+1}^c , the summation of $\Delta V(x_k)$ is less than 0.

Therefore, it is sufficient to choose $\alpha_{k+1} = \alpha_{k+1}^c$ i.e.

$$\alpha_{k+1} = \alpha_{k+1}^c = \alpha_k \frac{g_k^T g_k}{g_{k+1}^T g_{k+1} - g_k^T \Delta g_k} \quad (8.25)$$

to ensure that $\Delta V(x_k) < 0$ for all $x_k \neq x^*$ in \mathbb{R}^n . It remains to note that since f is convex, then $-g_k^T \Delta g_k > 0$ and the supposition $0 < \alpha_{k+1} \leq \alpha_k \frac{g_k^T g_k}{-g_k^T \Delta g_k}$ holds. Thus, the choice of α_{k+1} in (8.25) guarantees that the closed-loop system (8.18) is globally asymptotically stable.

Furthermore, with the choice of α_{k+1} (8.25), then $\Delta V(x_k)$ reduces to

$$\Delta V(x_k) = -\eta_k V(x_k). \quad (8.26)$$

Remark 8.1. *The scalar $\eta_k \in (0, 1)$ since $\alpha_{k+1} = \alpha_k \frac{g_k^T g_k}{-g_k^T \Delta g_k}$ if and only if $x_{k+1} = x^*$.*

It follows from (8.26) that $V(x_{k+1}) = (1 - \eta_k)V(x_k)$. Consequently, we have

$$\alpha_k \|g_k\|_2^2 = \alpha_0 (1 - \eta_k)^k \|g_0\|_2^2, \quad (8.27)$$

i.e. $\alpha_k \|g_k\|_2^2$ monotonically decreases to 0. This implies that $\lim_{k \rightarrow \infty} \alpha_k \|g_k\|_2^2 \rightarrow 0$. Since we have $0 < \alpha_{k+1} \leq \alpha_{k+1}^{BB1}$, thus α_{k+1} is bounded away from zero and we have that therefore $\lim_{k \rightarrow \infty} \|g_k\|_2 \rightarrow 0$. ■

Remark 8.2. *The proof of Theorem 8.1 guarantees that $\alpha_k \|g_k\|_2^2$ decreases monotonically. It is to be emphasized that $\|g_k\|_2^2$ does not necessarily decrease monotonically.*

Remark 8.3. *Note that as $\lim_{k \rightarrow \infty} \|g_k\|_2^2 \rightarrow 0$, the step-size (8.25) tends to α_{k+1}^{BB1} and as such the proposed step-size (8.25) can be interpreted as a scaled BB step-size.*

Algorithm 8.1 (Scaled-BBGM). The outline is as follows:

Scaled-BBGM.

Given a starting point $x_0 \in \text{dom } f$ and $\alpha_0 > 0$.

repeat until stopping criterion is satisfied

1. Update: $x_{k+1} = x_k - \alpha_k g_k$.
2. Update: $\alpha_{k+1} = \alpha_k \frac{\|g_k\|_2^2}{\|g_{k+1}\|_2^2 - g_k^T \Delta g_k}$. \dots **scaled BB step-size**

end (repeat)

Remark 8.4. *If $\alpha_1 \approx 0$, the **Scaled-BBGM** might stagnate. The choice of $\alpha_0 = \frac{1}{\|g_k\|}$ ensures that $\alpha_1 \not\approx 0$ and this corresponds to taking a unit step in the negative gradient direction at the first iterate.*

8.5 Simulation Examples I

Consider two test examples of the form

$$\text{UOP} : \quad \min_{x \in \mathbb{R}^n} f(x), \quad (8.28)$$

where $x \in \mathbb{R}^n$. The numerical tests investigate the effects of increasing the dimension and condition number respectively on the performance of the proposed algorithm.

8.5.1 Simulation Setup

The simulation setup is same as in Chapter 7. The stopping criterion is $\|\nabla f(x)\| \leq 10^{-6}$ for Ex. 1 and 2 respectively. All matrices are square and random such that the Hessian has eigenvalues in the interval $[1, L]$. This was achieved using singular value decomposition to allocate the desired eigenvalues. All matrices and vectors were randomly generated in MATLAB with seed `rng(1234, 'twister')` unless otherwise stated. The simulations were also repeated for seed `rng(5678, 'twister')`.

8.5.2 Test Functions and Solvers

The test function in Ex. 1 is a convex non-quadratic function introduced by Raydan [26] while the test function in Ex. 2 is also a convex non-quadratic function usually encountered in machine-learning literature. In all cases and examples, all

line-searches implement one-dimensional quadratic interpolation ($[0.1, 0.9]$) with a bisection safeguard (see [26, 148]). Also, the step-size in the line-searches are bounded as $\min\{10^{30}, \max\{10^{-30}, \alpha_k\}\}$. In all cases and examples, $\alpha_0 = \frac{1}{\|g_k\|}$. The set of solvers \mathcal{S} considered in Ex. 1 is

- GLL+BB : GLL [142]+BB step (α_{k+1}^{BB1}),
- GLL+DYY : GLL [142]+DYY step (α_{k+1}^{DYY2}),
- GLL+BKF : GLL [142]+BKF step [157],
- GLL+BS : GLL [142]+BS step [158],
- Proposed algorithm **Scaled-BBGM**,

where GLL denotes the "*Grippo, Lampariello, and Lucidi*" nonmonotonic line-search. The above Globalized BB gradient methods differ only in the initial step-size of the quadratic interpolation (i.e. BB, BKF and BS). In Ex. 1. the value of M is taken as: $M = 25$ if $N \leq 1000$ and $M = 100$ if otherwise, except for $N = 5000$ where $M = 120$.

8.5.3 Numerical Results 1

Ex. 1: Raydan's Strictly Convex "2" Function [26]

$$f(x) = \sum_{i=1}^n \frac{i}{10} (e^{x_i} - x_i), \quad x_0 = [1, 1, \dots, 1]^T. \quad (8.29)$$

The objective function $f(x) \in \mathcal{S}_{\mu, L}^{1,1}$ is a strictly convex function with convexity parameter $\mu = 0$ and whose gradient is not globally Lipschitz-continuous.

The plot in Fig. 8.4(a) show the effect of increasing the problem size (N). As expected the proposed **Scaled-BBGM** performs significantly better than other algorithms since it does not involve an inexact line-search. The numerical simulations were repeated for an

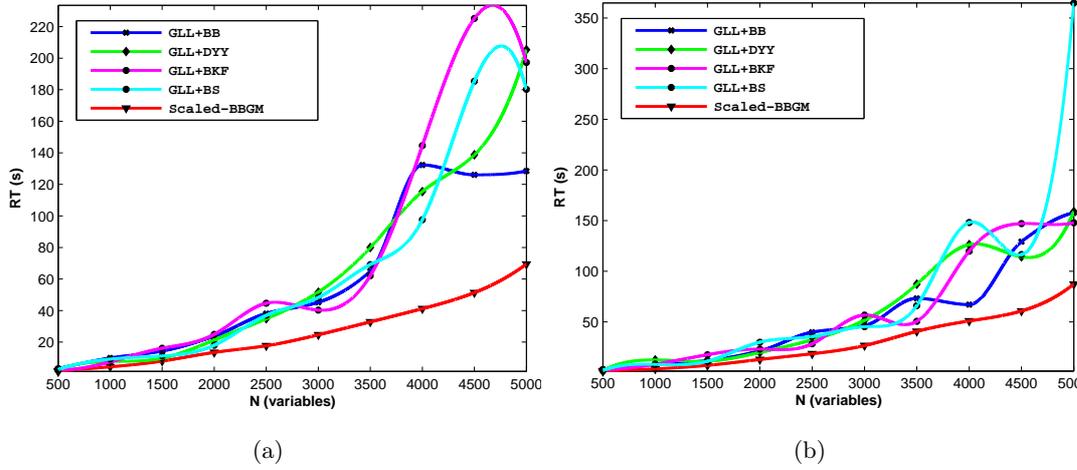


Figure 8.4: Effect of problem size (N) on the run-time.

initial-vector x_0 randomly-generated with seed $rng(1234, 'twister')$. The plot in Fig 8.4(b) show the effect of increasing the problem size (N). This plot in Fig. 8.4(b) also indicates that the **Scaled-BBGM** is the optimal solver among the considered set of solvers \mathcal{S} .

8.6 Nonmonotone Lyapunov Functions

Numerical examples indicate that **Scaled-BBGM** converges impressively especially if μ is 0 (and L does not exist). Nonmonotone Lyapunov Functions (LF) relaxes the monotonicity requirement of Lyapunov's theorem in order to obtain less conservative stability conditions. Nonmonotone LF given in [159] allows the Lyapunov functions to increase locally, but guarantee a weighted decrease every few steps. In particular, a generalization of Corollary 2.1 in [159] is established in Theorem 8.2.

Theorem 8.2. *Let x^* be the equilibrium point of the closed-loop system (8.1). Assume that $\nabla f(x_k)$ is locally Lipschitz and let $V : \mathbb{R}^n \rightarrow \mathbb{R}$ be a scalar-valued continuous differentiable function defined on \mathbb{R}^n . Let $\beta_k^{i=1:m} \in [0, \infty)$ be dynamic nonnegative scalars. Define $\Delta V(x_k) := V(x_{k+1}) - V(x_k)$. Suppose there exists V such that*

(i) $V(x^*) = 0$,

(ii) $V(x_k) > 0$ for all $x_k \neq x^*$ in \mathbb{R}^n ,

(iii) $V(x) \rightarrow \infty$ as $\|x\| \rightarrow \infty$,

(iv) $\beta_k^m \Delta V(x_{k+m}) + \dots + \beta_k^1 \Delta V(x_{k+1}) + \Delta V(x_k) < 0$ for all $x_k \neq x^*$ in \mathbb{R}^n

then the equilibrium point x^* is globally asymptotically stable.

Proof:

The fact that there exists β_k^i that satisfies (iv) implies that either of the corresponding $\Delta V(x_{k+m})$ should be strictly less than 0 i.e. the corresponding $V(x_{k+m})$ is strictly less than $V(x_k)$. Therefore, there exists a subsequence of $V(x_k)$ that is monotonically decreasing. Since the subsequence is lower bounded by zero, it must converge to some $c \geq 0$. It can be shown (e.g. by contradiction) that because of continuity of $V(x_k)$, then c must be zero. This part of the proof is similar to the proof of standard Lyapunov theory (see e.g. [48]). Now that a converging subsequence has been established, then for a given $0 \leq \beta_k^i \leq \tilde{\beta}$ and any $\epsilon > 0$, one can find \bar{k} such that $V(x_{\bar{k}}) \leq \{\frac{\epsilon}{1+\tilde{\beta}}, \frac{\tilde{\beta}\epsilon}{1+\tilde{\beta}}\}$. Because of positivity of $V(x_k)$ and (iv), we have $V(x_k) \leq \epsilon$ for $k \geq \bar{k}$. Therefore $V(x_k) \rightarrow 0$ and thus the equilibrium point x^* is globally asymptotically stable. ■

Remark 8.5. Furthermore, if $\beta_k^{i=1:m} = \infty$ then we must have the corresponding $V(x_{k+m}) - V(x_k) < 0$ for stability to be established. In this case, Theorem 8.2 reduces to the result given in [160].

A less conservative nonmonotone Lyapunov stability condition is given in [161]. The following theorem is adapted from [161] and the proof can be found in [161, 162].

Theorem 8.3. [161] Let x^* be the equilibrium point of the closed-loop system (8.18). Assume that $\nabla f(x_k)$ is locally Lipschitz and let $V : \mathbb{R}^n \rightarrow \mathbb{R}$ be a scalar-valued continuous differentiable function defined on \mathbb{R}^n . Define $\Delta V(x_k) := V(x_{k+1}) - V(x_k)$. Assume that there exists a function $h \in C[\mathbb{R}^+, \mathbb{R}^+]$ independent of x_k such that $h(0) = 0$. Let $\mathbb{E} = \{t_0, t_1, t_2, \dots, t_n\} \subset \mathbb{N}$ be discrete set with $t_p > t_q$ when $p > q$. Suppose there exists V such that

(i) $V(x^*) = 0$,

(ii) $V(x_k) > 0$ for all $x_k \neq x^*$ in \mathbb{R}^n

(iii) $V(x) \rightarrow \infty$ as $\|x\| \rightarrow \infty$,

(iv) $V(x_{t_{n+1}}) - V(x_{t_n}) < 0$ for all $x_{t_n} \neq x^*$ in \mathbb{R}^n , $t_n \in \mathbb{E}$,

(v) $V(x_{t_m}) < h(V(x_{t_n}))$ for all $t_n < m < t_{n+1}$, $m \in \mathbb{N}$, $t_n \in \mathbb{E}$,

then the equilibrium point x^* is globally asymptotically stable. ■

Remark 8.6. Theorem 8.3 requires the Lyapunov function to decrease monotonically only in an unbounded set \mathbb{E} of the time-instants. The last condition in the Theorem ensures that the Lyapunov function is reasonably bounded between the time constants not included in the set \mathbb{E} .

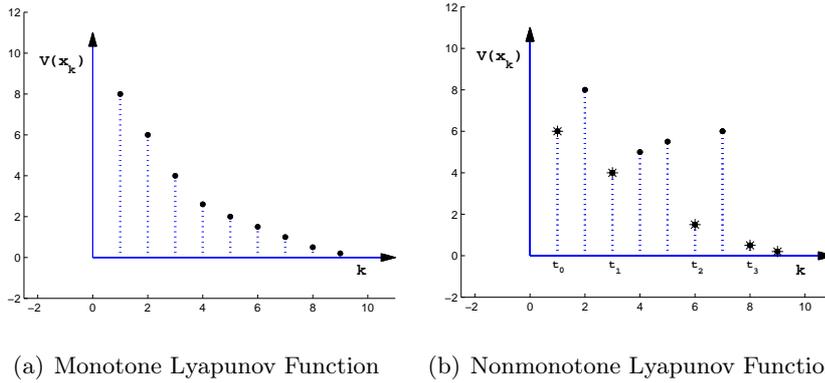


Figure 8.5: Plot of Lyapunov Functions.

The monotone Lyapunov function is required to decrease monotonically for all $k \in \mathbb{N}$ while the nonmonotone Lyapunov function [161] is required to decrease monotonically in an unbounded set $\{t_0, t_1, t_2, t_3 \dots\} \in \mathbb{E} \subseteq \mathbb{N}$.

8.7 A Hybrid BB Gradient Method: Hybrid-BBGM

In this section, a hybrid BB gradient method **Hybrid-BBGM** which interleaves the scaled BB step-size (α_k^{SBB}) with the BB step-size (α_k^{BB}) is constructed. Global convergence of the **Hybrid-BBGM** is established by using the sufficient conditions in Theorem 8.2. Allowing the LF to be nonmonotone further increases the amount of nonmonotonicity in the minimization process. Subsequently, the objective function f

is restricted to subclass $f \in \mathcal{S}_{\mu>0,L}^{1,1}$ i.e f is strongly-convex with Lipschitz-continuous gradient.

Algorithm 8.2 (Hybrid-BBGM). The outline is as follows:

Hybrid-BBGM.

Given a starting point $x_0 \in \mathbf{dom} f$, $\text{sum} = 0$, $\text{sumcheck} = -1$ and $\alpha_0 > 0$.

repeat until stopping criterion is satisfied

1. Update: $x_{k+1} = x_k + \alpha_k d_k$.
2. Update: $\alpha_{k+1} = \alpha_{k+1}^{BB1}$
3. Update: $\Delta V = \alpha_{k+1} \|g_{k+1}\|_2^2 - \alpha_k \|g_k\|_2^2$
4. if $t == 0$ && $\Delta V > 0$, $\text{sum} = \text{sum} + \Delta V$, $\text{sumcheck} = 1$, $t = 1$, $m_1 = 1$;
 elseif $m_1 == 1$ && $\Delta V > 0$, $\text{sum} = \text{sum} + \Delta V$, $\text{sumcheck} = 1$;
 elseif $m_1 == 1$ && $\Delta V < 0$, $\text{sum} = \text{sum} + \Delta V$, $\text{sumcheck} = \text{sum}$, $m_1 = 0$, $m_2 = 1$;
 elseif $m_2 == 1$ && $\Delta V < 0$, $\text{sum} = \text{sum} + \Delta V$, $\text{sumcheck} = \text{sum}$;
 elseif $m_2 == 1$ && $\Delta V \geq 0$, if $\alpha_{k+1}^{SBB} \geq \frac{2}{L}$, $\alpha_{k+1} = \alpha_{k+1}^{SBB}$. end;
 $\text{sum} = \text{sum} + \alpha_{k+1} \|g_{k+1}\|_2^2 - \alpha_k \|g_k\|_2^2$, $\text{sumcheck} = \text{sum}$, $m_1 = 0$, $m_2 = 1$;
 end
5. Reset: if $\text{sumcheck} < 0$, $\text{sum} = 0$, $t = 0$, $m_1 = 0$, $m_2 = 0$; end

end (repeat)

Remark 8.7. Step 4 of *Hybrid-BBGM* detects the local peak of the LF after time-instant $t_i \in \mathbb{E}$. Once the peak has been identified, then the algorithm takes a BB step if a BB step reduces the Lyapunov function. Otherwise, it takes a scaled BB step. However, if the scaled BB step is less than $2/L$, then a BB step is taken. This is continued until the value of the LF reduces below the value at time-instant t_i . This new time-instant is $t_{i+1} \in \mathbb{E}$.

8.7.1 GLOBAL CONVERGENCE

In this section, global convergence of the **Hybrid-BBGM** is established in Theorem 8.3.

The following Lemma is essential to the proof of Theorem 8.3.

Lemma 8.1. *Suppose f is convex and $g_k = \nabla f(x_k)$ is globally Lipschitz with constant L . If $\alpha_{k+1} = \alpha_k \frac{g_k^T g_k}{-g_k^T \Delta g_k}$, then $\alpha_{k+1} \|g_{k+1}\|_2^2 \leq (\kappa^2 + \kappa - 1) \alpha_k \|g_k\|_2^2$ where $\kappa = \frac{L}{\mu}$ and μ is the convexity parameter. If $\alpha_{k+1} = \alpha_k \frac{g_k^T g_k}{\|g_{k+1}\|_2^2 - g_k^T \Delta g_k}$, then $\alpha_{k+1} \|g_{k+1}\|_2^2 < \alpha_k \|g_k\|_2^2$.*

Proof:

The first part of the lemma is proved as follows: If $\alpha_{k+1} = \alpha_k \frac{g_k^T g_k}{-g_k^T \Delta g_k}$, then

$$g_k^T \Delta g_k = -\frac{\alpha_k g_k^T g_k}{\alpha_{k+1}}. \quad (8.30)$$

From the definition of convexity, we have

$$(x_{k+1} - x_k)^T (g_{k+1} - g_k) \geq \mu \|x_{k+1} - x_k\|^2, \quad (8.31)$$

and we can derive from (8.31) that

$$g_k^T g_{k+1} \leq (1 - \mu \alpha_k) g_k^T g_k \leq g_k^T g_k. \quad (8.32)$$

Also using the identity $a^T b \leq \|a\| \|b\|$ and the definition of Lipschitz continuity, we have

$$(g_{k+1} - g_k)^T (g_{k+1} - g_k) \leq L^2 \|x_{k+1} - x_k\|^2. \quad (8.33)$$

It follows from (8.33) that

$$\|g_{k+1}\|_2^2 \leq L^2 \alpha_k^2 \|g_k\|_2^2 + g_k^T \Delta g_k + g_k^T g_{k+1}. \quad (8.34)$$

Using (8.30) and (8.32), it follows from (8.34)

$$\alpha_{k+1}\|g_{k+1}\|_2^2 \leq L^2\alpha_{k+1}\alpha_k^2\|g_k\|_2^2 - \alpha_k g_k^T g_k + \frac{\alpha_{k+1}}{\alpha_k}\alpha_k g_k^T g_k. \quad (8.35)$$

Since $\frac{1}{L} \leq \alpha_k \leq \frac{1}{\mu}$ for all $k \geq 0$, it follows from (8.35)

$$\alpha_{k+1}\|g_{k+1}\|_2^2 \leq \left(\frac{L^2}{\mu^2} + \frac{L}{\mu} - 1\right)\alpha_k\|g_k\|_2^2. \quad (8.36)$$

The inequality (8.36) proves the first part of the Lemma. The proof of the second part of the Lemma is same as in Theorem 8.1 i.e. if $\alpha_{k+1} = \alpha_k \frac{g_k^T g_k}{\|g_{k+1}\|_2^2 - g_k^T \Delta g_k}$, then $\alpha_{k+1}\|g_{k+1}\|_2^2 < \alpha_k\|g_k\|_2^2$. ■

Remark 8.8. In general, if $\frac{1}{\epsilon} \leq \alpha_{k+1} \leq \alpha_k \frac{g_k^T g_k}{-g_k^T \Delta g_k}$ with $\epsilon > 0$, then $\alpha_{k+1}\|g_{k+1}\|_2^2 \leq (\kappa^2 + \frac{\epsilon}{\mu} - 1)\alpha_k\|g_k\|_2^2$.

Theorem 8.4. Given a convex continuous differentiable function $f \in \mathcal{S}_{\mu>0,L}^{1,1} : \mathbb{R}^n \rightarrow \mathbb{R}$. Let α_k be as constructed in **Hybrid-BBGM**. Suppose $g_k = \nabla f(x_k)$ is globally Lipschitz with constant L , then the closed-loop system (8.18) with the choice of this α_{k+1} is globally asymptotically stable with the $\lim_{k \rightarrow \infty} \|g_k\|_2 \rightarrow 0$.

Proof:

Choose the Lyapunov function candidate be such that $V(x_k) = \alpha_k g_k^T g_k$. Then conditions (i), (ii) and (iii) of Theorem 8.3 are satisfied. Condition (iv) is satisfied by construction of the algorithm. Furthermore by virtue of Lemma 8.1, condition (v) of Theorem 8.3 is satisfied with $h(s) = (\kappa^2 + \kappa - 1)s$, $s \in \mathbb{R}^+$ if $\alpha_k = \alpha_k^{BB}$ and $h(s) = s$, $s \in \mathbb{R}^+$ if $\alpha_k = \alpha_k^{SBB}$. Consequently, we have $\lim_{k \rightarrow \infty} \alpha_k\|g_k\|_2^2 \rightarrow 0$. Since α_{k+1} is bounded away from zero, therefore the closed-loop system (8.18) (i.e **Hybrid-BBGM**) is globally asymptotically stable with $\lim_{k \rightarrow \infty} \|g_k\|_2 \rightarrow 0$. ■

8.8 Simulation Examples II

The simulation setup is same as in § 8.5.1 and the test function is a convex non-quadratic function. In this example, the set of solvers \mathcal{S} considered is

- GLL+BB : GLL [142]+BB step (α_{k+1}^{BB1}),
- ZH+BB : ZHL [29]+BB step (α_{k+1}^{BB1}),
- AAN+BB : AAN [35]+BB step (α_{k+1}^{BB1}),
- Proposed algorithm **Hybrid-BBGM**,

where GLL denotes "*Grippo, Lampariello, and Lucidi*", ZH denotes "*Zhang and Hager*" and AAN denotes "*Amini, Ahookhosh and Nosratipour*" nonmonotonic line-search respectively. The above Globalized BB gradient methods differ only in the nonmonotonic line-search condition (i.e. GLL, ZH and AAN). In Ex. 2, the value of M is taken as :

$$M = \begin{cases} 20 & N \leq 500, \\ 50 & 500 < N < 2000, \\ 80 & 2000 \leq N < 2500, \\ 150 & N > 2500, \end{cases} \quad \text{and} \quad M = \begin{cases} 20 & \kappa \leq 10^3, \\ 50 & 10^3 < \kappa \leq 10^4, \\ 80 & 10^4 < \kappa < 10^6, \\ 150 & \kappa \geq 10^6, \end{cases}$$

where κ is the condition number.

8.8.1 Numerical Results 2

Ex. 2: Approximate Huber loss

Given $z_i \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$, $\mu = 1$ and $N \geq 1$.

$$f(x) = \frac{\mu}{2} \|x\|_2^2 + \sum_{i=1}^N \log(\cosh(y_i - z_i^T x)). \quad (8.37)$$

The objective function $f(x) \in \mathcal{S}_{\mu, L}^{1,1}$ is a non-quadratic convex function with Lipschitz gradient of $L = \lambda_{max}(F^T F)$ and global convexity parameter of $\mu = 1$, where $F =$

$[z_1 \cdots z_N]^T \in \mathbb{R}^{N \times n}$. The design matrix F^T and the explained variable $y = \text{sign}(w^T F^T)$ were generated as described in [137, 138] except that $w = [1; 1; 1 \cdots 1]$. The choice of w means that each feature has an equal effect on the explained variable y .

The plots in Fig. 8.6(a) and Fig. 8.6(b) show the effect of increasing the problem size (N) and condition number (L/μ) respectively. The run-time of the **Scaled-BBGM** has been scaled by 0.25 and simulation stopped once the run-time exceeds 5000s. In any case, the **Scaled-BBGM** performs miserably for this strongly-convex example. However, its performance is significantly improved by hybridizing it with BB steps as indicated for the case of **Hybrid-BBGM**. It is clear from Fig. 8.6(a) that as the dimension increases, the proposed **Hybrid-BBGM** becomes more significantly efficient than any of the considered Globalized BB methods. Furthermore, as shown in Fig. 8.6(b), the performance of the **Hybrid-BBGM** is significantly better than all other algorithms.

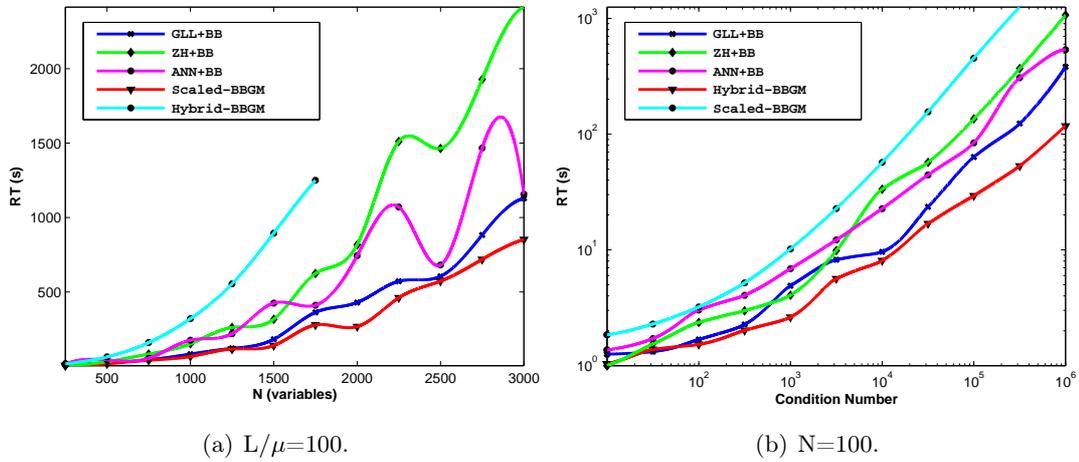


Figure 8.6: Effect of problem size (N) and condition number (L/μ) on the run-time.

The numerical simulations were repeated for randomly-generated matrices and vectors with seed `rng(5678, 'twister')`. The plots in Fig. 8.7(a) and Fig. 8.7(b) show the effect of increasing the problem size (N) and condition number (L/μ) respectively. Similar

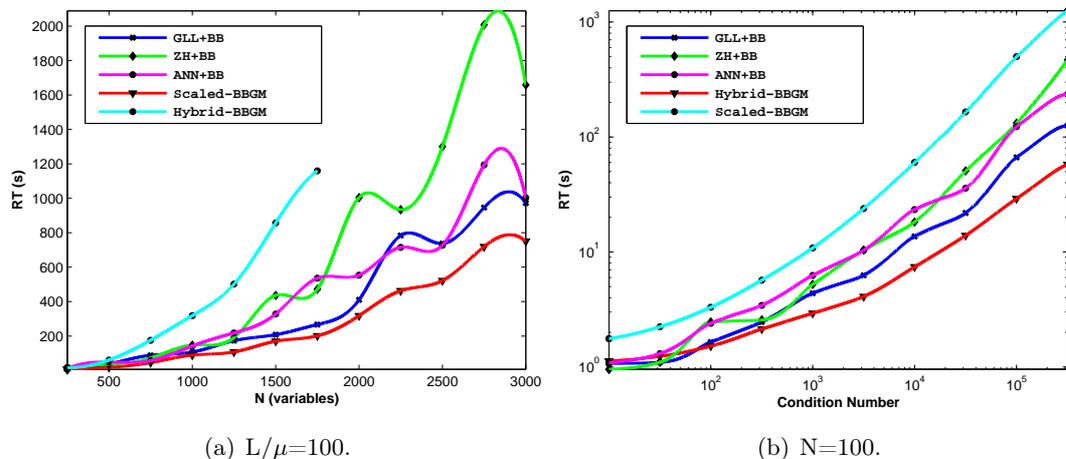


Figure 8.7: Effect of problem size (N) and condition number (L/μ) on the run-time.

conclusions can be drawn from both plots in Fig. 8.7 about the effectiveness of the proposed **Hybrid-BBGM** for strongly-convex functions .

8.9 Conclusion

This chapter has introduced two modified-BB gradient methods - **Scaled-BBGM** and **Hybrid-BBGM**. Neither of the proposed algorithms requires any inexact line-search and both are globally convergent for all convex functions. The effectiveness of the algorithms are confirmed in numerical simulation of a few test functions with varying dimension and condition number. For convex functions with convexity parameter $\mu = 0$ and non-Lipschitz gradient, numerical simulations indicates that the **Scaled-BBGM** performs significantly better than Globalized BB gradient methods [26, 148, 157, 158]. Furthermore, if the objective function is strongly-convex with Lipschitz-continuous gradient, then the **Hybrid-BBGM** is significantly more efficient than the Globalized BB gradient methods reported in [26, 29, 35]. However, the proposed algorithms are distinguished from the Globalized BB gradient methods in that they do not require any inexact line-search or specifying any problem-dependent parameter such as M. Moreover, the efficiency of the proposed algorithms derives from the proper choice of Lyapunov function $V(x_k)$.

Chapter 9

Conclusion and Recommendations

9.1 Conclusion

This thesis have explored the subject matter of online-optimizing anti-windup control. The contribution to this field is two-fold: analysis and implementation. The first part of the thesis establishes a general framework for analyzing robust preservation in anti-windup control systems. This framework - the robust Kalman conjecture - has been verified for first-order SISO plants. Three main results were obtained regarding this verification for the cases of additive uncertainty, input-multiplicative uncertainty and feedback uncertainty. The results obtained indicate that the robust interval of an uncertain first-order plant coincides with the nonlinearity's slope interval for which the robust Lur'e structure is absolutely stable. The saturation nonlinearity is an example of such nonlinearities whose slope interval is bounded. Thus, anti-windup control systems that can be reduced to a first-order Lur'e structure has the same measure of robustness as their unconstrained counterpart.

Furthermore, this thesis presented three secant-based unconstrained convex optimization algorithms. The simplified secant algorithm (**Secant-Based-NGM**) is based on updating the estimate-sequence parameter of Nesterov gradient method with secant information whenever possible. This was achieved by imposing a secant condition on the choice of search point. The **Scaled-BBGM** is a scaled Barzilai-Borwein gradi-

ent method without line search that is shown to be globally convergent for all convex functions. Finally, the **Hybrid-BBGM** accelerates the scaled-BB gradient method by relaxing the monotonicity requirement of the discrete Lyapunov theorem. Numerical examples (e.g. logistic and least-square losses) were used to demonstrate the effectiveness of the proposed algorithms and the numerical results obtained were analyzed with the aid of performance profiles.

9.2 Recommendations

This section highlights potential areas of future research.

9.2.1 Research Direction 1

The robust Kalman conjecture (RKC) has been verified true for first-order plants. Since the Kalman conjecture is not valid beyond third-order plants, it would be interesting to verify if the robust variant holds for second-order and third-order plants. Such analysis would enlarge the class of anti-windup control systems for which robust preservation holds or otherwise. Moreover, such theoretical work would provide more insights for robustness analysis in general. Analyzing the RKC for 2nd or 3rd order plants should preferably be done on a subclass basis as it is very likely that a single stability-multiplier class (e.g. constant, Popov and Zames-Falb multipliers) might not be able to establish the RKC for the entire class of 2nd or 3rd order plants.

9.2.2 Research Direction 2

The projected gradient method, allow more substantial rapid changes to the working set by choosing a descent direction Δx and searching along the piecewise linear path $\mathcal{P}(x - \alpha \Delta x)$, where $\alpha > 0$ and \mathcal{P} is the projection onto a feasible set. It is most efficient when the constraints are simple in form, in particular, when there are only bounds on the variables (box constraints). Richter et al [20,21] provided certification guarantees to real-time model predictive control (MPC) applications based on projected fast gradient

methods [23]. This has renewed interest in hardware and software implementations of projected fast gradient methods (see [118, 163–165]) as viable alternative algorithms for solving bound-constrained optimization problems typically present in the control community. Consequently, further research into increasing the efficiency of unconstrained gradient methods is a recipe for improved practical implementations of large-scale optimization algorithms in MPC and related applications.

9.2.3 Research Direction 3

The nonlinear conjugate gradient method (CGM) variants for non-quadratic convex functions are no longer optimal due to the degradation of the conjugate property of the gradients. Nonlinear CGM variants do not only employ a Wolfe-type line-search, but must constrain the step-size α_k such that d_{k+1} is a descent direction [166]. The nonlinear CGM reported in [167] remains the most effective/robust variant and ensures sufficient descent independent of the accuracy of the line search. An important feature of these scheme which distinguishes it from the other schemes is that with their β_k^{ZH} choice, d_{k+1} is always a descent direction for any step-size $\alpha_k > 0$, as long as $d_k y_k \neq 0$, where $y_k = g_{k+1} - g_k$. A similar algorithm was also presented in [168]. A related result was earlier reported in [169] in which with their β_k^{DY} choice, only a standard Wolfe line-search is required to generate the descent direction d_{k+1} . It can be shown that the scaled BB step-size proposed in this thesis satisfies some Wolfe-type line-search condition under certain restrictions. As noted in [124], it is expected that a convergent BB-like method might show improved performance compared to the CGM for this class of functions. It would be of interest to further explore how this scaled BB step-size or related results could fit in conjugate gradient methods. Moreso, the recent surge in developing spectral conjugate gradient methods for unconstrained optimization is evident in [170, 171].

Appendix A

A.1 Robust Circle Criterion for Input-Multiplicative Uncertainty

Set $W(j\omega) = \eta$, substitute $G(j\omega)$ and let $x_\omega = b^2 + \omega^2$, then stability condition (4.17) reduces to

$$2\eta kab + (2\eta - 1)x_\omega - \eta^2 k^2 \bar{w}^2 a^2 > 0 \quad \forall \omega \in \mathbb{R}. \quad (\text{A.1})$$

It is sufficient that (A.1) is satisfied for $n \geq \frac{1}{2}$ and $2\eta kab - \eta^2 k^2 \bar{w}^2 a^2 > 0$. Moreover, larger values of k are obtained for smaller values of n . Choose $\eta = \frac{1}{2}$, then the stability condition (A.1) reduces to

$$4kab - \frac{k^2 \bar{w}^2 a^2}{4} > 0, \quad (\text{A.2})$$

$$k < \frac{4b}{a\bar{w}^2}. \quad (\text{A.3})$$

This equality (A.3) is subsequently shown to be indeed tight for the case of $\bar{w} < 2$. Rewriting the stability condition (A.1) as

$$-n^2 k^2 \bar{w}^2 a^2 + 2\eta(x_\omega + kab) - x_\omega. \quad (\text{A.4})$$

There $\exists \eta$ s.t (A.4) is satisfied provided (A.4) has real roots. We therefore require for real roots of (A.4) that we have

$$(2kab + 2x_\omega)^2 - 4(k^2a^2\bar{w}^2x_\omega) > 0, \quad (\text{A.5})$$

$$x_\omega^2 + (2kab - k^2a^2\bar{w})x_\omega + k^2a^2b^2 > 0. \quad (\text{A.6})$$

(A.6) is satisfied $\forall \omega$ provided (A.6) has no real roots. We therefore require for no real roots of (A.6) that we have

$$(2kab - k^2a^2\bar{w})^2 - 4k^2a^2b^2 \leq 0, \quad (\text{A.7})$$

$$k^4a^4\bar{w}^4 - 4k^3a^3b\bar{w}^2 \leq 0, \quad (\text{A.8})$$

$$k \leq \frac{4b}{a\bar{w}^2}. \quad (\text{A.9})$$

The minimum of the function in (A.6) occurs at

$$\min x_\omega = -\frac{1}{2}(2kab - k^2a^2\bar{w}) \Big|_{k=\frac{4b}{a\bar{w}^2}}, \quad (\text{A.10})$$

$$\min x_\omega = \frac{4b^2}{\bar{w}^2}, \quad (\text{A.11})$$

$$\omega_{min}^2 = \frac{4b^2}{\bar{w}^2} - b^2. \quad (\text{A.12})$$

Thus with $\bar{w} < 2$, we have $k \leq \frac{4b}{a\bar{w}^2}$ for robust absolute stability. With $\bar{w} \geq 2$, the minimum of the function in (A.6) occurs at $\omega_{min} = 0$. Hence, for $\bar{w} \geq 2$, choose $\eta = \frac{kab+b^2}{k^2\bar{w}^2a^2}$, then the stability condition (A.1) reduces to

$$b^2 + 2kab + k^2a^2(1 - \bar{w}^2) > 0. \quad (\text{A.13})$$

Thus with $\bar{w} \geq 2$, we therefore have $k < \frac{b}{a(\bar{w}-1)}$ for robust absolute stability.

Bibliography

- [1] M. A. Dornheim, “Report pinpoint factors leading to YF-22 crash,” Aviation Week Space Technology., Technical Report, pp. 53–54, 1992.
- [2] J. C. Doyle, R. S. Smith, and D. F. Enns, “Control of plants with input saturation nonlinearities,” in *Proceedings of the American Control Conference.*, Minneapolis, US, June 1987.
- [3] P. A. Perez, R. Strietzel, and N. Mort, *Control Engineering Solutions: A Practical Approach.* The Institution of Engineering and Technology, 1997.
- [4] A. Visioli, *Advances in Industrial Control: Practical PID Control.* Springer-Verlag London Limited, 2006.
- [5] M. Kothare, P. Campo, M. Morari, and C. Nett, “A unified framework for the study of anti-windup designs,” *Automatica*, vol. 30, pp. 1869–1883, 1994.
- [6] M. V. Kothare and M. Morari, “Multiplier theory for stability analysis of anti-windup control systems,” in *Proceedings of the 34th IEEE Conference on Decision and Control*, Los Angeles, US, December 1995.
- [7] F. D’Amato, M. Rotea, A. Megretski, and U. Jönsson, “New results for analysis of systems with repeated nonlinearities,” *Automatica*, vol. 37, pp. 739–747, 2001.
- [8] K. S. Walgama and J. Sternby, “Conditioning technique for multiinput-multioutput processes with input saturation,” *IET Journal on Control Theory and Applications*, vol. 140(4), pp. 231–242, 1993.

- [9] Y. Peng, D. Vrancic, R. Hanus, and S. S. R. Weller, “Anti-windup designs for multivariable controllers,” *Automatica*, vol. 12, pp. 1559–1565, 1998.
- [10] M. Soroush and S. Valluri, “Optimal directionality compensation in processes with input saturation nonlinearities.” *International Journal of Control*, vol. 72(17), pp. 1555–1564, 1999.
- [11] A. Adegbege and W. Heath, “Two-stage multivariable antiwindup design for internal model control,” in *Proceedings of the 9th International Symposium on Dynamics and Control of Process Systems*, Leuven, Belgium, July 2010.
- [12] A. A. Adegbege and W. P. Heath, “Internal model control design for input-constrained multivariable process,” *American Institute of Chemical Engineers (AIChE) Journal.*, vol. 57(12), pp. 3459–3472, 2011.
- [13] W. P. Heath, A. G. Wills, and J. A. G. Akkermans, “A sufficient condition for the stability of optimizing controllers with saturating actuators.” *International Journal of Robust and Nonlinear Control.*, vol. 15, pp. 515–529, 2005.
- [14] M. C. Turner, G. Herrmann, and I. Postlethwaite, “Incorporating robustness requirements into anti-windup design,” *IEEE Journal of Automatic Control*, vol. 52(10), pp. 1842–1855, 2007.
- [15] R. M. Morales, G. Li, and W. P. Heath, “Anti-windup and the preservation of robustness against structured norm-bounded uncertainty,” in *Proceedings of the 17th World Congress, The International Federation of Automatic Control*, Seoul, Korea, July 2008.
- [16] A. A. Adegbege and W. P. Heath, “Robust anti-windup synthesis for directionality compensation,” in *7th IFAC Symposium on Robust Control Design*, Aalborg, Denmark, July 2012.

- [17] P. F. Weston and Postlethwaite, “Analysis and design of linear conditioning schemes for systems containing saturating actuators.” in *Proceedings of the IFAC Nonlinear Control System Design Symposium*, Enschede, 1998.
- [18] G. Grimm, J. Hatfield, I. Postlethwaite, A. R. Teel, M. C. Turner, and L. Zaccarian, “Antiwindup for stable linear systems with input saturation: an LMI-based synthesis,” *IEEE Journal of Automatic Control*, vol. 48(9), pp. 1509–1525, 2003.
- [19] R. Alli-Oke, J. Carrasco, W. P. Heath, and A. Lanzon, “A robust Kalman conjecture for first-order plants,” in *Proceedings of 7th IFAC Symposium on Robust Control Design*, Aalborg, Denmark, June 2012.
- [20] S. Richter, C. N. Jones, and M. Morari, “Real-time input-constrained MPC using fast gradient methods,” in *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, China, December 2009.
- [21] —, “Computational complexity certification for real-time MPC with input constraints based on the fast gradient method,” *IEEE Transactions on Automatic Control*, vol. 57(6), pp. 1391–1403, 2012.
- [22] Y. Nesterov, “A method of solving a convex programming problem with convergence rate $o(1/k^2)$,” *Soviet Mathematics Doklady*, vol. 27(2), pp. 372–376, 1983.
- [23] —, *Introductory Lectures on Convex Programming, Volume I: Basic course*. Kluwer Academic Publishers, 2004.
- [24] R. M. Morales, G. Li, and W. P. Heath, “Anti-windup and the preservation of robustness against structured norm-bounded uncertainty,” *International Journal of Robust and Nonlinear Control*, 2013.
- [25] B. O’Donoghue and E. Candès, “Adaptive restart for accelerated gradient schemes,” *The Journal of the Society for Foundations of Computational Mathematics*, 2013.

- [26] M. Raydan, “The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem,” *SIAM Journal on Optimization*, vol. 7(1), pp. 26–33, 1997.
- [27] E. G. Birgin, J. M. Martínez, and M. Raydan, “Nonmonotone spectral projected gradient methods on convex sets,” *SIAM Journal on Optimization*, vol. 10(4), pp. 1196–1211, 2000.
- [28] L. Francisco and M. Raydan, “Gradient method with dynamical retards for large-scale optimization problems.” *Electronic Transactions on Numerical Analysis*, vol. 16, pp. 186–193, 2003.
- [29] H. Zhang, William, and W. Hager, “A nonmonotone line search technique and its application to unconstrained optimization,” *SIAM Journal of Optimization*, vol. 14, pp. 1043–1056, 2004.
- [30] Y.-H. Dai and R. Fletcher, “Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming,” *Numerische Mathematik*, vol. 100(1), pp. 21–47, 2005.
- [31] Z.-J. Shi and J. Shen, “Convergence of nonmonotone line search method,” *Journal of Computational and Applied Mathematics*, vol. 193(2), pp. 397–412, 2006.
- [32] Y.-H. Dai, W. W. Hager, K. Schittkowski, and H. Zhang, “The cyclic Barzilai-Borwein method for unconstrained optimization,” *IMA Journal of Numerical Analysis*, vol. 26(3), pp. 604–627, 2006.
- [33] B. S. Goh, F. Ye, and S. S. Zhou, “Steepest descent algorithms in optimization with good convergence properties,” in *Proceedings of the Chinese Control and Decision Conference, CCDC, Yantai, China, July 2008*.
- [34] W. J. Leong, M. A. Hassan, and M. Farid, “A monotone gradient method via weak secant equation for unconstrained optimization,” *Taiwanese Journal of Mathematics*, vol. 14(2), pp. 413–423, 2010.

- [35] K. Amini, M. Ahookhosh, and H. Nosratipour, “An inexact line search approach using modified nonmonotone strategy for unconstrained optimization,” *Numerical Algorithms*, pp. 1–30, 2013.
- [36] S. E. Lyshevski, *Control Systems Theory with Engineering Applications*. Birkhauser Boston, 2001.
- [37] L. Debnath and P. Mikusinski, *Introduction to Hilbert Spaces with Applications*. Academic Press, 1990.
- [38] W. Heath, “Robust control notes,” University of Manchester, 2008.
- [39] J. Willems, *The Analysis of Feedback Systems*. M. I. T. Press, 1971.
- [40] J. Doyle, B. Francis, and A. Tannenbaum, *Feedback Control Theory*. Macmillan Publishing Corporation, 1990.
- [41] U. Jönsson, “Lecture notes on integral quadratic constraints,” 2001.
- [42] A. Lyapunov, “The general problem of motion stability(in Russian). Translated to English, Ann. Math. Study no. 17, 1949,” Ph.D. dissertation, Princeton University Press, 1892.
- [43] M. Gopal, *Modern Control Systems*. New Age International Publishers, 1993.
- [44] G. Chen, “Stability of nonlinear systems,” in *Encyclopedia of RF and Microwave Engineering*. Wiley, New York, 2004, pp. 4881–4896.
- [45] Z. Zhijun, “Converse Lyapunov theorems for nonautonomous discrete-time systems,” *Journal of Mathematical Sciences*, vol. 161(2), pp. 337–343, 2009.
- [46] A. Megretski and A. Rantzer, “System analysis via integral quadratic constraints,” *IEEE Journal of Automatic Control*, vol. 42(6), pp. 819–830, 1997.
- [47] J. C. Willems and K. Takaba, “Dissipativity and stability of interconnections,” *International Journal of Robust And Nonlinear Control*, vol. 17, pp. 563–586, 2007.

- [48] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River : Pearson Prentice-Hall, 2000.
- [49] J. Bao and P. L. Lee, *Process Control: The Passive Systems Approach*. Springer London, pp 15, 2007.
- [50] M. Vidyasagar, *Nonlinear Systems Analysis*. Prentice-Hall International, 1978.
- [51] M. G. Safonov, *Stability and Robustness of Multivariable Feedback Systems*, A. S. Willsky., Ed. The MIT Press Classics Series, 1980.
- [52] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics (SIAM), 1994.
- [53] A. Zheng, M. V. Kothare, and M. Morari, “Anti-windup design for internal model control,” *International Journal of Control*, vol. 60, no. 5, pp. 1015–1024, 1994.
- [54] G. Stein, “Respect the unstable,” in *Proceedings of the IEEE Conference of Decision and Control, Tampa, Florida, Bode Prize Lecture*, 1989.
- [55] J. M. Maciejowski, *Predictive control: with constraints*. Pearson Education Limited, 2002.
- [56] F. Tyan and D. S. Bernstein, “Anti-windup compensator synthesis for systems with saturation actuators,” *International Journal of Robust And Nonlinear Control*, vol. 5, pp. 521–537, 1995.
- [57] E. D. Sontag and H. J. Sussmann, “Nonlinear output feedback design for linear systems with saturating controls,” in *Proceedings of 29th IEEE Conference on Decision and Control*, Honolulu, US, December 1990.
- [58] S. Tarbouriech and M. Turner, “Anti-windup design: an overview of some recent advances and open problems,” *IET Control Theory & Applications*, vol. 3(1), pp. 1–19, 2009.

- [59] S. Miyamoto and G. Vinnicombe, “Robust control of plants with saturation nonlinearity based on coprime factor representations,” in *Proceedings of the IEEE Conference on Decision and Control*, Kobe, Japan, December 1996.
- [60] P. F. Weston and I. Postlethwaite, “Analysis and design of linear conditioning schemes for systems containing saturating actuators,” in *Proceedings of the IFAC Nonlinear Control System Design Symposium*, Enschede, 1998.
- [61] A. R. Teel and N. Kapoor, “The \mathcal{L}_2 anti-windup problem: Its definition and solution,” in *Proceedings of the 4th European Control Conference*, Brussels, Belgium, July 1997.
- [62] R. Hanus, M. Kinnaert, and J. L. Henrotte, “Conditioning technique, a general antiwindup and bumpless transfer method,” *Automatica*, vol. 23, pp. 729–739, 1987.
- [63] K. J. Åström and B. Wittenmark, *Computer Controlled Systems: Theory and Design*. Prentice-Hall, 1984.
- [64] H. Fertik and C. Ross, “Direct digital control algorithm with anti-windup feature,” *ISA Transactions*, vol. 6, pp. 317–328, 1967.
- [65] P. F. Weston and I. Postlethwaite, “Linear conditioning for systems containing saturating actuators,” *Automatica*, vol. 36(9), pp. 1347–1354, 2000.
- [66] M. C. Turner and I. Postlethwaite, “A new perspective on static and low order anti-windup synthesis,” *International Journal of Control*, vol. 77, pp. 27–44, 2004.
- [67] M. Bruckner and E. Grunbacher, “Optimal model based anti-windup design,” in *Proceedings of the IEEE International Conference on Control Applications (CCA)*, San Antonio, US, September 2008.
- [68] G. Li, G. Herrmann, D. P. Stoten, J. Tu, and M. C. Turner, “A disturbance rejection anti-windup framework and its application to a substructured system,”

in *47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008.

- [69] M. Rehan, A. Ahmed, and N. Iqbal, "Design & implementation of full order anti-windup with actuator amplitude-rate limiter for an AC motor speed control system," *Journal of the Chinese Institute of Engineers*, vol. 33(3), pp. 397–404, 2010.
- [70] M. Kerr, M. C. Turner, E. Villota, S. Jayasuriya, and I. Postlethwaite, "A robust anti-windup design procedure for SISO systems," *International Journal of Control*, vol. 84(2), pp. 351–369, 2011.
- [71] J.-S. Young, "Robust compensator synthesis for antiwindup design with coprime factor uncertainties," *International Journal of Innovative Computing, Information and Control*, 2012.
- [72] U. Jönsson and A. Rantzer, "Optimization of integral quadratic constraints," in *Advances in linear matrix inequality methods in control*. Society for Industrial and Applied Mathematics, 1999, pp. 109–127.
- [73] V. Bragin, V. Vagaitsev, N. Kuznetsov, and G. Leonov, "Algorithms for finding hidden oscillations in nonlinear systems: The Aizerman and Kalman conjectures and Chua's circuits," *International Journal of Computer and Systems Sciences*, vol. 50(4), pp. 511–543, 2011.
- [74] N. E. Barabanov, "On the Kalman problem," *Siberian Mathematical Journal*, vol. 29(3), pp. 3–11, 1988.
- [75] G. Leonov, I. Burkin, and A. Shepeljavyi, *Frequency methods in oscillation theory*. Kluwer Academic Publishers, 1996.
- [76] G. Zames and P. L. Falb, "Stability conditions for systems with monotone and slope-restricted nonlinearities," *SIAM Journal on Control and Optimization*, vol. 6, pp. 89–108, 1968.

- [77] S. Impram and N. Munro, “A note on absolute stability of uncertain systems,” *Automatica*, vol. 37, pp. 605–610, 2001.
- [78] V. L. Kharitnov, “Asymptotic stability of a family of systems of linear differential equations,” *In: Translation in Differential Equations*, vol. 14, pp. 1483–1485, 1979.
- [79] Y. Z. Tsypkin and B. T. Polyak, *Robustness of dynamic systems with parameter uncertainties*. Basel, Switzerland: BirkhaK user Verlag., 1992, pp. 113–121.
- [80] U. Jonsson and A. Megretski, “The Zames–Falb IQC for systems with integrators,” *IEEE Transactions on Automatic Control*, vol. 45(3), pp. 560–565, 2000.
- [81] V. V. Kulkarni and M. G. Safonov, “All multipliers for repeated monotone nonlinearities,” *IEEE Journal of Automatic Control*, vol. 47(7), pp. 1209–1212, 2002.
- [82] B. T. Polyak and Y. Z. Tsypkin, “Robust Nyquist Criterion,” *Automation and Remote Control*, vol. 53, pp. 972–977, 1992.
- [83] W. Heath, R. Morales, and G. Li, “Robustness preserving antiwindup: Examples and counterexamples,” in *Proceedings of the IEEE International Computer-Aided Control System Design Symposium*, Denver, US, September 2011.
- [84] R. Morales, W. Heath, and G. Li, “Robustness preserving anti-windup for SISO systems,” in *Proceedings of the 50th IEEE Conference on Decision and Control*, Orlando, US, December, 2011.
- [85] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.
- [86] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2nd ed., 2006.
- [87] N. Gould and S. Leyffer, “An introduction to algorithms for nonlinear optimization,” in *Frontiers in Numerical Analysis*, J. Blowey, A. Craig, and T. Shardlow, Eds. Springer Berlin Heidelberg, 2003, pp. 109–197.

- [88] G. B. Dantzig, “Reminiscences about the origins of linear programming,” *Operations Research Letters*, vol. 1(2), pp. 43–48, 1982.
- [89] N. Karmarkar, “A new polynomial-time algorithm for linear programming,” *Combinatorica*, vol. 4, pp. 373–395, 1984.
- [90] S. J. Wright, *Primal-Dual Interior-Point Methods*. SIAM, 1997.
- [91] M. Kojima, N. Megiddo, and S. Mizuno, “A primal-dual infeasible-interior-point algorithm for linear programming,” *Mathematical Programming*, vol. 63, pp. 263–280, 1993.
- [92] S. Mehrotra, “On the implementation of a primal-dual interior point method,” *SIAM Journal of Optimization*, vol. 2(4), pp. 575–601, 1992.
- [93] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, Massachusetts, 2nd ed., 1999.
- [94] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38(1), pp. 3–20, 2002.
- [95] P. Tøndel, T. A. Johansen, and A. Bemporad, “An algorithm for multi-parametric quadratic programming and explicit MPC solutions,” *Automatica*, vol. 39(3), pp. 489–497, 2003.
- [96] Y. Wang and S. Boyd, “Fast model predictive control using online optimization,” *IEEE Transactions on Control Systems Technology*, vol. 18(2), pp. 267–278, 2010.
- [97] C. N. Jones and M. Morari, “The double description method for the approximation of explicit MPC control laws,” in *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008, pp. 4724–4730.
- [98] C. Feller and T. Johansen, “Explicit MPC of higher-order linear processes via combinatorial multi-parametric quadratic programming,” in *Proceedings of the European Control Conference (ECC)*, Zürich, Switzerland, July 2013.

- [99] A. Gupta, S. Bhartiya, and P. Nataraj, “A novel approach to multi-parametric quadratic programming,” *Automatica*, vol. 47(9), pp. 2112–2117, 2011.
- [100] C. V. Rao, S. J. Wright, and J. B. Rawlings, “Application of interior-point methods to model predictive control,” *Journal of optimization theory and applications*, vol. 99(3), pp. 723–757, 1998.
- [101] R. Bartlett, A. Wachter, and L. Biegler, “Active set vs. interior point strategies for model predictive control,” in *Proceedings of the American Control Conference*, Chicago, US, June 2000.
- [102] R. Milman and E. Davison, “A fast MPC algorithm using nonfeasible active set methods,” *Journal of Optimization Theory and Applications*, vol. 139(3), pp. 591–616, 2008.
- [103] A. Richards, “Fast model predictive control with soft constraints,” in *Proceedings of the European Control Conference (ECC)*, Zürich, Switzerland, July 2013.
- [104] A. G. Wills, D. Bates, A. J. Fleming, B. Ninness, and S. R. Moheimani, “Model predictive control applied to constraint handling in active noise and vibration control,” *IEEE Transactions on Control Systems Technology*, vol. 16(1), pp. 3–12, 2008.
- [105] L. K. McGovern, “Computational analysis of real-time convex optimization for control systems,” Ph.D. dissertation, Massachusetts Institute of Technology, 2000.
- [106] M. Kogel and R. Findeisen, “Fast predictive control of linear systems combining Nesterov’s gradient method and the method of multipliers,” in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, US, December 2011.
- [107] P. Zometa, M. Kogel, T. Faulwasser, and R. Findeisen, “Implementation aspects of model predictive control for embedded systems,” in *Proceedings of the American Control Conference (ACC)*, Montreal, Canada, June 2012.

- [108] P. Giselsson, “Execution time certification for gradient-based optimization in model predictive control,” in *Proceedings of the IEEE 51st Annual Conference on Decision and Control (CDC)*, Maui, US, December 2012.
- [109] J. Barzilai and J. Borwein, “Two-point step size gradient methods,” *IMA Journal of Numerical Analysis*, vol. 8, pp. 141–148, 1988.
- [110] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*. Chapter 2.2: Analyzing algorithms, pp. 25-27. MIT Press and McGraw-Hill (Second Edition), 2001.
- [111] R. L. Burden and J. D. Faires, *Numerical Analysis*. Brooks/Cole Cengage Learning, 2011.
- [112] W. Sun and Y.-X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*. Springer science+Business Media, LLC, 2006.
- [113] M. Schatzman, *Numerical analysis: a mathematical introduction*. Clarendon Press, Oxford, 2002.
- [114] A. H. V. Tuyl, “Acceleration of convergence of a family of logarithmically convergent sequences,” *Mathematics of Computation*, vol. 62(207), pp. 229–246, 1994.
- [115] D. A. Bader, B. M. E. Moret, and P. Sanders, *Experimental Algorithmics: From Algorithm Design to Robust and Efficient Software*, R. Fleischer, B. Moret, and E. M. Schmidt, Eds. Springer-Verlag Berlin Heidelberg New York, 2002.
- [116] E. D. Dolan and J. J. More, “Benchmarking optimization software with performance profiles,” *Mathematical Programming*, vol. 91(2), pp. 201–213, 2002.
- [117] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems.” *SIAM Journal of Imaging Sciences*, vol. 2(1), pp. 183–202, 2009.

- [118] P. Patrinos and A. Bemporad, “An accelerated dual gradient-projection algorithm for linear model predictive control,” in *Proceedings of the 51st IEEE Conference on Decision and Control*, Maui, US, December 2012.
- [119] C. C. Gonzaga and E. W. Karas, “Fine tuning Nesterov’s steepest descent algorithm for differentiable convex programming,” *Mathematical Programming, Series A*, vol. 138, p. 141–166, 2013.
- [120] B. Shah, R. Buehler, and O. Kempthorne, “Some algorithms for minimizing a function of several variables,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 12(1), pp. 74–92, 1964.
- [121] B. Polyak, “Some methods of speeding up the convergence of iteration methods,” *{USSR} Computational Mathematics and Mathematical Physics*, vol. 4(5), pp. 1–17, 1964.
- [122] A. Bhaya and E. Kaszkurewicz, “Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method,” *Neural Networks*, vol. 17(1), pp. 65–71, 2004.
- [123] M. Torii and M. T. Hagan, “Stability of steepest descent with momentum for quadratic functions,” *IEEE Transactions on Neural Networks*, vol. 13(3), pp. 752–756, 2002.
- [124] R. Fletcher, “On the Barzilai-Borwein method,” in *Optimization and Control with Applications*, Applied Optimization, L. Qi, K. Teo, and X. Yang, Eds. Springer US, 2005, vol. 96, pp. 235–256.
- [125] S.-L. Hu, Z.-H. Huang, and N. Lu, “A nonmonotone line search algorithm for unconstrained optimization,” *Journal of Scientific Computing*, vol. 42, pp. 38–53, 2010.
- [126] A. S. Nemirovski, “Efficient methods in convex programming,” 1995.

- [127] S. R. Becker, E. J. Candes, and M. C. Grant, “Templates for convex cone problems with applications to sparse signal recovery,” *Mathematical Programming Computation*, vol. 3(3), pp. 165–218, 2011.
- [128] Y. Nesterov, “Gradient methods for minimizing composite objective function,” 2007.
- [129] M. Gu, L.-H. Lim, and C. Wu, “Parnes: A rapidly convergent algorithm for accurate recovery of sparse and approximately sparse signals,” *Numerical Algorithms*, vol. 64(2), pp. 321–347, 2013.
- [130] G. Lan and R. D. Monteiro, “Iteration-complexity of first-order penalty methods for convex programming,” *Mathematical Programming*, vol. 138(1-2), pp. 115–139, 2013.
- [131] T. P. Vogl, J. Mangis, A. Rigler, W. Zink, and D. Alkon, “Accelerating the convergence of the back-propagation method,” *Biological cybernetics*, vol. 59(4-5), pp. 257–263, 1988.
- [132] T. Goldstein, B. O’Donoghue, and S. Setzer, “Fast alternating direction optimization methods,” UCLA, Technical Report, May 2012 (Revised January 2014).
- [133] A. Kozma, J. V. Frasch, and M. Diehl, “A distributed method for convex quadratic programming problems arising in optimal control of distributed systems,” in *Proceedings of the 52nd IEEE Conference on Decision and Control*, Florence, Italy, December 2013.
- [134] Q. Lin and L. Xiao, “An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization,” in *Proceedings of The 31st International Conference on Machine Learning*, 2014.
- [135] X. Meng and H. Chen, “Accelerating Nesterov’s method for strongly convex functions with Lipschitz gradient,” *Mathematics - Optimization and Control*, 90C25. *arXiv:1109.6058v1*, pp. 1–13, 2011.

- [136] L. R. Nicolas, S. Mark, and B. Francis, “A stochastic gradient method with an exponential convergence rate for finite training sets,” *Mathematics - Optimization and Control*, *arXiv:1202.6258v4*, pp. 1–34, 2013.
- [137] M. Collins, R. E. Schapire, and Y. Singer, “Logistic regression, Adaboost and Bregman distances,” *Machine Learning*, vol. 48, pp. 253–285, 2002.
- [138] T. P. Minka, “Numerical optimizers for logistic regression,” (2013). [Online]. Available: <http://fa.bianp.net/blog/2013/numerical-optimizers-for-logistic-regression/#fn:2>
- [139] Y.-H. Dai and L.-Z. Liao, “R-linear convergence of the Barzilai and Borwein gradient method,” *IMA Journal of Numerical Analysis*, vol. 22, pp. 1–10, 2002.
- [140] A. Friedlander, J. M. Martínez, and M. Raydon, “A new method for large-scale box constrained convex quadratic minimization problems,” *Optimization Methods and Software*, vol. 5(1), pp. 57–74, 1995.
- [141] R. De Asmundis, D. di Serafino, F. Riccio, and G. Toraldo, “On spectral properties of steepest descent methods,” *IMA Journal of Numerical Analysis*, 2013.
- [142] L. Grippo, F. Lampariello, and S. Lucidi, “A nonmonotone line search technique for Newton’s method,” *SIAM Journal on Numerical Analysis*, vol. 23, pp. 707–716, 1986.
- [143] P. L. Toint, “Nonmonotone trust-region algorithms for nonlinear optimization subject to convex constraints,” *Mathematical Programming*, vol. 77(3), pp. 69–94, 1997.
- [144] Y. H. Dai, “On the nonmonotone line search,” *Journal of Optimization Theory and Applications*, vol. 112, pp. 315–330, 2002.
- [145] Y.-H. Dai and H. Zhang, “Adaptive two-point stepsize gradient algorithm,” *Numerical Algorithms*, vol. 27(4), pp. 377–385, 2001.

- [146] B. Zhou, L. Gao, and Y.-H. Dai, “Gradient methods with adaptive step-sizes,” *Computational Optimization and Applications*. Kluwer Academic Publishers., vol. 35(1), pp. 69–86, 2006.
- [147] Y. Yuan, “Gradient methods for large scale convex quadratic functions,” in *Optimization and Regularization for Computational Inverse Problems and Applications*. Springer, 2011, pp. 141–155.
- [148] Y. Dai, J. Yuan, and Y.-X. Yuan, “Modified two-point stepsize gradient methods for unconstrained optimization,” *Computational Optimization and Applications*, vol. 22(1), pp. 103–109, 2002.
- [149] R. Tavakoli and H. Zhang, “A nonmonotone spectral projected gradient method for large-scale topology optimization problems,” *Numerical Algebra, Control and Optimization*, vol. 2(2), pp. 395–412, 2012.
- [150] Y.-H. Dai, “Alternate step gradient method,” *A Journal of Mathematical Programming and Operations Research*, vol. 52(4,5), pp. 395–415, 2003.
- [151] A. Friedlander, J. Martinez, B. Molina, and M. Raydan, “Gradient method with retards and generalizations,” *SIAM Journal of Numerical Analysis*, vol. 36, pp. 275–289, 1999.
- [152] R. Varadhan and P. Gilbert, “BB: An R package for solving a large system of nonlinear equations and for optimizing a high-dimensional nonlinear objective function,” *Journal of Statistical Software*, vol. 32(4), pp. 1–26, 2009.
- [153] E. G. Birgin, J. M. Martínez and M. Raydan, “Spectral projected gradient methods: Review and perspectives,” *Journal of Statistical Software*, pp. 1–21, 2012.
- [154] A. Bhaya and E. Kaszkurewicz, “Iterative methods as dynamical systems with feedback control,” in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, US, December 2003.

- [155] J.-P. Chehab and J. Laminie, “Differential equations and solution of linear systems,” *Numerical Algorithms*, vol. 40(2), pp. 103–124, 2005.
- [156] A. Bhaya and E. Kaszkurewicz, “A control-theoretic approach to the design of zero finding numerical methods,” *IEEE Transactions on Automatic Control*, vol. 52(6), pp. 1014–1026, 2007.
- [157] S. Babaie-Kafaki and M. Fatemi, “A modified two-point stepsize gradient algorithm for unconstrained minimization,” *Optimization Methods and Software*, vol. 28(5), pp. 1040–1050, 2013.
- [158] F. Biglari and M. Solimanpur, “Scaling on the spectral gradient method,” *Journal of Optimization Theory and Applications*, vol. 158, pp. 626–635, 2013.
- [159] A. A. Ahmadi and P. A. Parrilo, “Nonmonotonic Lyapunov functions for stability of discrete time nonlinear and switched systems,” in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008.
- [160] S. F. Derakhshan and A. Fatehi, “Nonmonotonic Lyapunov functions for stability analysis and stabilization of discrete time Takagi-Sugeno fuzzy systems,” *International Journal of Innovative Computing, Information and Control*, 2012.
- [161] A. N. Michel and L. Hou, “Stability results for finite-dimensional discrete-time dynamical systems involving nonmonotonic Lyapunov functions,” in *American Control Conference (ACC)*, Baltimore, US, December 2010.
- [162] A. N. Michel and L. Hou, “Stability theory of discrete-time dynamical systems involving nonmonotonic Lyapunov functions,” *Nonlinear Studies*, vol. 21(1), pp. 53–75, 2014.
- [163] M. J. Koegel and R. Findeisen, “A fast gradient method for embedded linear predictive control,” in *18th IFAC World Congress*, Milano, Italy, August 2011.

- [164] M. Rubagotti, “Stabilizing linear model predictive control under inexact numerical optimization,” in *European Control Conference (ECC)*, Zürich, Switzerland, July 2013.
- [165] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari, “Embedded predictive control on an FPGA using the fast gradient method,” in *European Control Conference (ECC)*, Zürich, Switzerland, July 2013.
- [166] W. W. Hager and H. Zhang, “A survey of nonlinear conjugate gradient methods,” *Pacific Journal of Optimization*, vol. 2(1), pp. 35–58, 2006.
- [167] ———, “A new conjugate gradient method with guaranteed descent and an efficient line search,” *SIAM Journal on Optimization*, vol. 16(1), pp. 170–192, 2005.
- [168] J. Liu and S. Wang, “Modified nonlinear conjugate gradient method with sufficient descent condition for unconstrained optimization,” *Journal of Inequalities and Applications*, vol. 20(1), pp. 1–12, 2011.
- [169] Y.-H. Dai and Y. Yuan, “A nonlinear conjugate gradient method with a strong global convergence property,” *SIAM Journal on Optimization*, vol. 10(1), pp. 177–182, 1999.
- [170] E. G. Birgin and J. M. Martinez, “A spectral conjugate gradient method for unconstrained optimization,” *Applied Mathematics and Optimization*, vol. 43(2), pp. 117–128, 2001.
- [171] S. Deng, Z. Wan, and X. Chen, “An improved spectral conjugate gradient algorithm for nonconvex unconstrained optimization problems,” *Journal of Optimization Theory and Applications*, vol. 157(3), pp. 820–842, 2013.