First Order Optimisation Algorithms in Uncertain Environments

Iterate to minimise the uncertain gap between rationality and reality

潘周聘

Pan, ZhouDan

Control Group
Department of Engineering Science
St Peter's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Supervised by Prof. Mark Cannon

Dedicated to my family
for their unwavering support and encouragement.

Dedicated to the vicissitudes of life
for the uncertainties that shape our rationality and reality.

Bon voyage.

Abstract

With advances in onboard computing power and inter-processing-unit communication technology, decision-making systems facilitated with online distributed algorithms are becoming increasingly advantageous. However, in this setting uncertainties are prevalent and non-diminishing due to estimation errors and communication asynchrony.

The first part of this thesis focuses on solving convex distributed optimisation problems with local consensus coupling constraints via the alternating direction method of multipliers (ADMM), using an asynchronous methodology allowing for communication delays. We use a bipartite undirected graph to denote the update structure of the processing agents that cooperatively perform the distributed algorithm without a centralised aggregator. We introduce a data server to exchange the asynchronous consensus data among the processing agents. Under technical assumptions involving bounded delays, bounded step sizes, and strong convexities in parts of the local objectives, the running average of local iterates generated by the proposed asynchronous algorithm converges to an optimal solution.

To solve convex optimisation problems we rely on iterative algorithms, but when the problem contains parameters that need to be estimated from measurements with noise, another iterative process is needed to perform estimation. In the second part of this thesis we devise a modified version of ADMM that performs parameter estimation simultaneously with optimisation. Given convergent parameter estimates, and assuming the objective can be expressed in terms of a multi-parametric quadratic program (mp-QP), we prove convergence of the objective values, dual variables and primal residual. Simulation results show that the rate of convergence tracks that of the estimator up to an upper limit that is characterised by convergence rate of ADMM with no parametric uncertainty.

The third part of this thesis provides a general framework for analysing the convergence of online decision-making algorithms under uncertainty. These algorithms are formulated as recursive fixed-point iterations characterised by stochastic nonexpansive operators that are parametrised by recursive estimates of uncertain parameters in the presence of noise. Since nonexpansiveness is preserved under convex combinations, we propose the concept of the stochastic mean operator as the overall averaged operator. Assuming non-i.i.d. and finite variance convergence of the estimated parameters, we provide first and second moment convergence theorems of the iterates. Numerical results show the robustness of the iterates when almost all the parameters of the optimisation problem are uncertain. In addition to the obvious improvements in computational efficiency that this approach affords, we also observe an "advantage over perfectionism" effect, in which the proposed algorithm outperforms the optimal solutions defined by the most recent estimated parameters.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

Sets

$\mathbb{R}$	The set of real numbers.
$\mathbb{R}^n$	The set of real vectors of dimension n.
$\mathbb{R}^{m\times n}$	The set of real matrices of dimension $m \times n$.
$\mathbb{R}_+$	The set of non-negative real numbers.
$\mathbb{S}^n$	The set of real symmetric matrices of dimension $n \times n$.
$\mathbb{S}^n_+/\mathbb{S}^n_{++}$	The set of symmetric and positive definite/semidefinite real matrices of dimension $n \times n$.
$\mathbf{relint}(\mathcal{C})$	Relative interior of set $\mathcal{C}$.
$\mathbf{int}(\mathcal{C})$	Interior of set $\mathcal{C}$.
$\mathbf{bdry}(\mathcal{C})$	Boundary of set $\mathcal{C}$.

Vectors, matrices and norms

$\mathbf{tr}\,(A)$	The trace of A.
$\mathbf{null}(A)$	The null space of A.
$\mathcal{C}^\perp$	The orthogonal space of $\mathcal{C}$.
$\mathbf{col}(\mathcal{A})$	The column space of A.
$x^\intercal$	The transpose of x.
$\langle x, y\rangle$	The inner product between x and y.
$\|x\|/\|A\|$	The norm of vector x or matrix A,
$\|x\|_Q^2$	The square of norm scaled by Q (i.e. $\|x\|_Q^2 := \langle x, Qx\rangle$ with $Q \succ 0$.)
$A \otimes B$	The Kronecker product between A and B
$\mathbf{1}$	The vector/matrix of ones in appropriate dimension.
$\mathbf{0}$	The vector/matrix of zeros in appropriate dimension.
$\mathrm{diag}(\lambda)$	The diagonal matrix generated by vector λ

Definitions and inequalities

$A := B$	A is defined by B.
$A \leftarrow B$	B is assigned to A.
$A \succ / \succeq 0$	A is positive definite/semidefinite

Convex sets, functions and optimisation

$\mathbf{exp}\,(x)$	The exponential of x.
$\mathcal{I}_{\mathcal{C}}(x)$	The indicator function of set $\mathcal{C}$ (i.e. $\mathcal{I}_{\mathcal{C}}(x) = 0$ and $\mathcal{I}_{\mathcal{C}}(x) = +\infty$ otherwise).
∂f	The subdifferential of function f.
∇f	The gradient of function f.
$N_{\mathcal{C}}(x)$	The normal cone of set $\mathcal{C}$ (i.e. $N_{\mathcal{C}}(x) := \partial\mathcal{I}_{\mathcal{C}}(x)$).
$\mathbf{dist}\,(x, \mathcal{C})$	The distance between x and $\mathcal{C}$.

Operators

$\mathbf{Range}(T)$	The range of the operator T.
$\mathbf{dom}(T)$	The domain of T.
$\mathbf{Fix}(\mathcal{T})$	The fixed-point set of T.
$\mathbf{prox}_{\alpha f}\,(v)$	The proximal operator.
$\mathbf{proj}_{\mathcal{C}}\,(x)$	The projection of x onto the set $\mathcal{C}$.

Probabilities

$\mathbb{P}\,[A]$	The probability of the event A.
$\rho\,[x]$	The probability density function of the random variable x.
$\mathbb{E}\,[x]$	The expected value of x.
$x \sim N(\mu, \sigma^2)$	x follows the normal distribution $N(\mu, \sigma^2)$
$x \sim N_{tr}(\mu, \sigma^2, a, b)$	x follows the normal distribution $N(\mu, \sigma^2)$ truncated in $[a, b]$.
$A \overset{e.w.i.}{\sim} D$	Each element of A independently follows the distribution D.
$x \xrightarrow{\mathbf{P}} y$	x converges to y in probability.

.

Chapter 1

Introduction

Recent enhancements in communication technologies and embedded systems have spurred the progression of algorithms designed to coordinate intelligent agents distributively. Unlike centralized decision-making structures, distributed optimization algorithms [1]–[3] allow participating agents to iteratively solve local optimization problems and exchange information with their neighbours through predefined update and communication protocols. These protocols are integral for addressing large-scale problems where privacy is inherent, eventually leading to the asymptotic convergence to the global problem's optimal solution through local iterations.

Distributed optimisation algorithms generally employ iterative methods, relying on primal or primal-dual iterations, to converge to the optimal solution. Inspired by [4], a subset of distributed subgradient methods collectively approximate the common primal consensus by implementing weighted averaging of local objective subgradient updates across a potentially dynamic network [5]–[7]. For large scale problems, it may be impractical to maintain a local copy of the entire consensus vector for each agent. This motivates the use of dual decomposition algorithms [8] that allow agents to share only local variables, rendering them more advantageous. The Alternating Direction Method of Multipliers (ADMM) [1], [9] enhances dual decomposition by incorporating an augmented Lagrangian, expanding the range of solvable problems.

Many numerical algorithms are constructed with the aim of finding the fixed points of a given operator. In the context of convex optimisation, a very wide class of algorithms can be interpreted in terms of a search for the zeros of a monotone operator

[10], which is often recast in terms of the fixed points of a proximal operator [11], [12]. Such operators are necessarily nonexpansive (which means their repeated application defines a non-divergent iteration), and they are the basis of proximal algorithms such as the Douglas-Rachford splitting (DRS) method [11], which includes ADMM.

Conventional iterative algorithms face difficulties when applied to uncertain real-world environments. One of the primary challenges is the potential for delays within the interconnected communication network between computing nodes. The first part of this thesis focuses on the development of an asynchronous distributed ADMM algorithm that avoids delays across an entire network of agents caused by a variable (possibly stochastic) latency of individual agents. We propose the novel concept of a data exchange server, which solely facilitates communication, contrasting with traditional aggregators that participate in computing.

Secondly, many practical optimisation problems are not deterministically defined due to uncertainty in problem parameters that need to be estimated from data. This consideration is central to the second part of the thesis. We consider merging iterative parameter estimation algorithms with optimisation iterates to guarantee convergence of the combined iterative scheme. This is notably effective for l_1-norm stable estimators, e.g., the Luenberger observer, and linear or sub-linear variance converging estimators like the Kalman filter.

Lastly, a broad class of estimators employed in practice estimate the mean value of uncertain parameters by recursively minimising mean square estimation errors using computationally efficient online iterations. This idea can be applied in the context of nonexpansive operator theory by considering operators parametrised by random variables, and studying the stability of the associated mean nonexpansive operators. We thus derive a framework for analysing convergence that is applicable to computational optimal control laws (such as receding horizon control) with simultaneous parameter estimation, control and optimisation iterations.

Chapter 2: Preliminaries

The development of this thesis depends on the preliminary knowledge of convex optimisation, operator theory, distributed optimisation concepts, as well as the estimation and dynamic optimal control methods that constitute an online decision making system (e.g. a feedback controller). This chapter starts with the introduction of convex sets, functions and optimisation problems with Lagrangian duality. We then discuss operator theory, focusing on fixed-point iterations of nonexpansive operators that enable the solution of convex optimisation problems. This is followed by a brief elaboration of distributed optimisation algorithms that combine the previous two theories and their application to modern problems. Lastly, an overview of estimation and optimal control theory is presented to address the necessary elements for an online decision-making system, focusing on dealing with uncertainty while maintaining computational efficiency.

Chapter 3: Asynchronous ADMM via a Data Exchange Server

The increasing number of agents and inevitable delays—attributed to greater distances, packet congestion, and limitations in processing—pose challenges to synchronous algorithms, rendering them susceptible to delays by individual agents. The study in [13] assessed the efficacy of distributed machine learning over a "stale" synchronous server, spurring further investigations into distributed optimisation algorithms with inherent delays [14], [15]. These algorithms, instead of waiting for all agents to synchronise at each iterative step, utilise the most recent information available to compute subsequent updates, thus enhancing efficiency by minimising overall waiting time. However, using outdated data at each step can lead to an accumulation of errors in solution estimates. The research in [14] indicates that, for fixed-point algorithms, such a trade-off can be favourable under certain conditions. In this chapter we focus on asynchronous distributed optimisation via ADMM [15]–[20]. Recent studies [21]–[25] investigate the application of distributed optimisation algorithms through ADMM, thus eliminating the need for a centralised aggregator. On the other hand,

[15], [20] explore asynchronous ADMM with a centralised aggregator, and propose three algorithms whose convergence analyses are based on worst case bounded delay scenarios, to which our proposed algorithm is closely related.

This chapter begins with an introduction to convex optimisation, distributed optimisation, ADMM, and asynchronous algorithms, subsequently presenting the proposed asynchronous ADMM complemented by a data exchange server. Further, it incorporates numerical examples to elucidate the practical convergence conditions, offering insights into the algorithm's applicability and reliability. Following this, it conducts a comprehensive numerical comparison with pertinent algorithms, notably focusing on the one described in [15], to delineate the relative advantages of the introduced asynchronous ADMM within the scope of distributed optimisation. The scaling of communication cost is also discussed in this part. The chapter concludes by a brief outlook of potential future work.

The following paper is based on this chapter:

- Z. Pan and M. Cannon, "Asynchronous ADMM via a Data Exchange Server," *IEEE Transactions on Control of Network Systems*, pp. 1–12, 2024, ISSN: 2325-5870, 2372-2533. DOI: 10.1109/TCNS.2024.3354840. [Online]. Available: https://ieeexplore.ieee.org/document/10400939/

Chapter 4: Optimisation with Parametric Uncertainty

Numerous problems of interest encompass parameters that are uncertain and require estimation. For instance, in an optimal control problem, certain parameters of the controlled system may be imprecise but can be estimated using noisy measurements. Under such circumstances, system identification is often employed to estimate the unknown parameters of the plant [27]. Typically, the decision-making process, responsible for computing an optimal control law, is initiated once the estimated parameters have converged to an adequate level of accuracy. The study conducted by [28] considers solving semidefinite programming problems using an approximate ADMM solver. Our work differs in that [28] does not deal with problems involving uncertain parameters; instead, it utilizes approximations within the exact ADMM iterations to

lessen the computational demands of each iteration, thereby inducing errors in the estimated solutions. In [29] an investigation is described into the standard form of a multi-parametric quadratic program (mp-QP) in which the parameter appears on the right-hand side of the inequality constraints, elucidating that the optimiser is continuous and piecewise affine with respect to the perturbed parameter. This is important in the context of Model Predictive Control (MPC) where the estimation of initial state and polytopic constraints serves as the parameter.

This chapter begins with an introduction to firmly non-expansive operators and the relationship with common iterative algorithms. Then we address optimisation problems with uncertainty, considering in particular problems with mp-QP formulations solved by ADMM. The main results of the chapter are presented in two theorems, and relevant lemmas and the proofs are provided to analyse the deterministic and probabilistic convergence of the optimisation under uncertainty. A numerical study is provided to test the convergence of the theorems. The chapter concludes by a brief outlook of potential future works that links to Chapter 5.

The following paper is based on part of this work:

- Z. Pan and M. Cannon, "Optimisation with Parametric Uncertainty: An ADMM Approach," *IFAC World Congress 2023*, 2023

Chapter 5: Online optimisation with the Recursive Fixed-point method: A framework of stochastic nonexpansive operators

In the previous chapter, the core property that we exploit is firm non-expansiveness operators, namely $\forall v, w \in \mathbb{R}^n$, the operator T satisfies $\|Tv-Tw\|_2^2+\|(I-T)v-(I-T)w\|_2^2 \leq \|v-w\|_2^2$. More generally, a fixed-point iteration converges [10], [31] if the iteration can be represented by a non-expansive operator with the α-averaged property, i.e., the operator can be expressed as $T = \alpha R+(1-\alpha)I$, where R is non-expansive and $\alpha \in (0, 1)$ is a scalar constant[1]. In practical applications of optimisation-based control and decision making (e.g. estimation and robust control from electricity trading [32],

[1]The operator T is firmly non-expansive in the special case of $\alpha = 1/2$, in which case the convergence rate is optimal [10], [31].

[33], building control [34]–[36], optimal power flow [37]–[41], battery SOC estimation [42]–[44]), optimisation problems incorporate parameters that are often unknown or affected by measurement uncertainty and thus need estimation. Proximal operators that are frequently employed in optimisation algorithms have the form of α-averaged operators ([12, Sec. 6]). Moreover, stable estimators often provide estimations of the expected values of the uncertain parameters by recursively minimising the mean square error of the *a posteriori* estimate (e.g. Kalman filter [45]). On the other hand, averaged nonexpansive operators preserve their averagedness under convex combinations, and the expectation operator $\mathbb{E}[\cdot]$ is itself a convex operator. Therefore, the motivation of the work in this chapter is to synthesise the analysis of coupled systems comprising estimator, control and optimiser subsystems by representing the overall dynamic system as an iteration of a stochastic operator characterised by the overall mean nonexpansive operator and conditional i.i.d. disturbances. This allows the assumptions of the previous chapter to be relaxed and provides a very general framework for design and analysis of robust optimisation and optimal control systems.

The chapter begins with a deep review of nonexpansive operators, providing lemmas and different perspectives to view nonexpansive operators. Then we introduce stochastic nonexpansive operators with uncertain parameters, define the mean operator and the convex combination-invariance concept, followed by three main theorems that analyse the first and second moment convergence of the stochastic operators. The recursive fixed-point method is proposed and followed by a numerical study, highlighting the algorithm's convergence and robustness, even in challenging scenarios such as indefinite matrices and constant noise. Notably, an "advantage over perfectionism" effect is observed, where the proposed method achieves superior convergence compared to time-varying optimal solutions, particularly under limited computational resources.

Chapter 6: Conclusion and Outlook

This chapter summarises the contributions made by this thesis and outlines future research directions.

Chapter 2

Preliminaries

In this chapter, we provide an outline for the preliminaries to build up a feedback controller that is able to iteratively accomplish an objective amidst the real-world uncertain environments. This comprises of: (Sec. 2.1) *convex optimisation* that defines the problems to solve, (Sec. 2.2) *operator theory* that defines the iterative solvers, (Sec. 2.3) *distributed optimisation* that defines how the sublevel agents act

collectively, as well as (Sec. 2.4) *recursive estimation and optimal control* that takes the respective inductive or deductive role of a feedback controller. We refer the reader to the works of (Sec. 2.1) for a more comprehensive elaboration.

Notation: The n-dimensional real space is denoted $\mathbb{R}^n$. $\mathbf{tr}(A)$ denotes the trace of A. $\mathbf{int}(\mathcal{C})$ represents the interior of set $\mathcal{C}$ and $\mathbf{relint}(\mathcal{C})$ the relative interior of $\mathcal{C}$. $Q \succ 0$ and $R \succeq 0$ represent positive definite and positive semidefinite matrices. We define $\|x\|_Q^2 := x^\intercal Q x$ for $Q \succ 0$. $\mathcal{I}_\mathcal{C}(x)$ denotes the indicator function (i.e. $\mathcal{I}_\mathcal{C}(x) = 0$ for $x \in \mathcal{C}$ and $\mathcal{I}_\mathcal{C}(x) = +\infty$ otherwise) of the constraint set $x \in \mathcal{C}$. $\partial F(x)$ indicates the subdifferential of function F evaluated at x.

2.1 Convex Optimisation

To solve a problem subject to some constraints for optimal results that minimise an objective of cost, we mathematically formulate the problem $\mathcal{P}$ as:

$$\begin{aligned} \underset{x}{\text{minimise}} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0,\ i = 1, 2, \ldots, m, \\ & l_j(x) = 0,\ j = 1, 2, \ldots, q, \end{aligned} \tag{2.1}$$

in which $x \in \mathbb{R}^n$, and $f : \mathbb{R}^n \to \mathbb{R}^n$, $h_i(x) : \mathbb{R}^n \to \mathbb{R}$, $i = 1, 2, 3, \ldots m$, $l_j(x) : \mathbb{R}^n \to \mathbb{R}$, $j = 1, 2, 3, \ldots m$ are functions. The *feasible/constraint set* $C := \{x \in \mathbf{Dom}(f) | h_i(x) \leq 0, \forall i; l_j(x) = 0, \forall j\}$, and the problem $\mathcal{P}$ is *feasible* if $\mathcal{C} \neq \varnothing$. $\mathcal{P}$ can also be reformulated as the unconstrained problem:

$$\underset{x}{\text{minimise}} \quad \tilde{f}(x), \tag{2.2}$$

where $\tilde{f}(x) = f(x) + \mathcal{I}_\mathcal{C}(x)$ and $\mathcal{I}_\mathcal{C}(x)$ is the indicator function (i.e. $\mathcal{I}_\mathcal{C}(x) = 0$ for $x \in \mathcal{C}$ and $\mathcal{I}_\mathcal{C}(x) = +\infty$ otherwise) of the constraint set $x \in \mathcal{C}$. We note that the objective function $\tilde{f}$ is valued over the extended codomain $\mathbb{R} \cup \{+\infty\}$. We denote the *optimal solution* (if there exists) $x^* \in \arg\min_x \tilde{f}(x)$ of $\mathcal{P}$ as: $\forall x \in \mathbb{R}^n, \tilde{f}(x^*) \leq \tilde{f}(x)$, and the *optimal objective value* $p^* := \tilde{f}(x^*)$.

Remark 2.1. Naturally we ask the question: If $\tilde{f}(x)$ has a local extremum characterised by the first order condition $\partial\tilde{f} = 0$, will this extremum be the global one? Usually this is not true, and to solve a general optimisation problem is usually NP-hard [46], [47]. Therefore the framework of convex optimisation problems is preferred in order to have polynomial-time algorithms [48].

2.1.1 Convex sets, functions and optimisation problems

Definition 2.1 (Convex set)**.** A set $\mathcal{C} \subseteq \mathbb{R}^n$ is convex if $\forall x, y \in \mathcal{C}$ and $\alpha \in [0, 1]$:

$$\alpha x + (1 - \alpha)y \in \mathcal{C}. \tag{2.3}$$

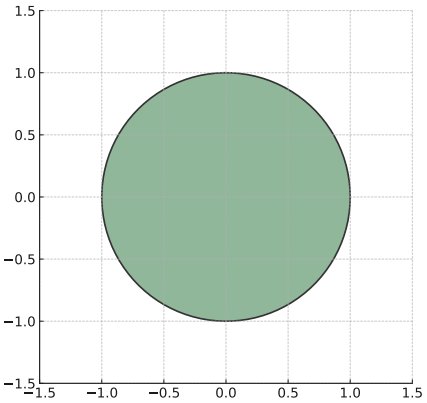
Remark 2.2. We denote a *convex combination* of a set of vectors $\{x_i\}, i = 1, 2, \ldots m$ as the linear combination $\sum_i \alpha_i x_i$ where $\alpha_i > 0, \forall i = 1, 2, \ldots, m$ and $\sum_i \alpha_i = 1$. Definition 2.1 shows a special case of a convex combination when $m = 2$, and in the context of convex analysis the definitions and properties can be extended to the case when $m > 2$. It is worth mentioning that the expectation operator $\mathbb{E}[x]$ (of a random variable x) is a convex (probabilistic) combination[1], and this property motivates Chapter 5.

Example 2.1. The following sets are convex:

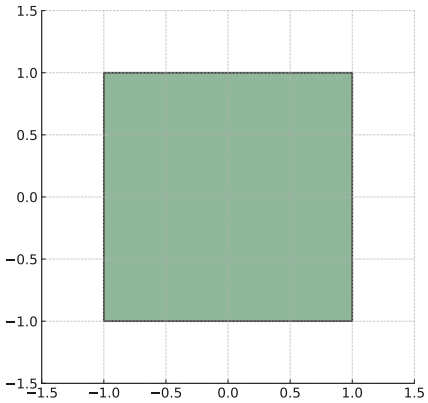
- *Convex cone*: $\mathcal{C} \subseteq \mathbb{R}^n$ is a convex cone if $\alpha x \in \mathcal{C}$ for all $x \in \mathcal{C}$ and $\alpha \geq 0$.
- *Half-plane*: $\mathcal{C} = \{x \in \mathbb{R}^n : \langle a, x \rangle \leq b\}$, where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$.
- *Polyhedron*: $\mathcal{C} = \{x \in \mathbb{R}^n : Ax \leq b\}$, where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.
- *Intersection of convex sets*: $\mathcal{C} = \bigcap_{i=1}^{k} \mathcal{C}_i$ is convex if each $\mathcal{C}_i$ is convex for $i = 1, \ldots, k$.
- *Cartesian product*: $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2 = \{(x, y) \in \mathbb{R}^{n_1+n_2} : x \in \mathcal{C}_1, y \in \mathcal{C}_2\}$ is convex if $\mathcal{C}_1$ and $\mathcal{C}_2$ are convex.

Figure 2.1 provides examples of convex and nonconvex sets.

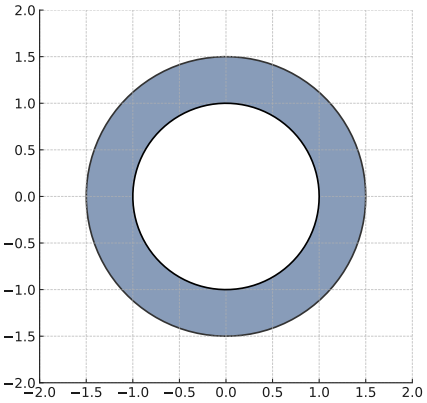
[1] For $\mathbb{E}[x] = \sum_i \alpha_i x_i, \sum_i \alpha_i = 1$, where x is a (discrete) random variable, the convex coefficient $\alpha_i = \mathbb{P}[x = x_i], \forall x_i \in \Omega$ is the probability assigned to each possible outcome in its sample space.



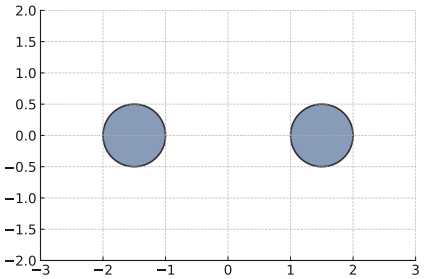
(a) Convex: solid disk.



(b) Convex: rectangle.



(c) Nonconvex: annulus.



(d) Nonconvex: union of two disjoint circles.

Figure 2.1: Examples of convex and nonconvex sets.

Definition 2.2 (Convex function)**.** A function $f : \mathbb{R}^n \to \mathbb{R}$ is called *convex* if $\forall x, y \in \mathbb{R}^n$ and any $\alpha \in [0, 1]$, the following inequality holds:

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y). \tag{2.4}$$

f is *strictly convex* if the inequality in strictly holds. f is *strongly convex with parameter* m if $\exists \beta > 0$ such that

$$g(x) := f(x) - \frac{\beta}{2} \|x\|^2 \tag{2.5}$$

is convex. We have *strong convexity* $\Rightarrow$ *strict convexity* $\Rightarrow$ *convexity.* f is *L-smooth* if f is differentiable and ∇f is Lipschitz with parameter L, which is $\forall x, y \in \mathbb{R}^n$:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|. \tag{2.6}$$

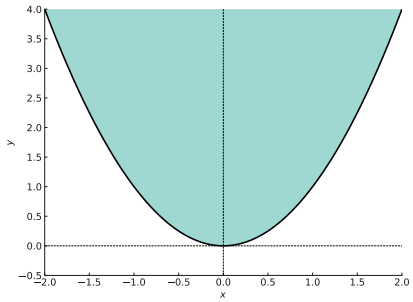
The *epigraph* of a function f is defined as:

$$\mathbf{Epi}(f) := \{(x, t) | x \in \mathbf{Dom}(f), f(x) \leq t\}, \tag{2.7}$$

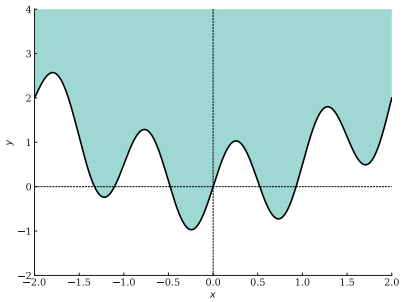
which can be perceived as the region above $f(x)$ defined in $\mathbb{R}^n \times \mathbb{R}$. A function f is convex if and only if $\mathbf{Epi}(f)$ is a convex set.

Example 2.2. The following functions are convex:

- *Affine*: $f(x) = \langle a, x \rangle + b$.
- *Quadratic*: $f(x) = \langle x, Qx \rangle + \langle c, x \rangle + d$, $Q \succeq 0$.
- *Support of a convex set*: $f(x) = \sup_{y \in \mathcal{C}} \langle x, y \rangle$, where $\mathcal{C}$ is convex.
- *Indicator function of a convex set*: $\mathcal{I}_{\mathcal{C}}(x) = 0$ for $x \in \mathcal{C}$ and $\mathcal{I}_{\mathcal{C}}(x) = +\infty$ otherwise, where $\mathcal{C}$ is convex.
- *Point-wise* max *of convex functions*: $f(x) = \max\{f_1(x), \ldots, f_n(x)\}$, f_i is convex.
- *Convex conjugate*: $f^*(y) = \sup_x \{\langle y, x \rangle - f(x)\}$ is convex with any f.



(a) Convex: epigraph of $f(x) = x^2$.



(b) Nonconvex: epigraph of $f(x) = \sin(2\pi x) + 0.5x^2$

Figure 2.2: Example of the epigraphs of a convex and a nonconvex function.

Figure 2.2 provides examples of a convex and a nonconvex function with their epigraphs.

Lemma 2.1. *Convex functions have several desirable properties:*

(1) Global minimum equals local minimum: *For a convex function, every local minimum is also a global minimum.*

(2) Monotonicity of the subdifferential: *The subdifferential $\partial f(x)$ is a monotone set-valued mapping, meaning if $g_1 \in \partial f(x_1)$ and $g_2 \in \partial f(x_2)$, then $\langle (g_1 - g_2), (x_1 - x_2) \rangle \geq 0$.*

(3) Positive semidefinite Hessian: *If f is twice differentiable, the Hessian matrix $\nabla^2 f(x)$ is positive semidefinite at all points in the domain, i.e., $z^T \nabla^2 f(x) z \geq 0$ for all $z \in \mathbb{R}^n$.*

(4) Jensen's inequality: *For a convex function f and any $x, y \in \mathbb{R}^n$ with $\alpha \in [0, 1]$, $f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha) f(y)$.*

(5) Sum of convex functions: *If f_1 and f_2 are convex functions, then $f(x) = f_1(x) + f_2(x)$ is also convex.*

Combining these properties with the question proposed in Remark 2.1, it motivates the convex optimisation problem.

Definition 2.3 (Convex optimisation problem)**.** The optimisation problem $\mathcal{P}$ is convex if in (2.1) the objective function f and the feasible set $\mathcal{C}$ are convex.

Remark 2.3. If we model $\mathcal{P}$ as an unconstrained problem with an extended-real-valued objective $\tilde{f}$ (2.2), the problem $\mathcal{P}$ is convex if $\tilde{f}$ is convex. In practice, in addition to the convexity of $\tilde{f}$, we focus on the optimisation problem defined by *Convex, Closed and Proper (CCP)* $\tilde{f}$, for the following reasons:

- *Properness*: $\tilde{f}$ has a non-empty domain and is never $-\infty$. This guarantees feasibility and existence of the optimal solution.
- *Closedness*: The epigraph of $\tilde{f}$ is closed (i.e. the convex $\tilde{f}(x)$ is lower-semicontinuous). This guarantees that the sublevel sets of $\tilde{f}(x)$ ($\{x \in \mathbb{R} | \tilde{f}(x) \leq c\}$) are closed, therefore the optimal solution x^* can be found within the domain.

Example 2.3. We list the standard forms of convex optimisation:

(a) *Linear Programming (LP):* Minimize $c^T x$, subject to $Ax \leq b$, $x \in \mathbb{R}^n$.

(b) *Quadratic Programming (QP):* Minimize $\frac{1}{2} x^T Q x + c^T x$, subject to $Ax \leq b$, $x \in \mathbb{R}^n$.

(c) *Second-Order Cone Programming (SOCP):* Minimize $e^T x$, subject to $\|A_i x + b_i\|_2 \leq c_i^T x + d_i$, $Fx = g$, $i = 1, 2, \ldots, m$, $x \in \mathbb{R}^n$.

(d) *Semidefinite Programming (SDP):* Minimize $\mathbf{tr}(CX)$, subject to $\mathbf{tr}(A_i X) \leq b_i$, $i = 1, 2, \ldots, m$, $X \succeq 0$.

(e) *Conic Programming (CP):* Minimize $c^T x$, subject to $Ax = b$, $x \in K$, where K is a convex cone.

They share the hierarchy: $LP \subseteq QP \subseteq SOCP \subseteq SDP \subseteq CP$.

2.1.2 Lagrangian duality

Lagrangian duality provides a different perspective to study optimisation problem (2.1). We *dualise* the constraints of the problem in (2.1) and define the *Lagrangian* function $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q \to \mathbb{R}$ as

$$\mathcal{L}(x, \lambda, \nu) := f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{q} \nu_j l_j(x), \tag{2.8}$$

in which we introduce the *Lagrangian multipliers / dual variables* $\{\lambda_i \geq 0\}, \{\nu_j \in \mathbb{R}\}$ to penalise the violation of the constraints in a relaxed fashion. In contrast to the objective function $f(x)$ of the *primal problem*, we further define the *Lagrangian dual function* $g : \mathbb{R}^m \times \mathbb{R}^q \to \mathbb{R}$ as

$$g(\lambda, \nu) := \inf_{x \in \mathcal{F}} \mathcal{L}(x, \lambda, \nu), \tag{2.9}$$

where $\mathcal{F} = \mathbf{Dom}(f) \cap \bigcap_{i=1}^{m} \mathbf{Dom}(h_i) \cap \bigcap_{j=1}^{p} \mathbf{Dom}(g_i)$. If we assume $\lambda_i \geq 0, \forall i$, and compare the dual function $g(\lambda, \nu)$ with the optimal primal objective value $p^* = f(x^*)$:

$$\begin{aligned} g(\lambda, \nu) \overset{(2.9)}{=} \inf_{x \in \mathcal{F}} \mathcal{L}(x, \lambda, \nu) \leq \mathcal{L}(x^*, \lambda, \nu) =& f(x^*) + \sum_{i=1}^{m} \underbrace{\lambda_i}_{\geq 0} \underbrace{h_i(x^*)}_{\leq 0} + \sum_{j=1}^{q} \nu_j \underbrace{l_j(x^*)}_{=0} \\ \leq& f(x^*) = p^*. \end{aligned} \tag{2.10}$$

On the other hand, $\forall x$, $\mathcal{L}(x, \lambda, \nu)$ in (2.8) can be viewed as an affine function with respect to λ and ν, therefore the dual function $g(\lambda, \nu) = \inf_{x \in \mathcal{C}} \mathcal{L}(x, \lambda, \nu)$ can be viewed as the pointwise minimisation over a group of affine functions. Hence as shown in Example 2.2, $g(\lambda, \nu)$ is concave (i.e., $-g(\lambda, \nu)$ is convex). Naturally we address the interest towards the maximisation of g and introduce the (convex optimisation) *dual problem* $\mathcal{D}$:

$$\begin{aligned} \underset{\lambda, \nu}{\text{maximise}} \quad & g(\lambda, \nu) \\ \text{subject to} \quad & \lambda \geq 0. \end{aligned} \tag{2.11}$$

We denote $d^* := g(\lambda^*, \nu^*)$ as the optimal value of the dual function and compare this value with the optimal primal objective $p^* = f(x^*)$. With (2.10), we have:

$$d^* = g(\lambda^*, \nu^*) \leq f(x^*) = p^*, \tag{2.12}$$

and this always-valid inequality is called *weak duality*. We denote the *duality gap* as the difference $p^* - d^*$. When the duality gap is zero (i.e., the equality holds in (2.12)), we say *strong duality* holds.

Remark 2.4 (Saddle-point interpretation of duality). An interesting perspective towards Lagrangian duality is via game theory. The optimal primal objective is equivalent to:

$$p^* = \inf_{x \in \mathcal{F}} \sup_{\lambda \geq 0,\, \nu} \mathcal{L}(x, \lambda, \nu). \tag{2.13}$$

On the other hand, the optimal dual objective is:

$$d^* = \sup_{\lambda \geq 0,\, \nu} \inf_{x \in \mathcal{F}} \mathcal{L}(x, \lambda, \nu). \tag{2.14}$$

From the *saddle-point theorem*, we have:

$$d^* = \inf_{x \in \mathcal{F}} \sup_{\lambda \geq 0,\, \nu} \mathcal{L}(x, \lambda, \nu) \leq \sup_{\lambda \geq 0,\, \nu} \inf_{x \in \mathcal{F}} \mathcal{L}(x, \lambda, \nu) = p^*, \tag{2.15}$$

which implies weak duality. Under strong duality, the equality holds in (2.15), and the optimiser pair (x^*, λ^*, ν^*) forms a saddle point of the minimax problem. This minimax problem can be interpreted as a continuous *zero-sum game* (one player with $x \in \mathcal{C}$ wants to minimise the payoff $\mathcal{L}(x, \lambda, \nu)$ and another one with $(\lambda \geq 0, \nu)$ wants to maximise it). The always valid inequality of weak duality is due to the fact that in a zero-sum game the player who goes after the other's decision will always take the advantage. When strong duality holds, a saddle-point pair (x^*, λ^*, ν^*) represents a *Nash Equilibrium* of the game, and there is no advantage to playing second.

Remark 2.5 (Geometric interpretation of duality). Lagrangian duality can also be interpreted geometrically. Figure 2.3 shows the duality of a simple problem $\min_x f(x)$, *s.t.* $h(x) \leq 0$. We construct $\mathcal{G} := \{(h(x), f(x)) \in \mathbb{R} \times \mathbb{R}^n | x \in \mathbf{Dom}(f) \cap \mathbf{Dom}(h)\}$ and the support (negative slope) hyperplane $h(\lambda) = \langle \lambda, u \rangle + t$ as the dual function (point-wise maximisation of the Lagrangian $g(\lambda) = \min_x \ f(x) + \langle \lambda, h(x) \rangle$). The primal optimal $p^* = t$ with $u = h(x) \leq 0$ (the lowest value of the right half-plane of $\mathcal{G}$), and the dual optimal $g^* = t$ with $u = 0$ (the largest value of this negative slope

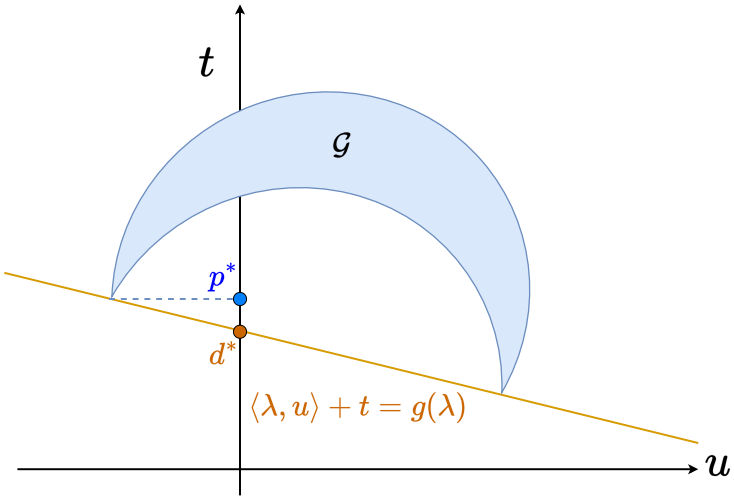


Figure 2.3: Example $\min_x f(x),\ s.t.\ h(x) \leq 0$ of a geometrical interpretation of Lagrangian dual as the support (negative-slope) hyperplane $h(\lambda) = \langle \lambda, u \rangle + t$ of $\mathcal{G} := \{(h(x), f(x)) \in \mathbb{R} \times \mathbb{R}^n | x \in \mathbf{Dom}(f) \cap \mathbf{Dom}(h)\}$. $d^* \leq p^*$ illustrates weak duality.

hyperplane with $u = g(x) \geq 0$). When the problem is not convex, $\mathcal{G}$ is usually not convex [49].

One way to guarantee strong duality is by *Constraint Qualifications (CQs)*[50], [51][2], among which a sufficient condition is *Slater's Condition*. It claims that for a convex optimisation problem formulated as (2.2), strong duality holds if $\exists x \in \mathbf{relint}(\mathcal{F})$ such that $h(x) < 0$ (there exists a *strictly feasible* point in the feasible set).

2.1.3 KKT conditions for optimality

From Remark 2.4, we obtain an insight: For an optimisation problem, the primal optimal p^* provides the upper bound of the dual function $g(\lambda, \nu)$ while the dual optimal d^* results in the lower bound of the primal objective. Therefore in Table 2.1 we conclude that, if the dual problem is unbounded (from above), then the primal problem is infeasible (from (2.2) the primal can only take the value of $+\infty$); alternatively, if the primal problem is unbounded (from below), then the dual problem is infeasible

[2]CQs may also guarantee other properties such as the existence and uniqueness of optimal dual variables.

(the dual can only take the value of $-\infty$).

Unbounded dual	Unbounded primal
$g(\lambda, \nu)$ can be $+\infty$	$f(x)$ can be $-\infty$
infeasible primal	infeasible dual

Table 2.1: Relationship between unboundedness and feasibility of Lagrangian duality.

We also observe, if strong duality holds, a saddle-point (x^*, λ^*, ν^*) must satisfy some conditions. From the definition of dual function, with $\lambda^* \geq 0$, $g(\lambda^*, \nu^*) = \inf_x \mathcal{L}(x, \lambda^*, \nu^*) > -\infty$; in order to attain the infimum, $0 \in \partial_x \mathcal{L}(x, \lambda^*, \nu^*)$; this can also be viewed as the supplementary condition of dual feasibility ($g(\lambda, \nu) > -\infty$) for general (x, λ, ν). On the other hand, in order to attain the supremum $\sup_{\lambda,\nu} \mathcal{L}(x^*, \lambda, \nu)$, $0 \in \partial_{\lambda,\nu}(\mathcal{L}(x^*, \lambda^*, \nu^*) + \mathcal{I}_{\lambda \geq 0}(\lambda^*))$, which is equivalent to the following (i) $\partial_\nu \mathcal{L}(x^*, \lambda^*, \nu^*) = l(x^*) = 0$, (ii) $\partial_\lambda \mathcal{L}(x^*, \lambda^*, \nu^*) = h(x^*) \leq 0$, and (iii) $\forall i,\ \lambda_i^* = 0$ or $\partial_{\lambda_i} \mathcal{L}(x^*, \lambda^*, \nu^*) = h_i(x^*) = 0$. We summarise these and obtain the Karush-Kuhn-Tucker (KKT) conditions for a saddle point (x^*, λ^*, ν^*) under strong duality, which are shown in Table 2.2.

Stationarity	$0 \in \partial_x \mathcal{L}(x^*, \lambda^*, \nu^*)$	attainment of $\inf_x \mathcal{L}(x, \lambda^*, \nu^*)$
Primal feasibility	$h(x^*) \leq 0,\ l(x^*) = 0$	attainment of $\sup_{\lambda,\nu} \mathcal{L}(x^*, \lambda, \nu)$
Dual feasibility	$\lambda^* \geq 0$	feasible set of dual problem
Complementary slackness	$\lambda_i^* h_i(x^*) = 0$ $\forall i = 1, 2, \ldots m$	attainment of $\sup_{\lambda,\nu} \mathcal{L}(x^*, \lambda, \nu)$

Table 2.2: KKT conditions for optimal saddle point (x^*, λ^*, ν^*).

We verify if KKT conditions are also sufficient conditions. If $\exists (x^*, \lambda^*, \nu^*)$ that satisfies KKT conditions, then

$$\begin{aligned} g(\lambda^*, \nu^*) &\overset{\text{(i)}}{=} f(x^*) + \sum_{i=1}^{m} \underbrace{\lambda_i^* h_i(x^*)}_{\overset{\text{(ii)}}{=} 0} + \sum_{j=1}^{p} \nu_i^* \underbrace{l_j(x^*)}_{\overset{\text{(iii)}}{=} 0} \\ &= f(x^*), \end{aligned} \tag{2.16}$$

in which (i) holds for stationarity, (ii) comes from complementary slackness, and (iii) is due to primal feasibility. In summary, the KKT conditions are as follows:

- *sufficient* for an optimal saddle-point (x^*, λ^*, ν^*) with strong duality,
- *necessary under strong duality* for the existence of such an optimal saddle-point (x^*, λ^*, ν^*).

2.2 Operator Theory and Fixed-point Algorithms

In the previous section we introduced convex functions that have desirable properties (Lemma 2.1), among which we have, for a convex, closed and proper (CCP) function $f(x)$, a local minimum x^* is a global minimum. This motivates the naïve (sub)gradient algorithm which acts as "to ski downhill until arriving at the valley" for a convex optimisation problem defined by a CCP objective function $f(x)$ as shown in Algorithm 2.1, in which k is the *step index*, $x^k \in \mathbb{R}^n$ is the *iterating variable*, α^k is the *step size*, and the condition $\|x^k - x^{k-1}\| \leq \epsilon$ is the *stopping criteria*. $T : \mathbb{R}^n \to \mathbb{R}^n$ is the *iterating operator*, and the algorithm can be viewed as the self-iteration of T until x^k approaches the *fixed-point* set $\mathbf{Fix}(T)$.

Algorithm 2.1 (Sub)gradient method

Input: x^0

Repeat:

1: $T(x^k) \leftarrow x^k - \alpha^k \partial f(x^k)$

2: $x^{k+1} \leftarrow T(x^k)$

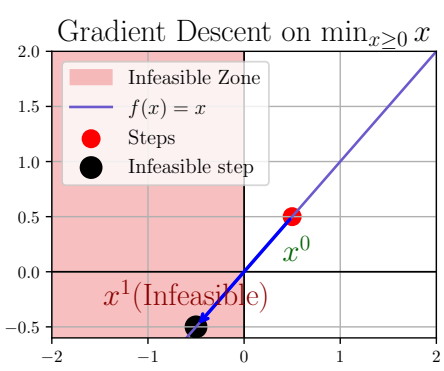
3: $k \leftarrow k + 1$

Until: $\|x^k - x^{k-1}\| \leq \epsilon$

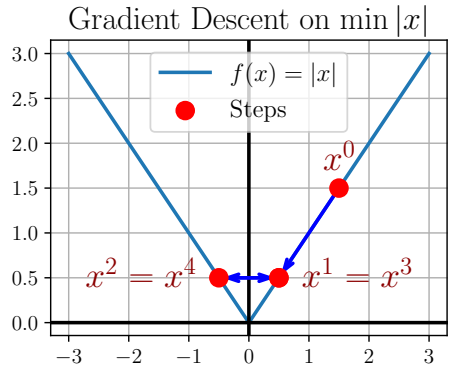
Output: x^k

Remark 2.6. However, such an algorithm may not work. For example, (i) $\partial f(x)$ may not be easily calculated; (ii) the iterate x^{k+1} may get out of the feasible set (the region where $\partial f(x^{k+1}) = +\infty$); in Figure 2.4(a) an example is shown of an out-of-boundary gradient step at $x = 1$ for $\min_{x \geq 0} x$ (iii) if the step size is not chosen properly, the iteration may not converge; in Figure 2.4(b) an example is shown of a non-converging gradient iteration of $\min_x |x|$ with a constant step size $\alpha = 1$. The framework of (algorithm-orientated) operator theory is therefore introduced to provide a method

to analyse the convergence behaviour of first-order algorithms, and this is the key methodology used in Chapter 4 and Chapter 5.



(a) Gradient Descent on $\min_{x\geq 0} x$ (Out of feasible set example).



(b) Gradient Descent on $\min |x|$ (Non-converging example).

Figure 2.4: Examples of failed gradient descent methods.

2.2.1 Operators and basic properties

Definition 2.4 (Operator)**.** An *operator (relation)* T on $\mathbb{R}^n$ is a point-to-set mapping characterised as:

$$(x, y) \in \mathbf{Graph}(T) := \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n | y \in T(x)\}, \tag{2.17}$$

in which $x, y \in \mathbb{R}^n$. If $T(x)$ is a singleton, then $y = T(x)$ is a (single-valued) function. The two notations $R(x)$ and Rx are used interchangeably.

Example 2.4. Here we provide some examples of operators.

- *Subgradient of a function* $f : \mathbb{R}^n \to \mathbb{R}$ at a point $x \in \mathbb{R}^n$ is defined as:

$$\partial f(x) = \{g | f(y) \geq f(x) + \langle g, y - x\rangle, \ \forall y \in \mathbb{R}^n\}. \tag{2.18}$$

  When $f(x)$ is convex, $\partial f(x) \neq \varnothing, \forall x \in \mathbf{Dom}(f)$. When a convex $f(x)$ is differentiable, $\partial f(x) = \nabla f(x)$, which is therefore a single-valued operator (function).

- *Identity* is a single-valued operator (function).

We can apply some operations to the operators. The *composition* of operator P and (single-valued) G is defined as:

$$PG(x) := \{z | y \in G(x),\ z \in P(y)\}. \tag{2.19}$$

Definition 2.5 (Inverse)**.** The *inverse* of an operator T is defined as:

$$T^{-1}(x) := \{y | x \in T(y)\}. \tag{2.20}$$

In general, we do not have $TT^{-1} = I$ as in the special case when T is a function (single-valued mapping). For a convex, closed and proper (CCP) function f, we have:

$$\partial f^{-1}(v) = \arg\min_x (f(x) - \langle v, x\rangle), \tag{2.21}$$

$$f(x) + f^*(v) = \langle x, v\rangle \quad \Leftrightarrow \quad \partial f^{-1}(v) = x \quad \Leftrightarrow \quad \partial f^{-1}(v) = \partial f^*(v), \tag{2.22}$$

where $f^*(v)$ is the convex conjugate (Example 2.2) of (CCP) $f(x)$, and $f^{**} = f$ [52]. This is consistent with the Fenchel-Young inequality ($f(x) + f^*(v) \geq \langle x, v\rangle$), which holds with equality for CCP $f(x)$.

Definition 2.6 (Monotone operator)**.** An operator T on $\mathbb{R}^n$ is said to be *monotone* if $\forall x, y \in \mathbb{R}^n$:

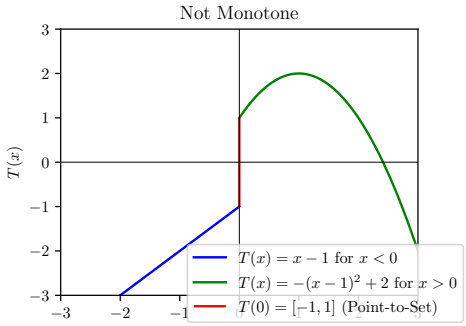
$$\langle u - v, x - y\rangle \geq 0, \tag{2.23}$$

where $u \in Tx$ and $v \in Ty$, and this can also be denoted as $\langle Tx - Ty, x - y\rangle \geq 0$. An operator T is *maximal monotone* if there is no monotone operator that properly contains it (i.e. There does not exist a monotone T' such that $\mathbf{Graph}(T) \subset \mathbf{Graph}(T')$.) An operator T is *strongly monotone* with parameter m if $\forall x.y \in \mathbb{R}^n$:

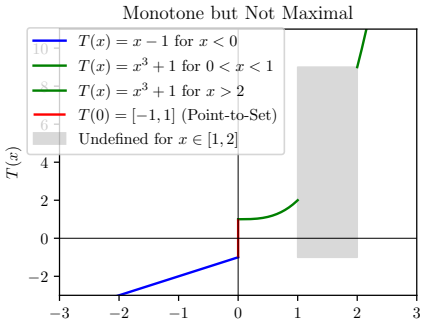
$$\langle Tx - Ty, x - y\rangle \geq m \|x - y\|^2. \tag{2.24}$$

In Figure 2.5 we provide 4 examples related to Definition 2.6.

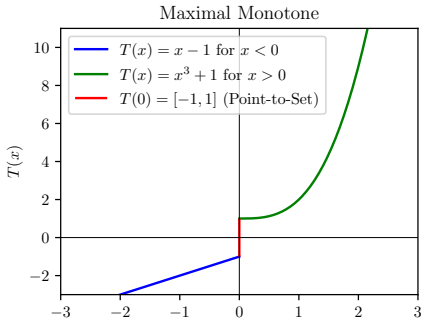
Lemma 2.2. *[10], [52], [53] For any function $f : \mathbb{R}^n \to \mathbb{R}$, the subdifferential $\partial f(x)$ is monotone, and this can be proven directly by applying the definition (2.18) of subgradient to $\forall x, y \in \mathbb{R}^n$. If f is convex, closed and proper (CCP), then $\partial f(x)$ is maximal monotone.*



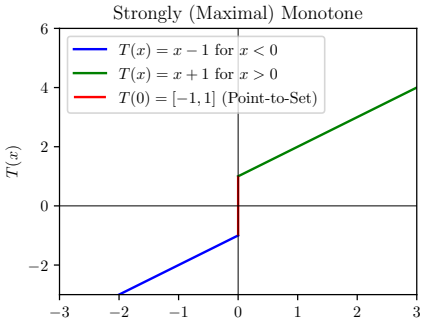
(a) When $x > 1$, $T(x)$ is not monotone.



(b) $T(x)$ can be contained by another monotone operator (e.g. Fig. 2.5(c)), hence is not maximal monotone.



(c) $T(x)$ is maximal monotone, but not strongly monotone since when $\forall x, y \to 0_+$, $\langle Tx - Ty, x - y \rangle \to 0$.



(d) $T(x)$ is strongly (maximal) monotone because $\forall x, y \in \mathbb{R}^n$, $\langle Tx - Ty, x - y \rangle \geq \|x - y\|^2$.

Figure 2.5: Examples regarding monotone operators.

The maximal monotone subgradient property of CCP functions plays a pivotal role in analysing first-order algorithms, especially for the nonexpansiveness and full domain properties of Cayley/Resolvent operators.

Lemma 2.3 (Strong convex functions [10], [54]–[57]). *For a CCP function $f : \mathbb{R}^n \to \mathbb{R}$, the following conditions are equivalent:*

(a) $f(x)$ *is strongly convex with parameter* m *(by definition/Jensen's inequality* $\forall x, y \in \mathbb{R}^n,\ \forall \lambda \in [0,1],\ f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) - \frac{1}{2}\lambda(1-\lambda)m\left\|x-y\right\|^2$*).*

(b) $\partial f(x)$ *is strongly monotone with parameter* m *(*$\forall x, y,\ \langle \partial f(x) - \partial f(y), x - y\rangle \geq m\left\|x-y\right\|^2$*).*

(c) $\forall y \in \mathbb{R},\ f(x) - \frac{1}{2}m\left\|x-y\right\|^2$ *is convex.*

(d) $\forall x, y \in \mathbb{R}^n,\ f(x) \geq f(y) + \langle \partial f(y), x - y\rangle + \frac{1}{2}m\left\|x-y\right\|^2$.

(e) If $f(x)$ *is twice continuous differentiable,* $\forall x \in \mathbb{R}^n,\ \nabla^2 f(x) - mI \succeq 0$.

The strongly monotone $\partial f(x)$ influences the contractiveness of the gradient descent operator and has a duality relationship with the Lipschitz condition of $\partial f(x)^{-1}$ $(= \partial f^*(x))$, which will be explained in Lemma 2.4 and Remark 2.7.

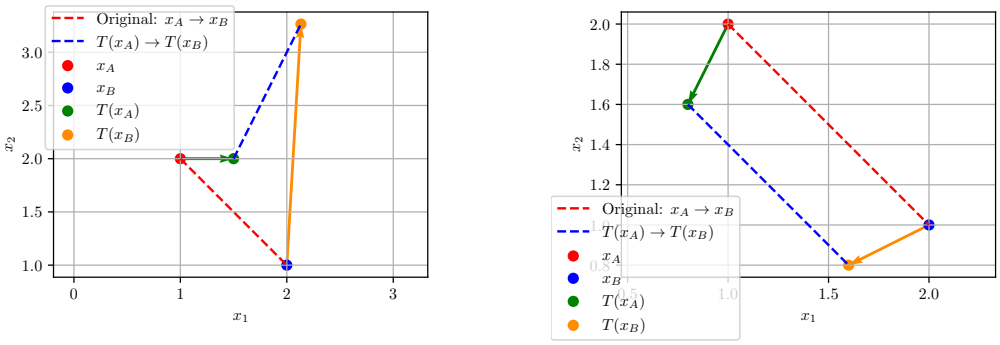
2.2.2 Nonexpansive operators and fixed-point iteration

In response to Remark 2.6, the framework of nonexpansive operators is provided to analyse the convergence of fix-point iterations.

Definition 2.7 (Nonexpansive operators). An operator $T(x)$ on $\mathbb{R}^n$ is *L-Lipschitz* if $\forall x, y \in \mathbb{R}^n,\ u \in T(x),\ v \in T(y)$:

$$\|u - v\| \leq L\|x - y\|. \tag{2.25}$$

If $T(x)$ is Lipschitz, then it degenerates to a single-valued function (When $x = y$, $u = v = T(x) = T(y)$ since $\|u - v\| \leq \|x - y\| = 0$.) When $L \leq 1$, $T(x)$ is *nonexpansive*. If $L < 1$, then $T(x)$ is *L-contractive*. (See Figure 2.6.)



(a) Nonexpansive $T(x)$. (b) Contractive $T(x)$.

Figure 2.6: Examples of nonexpansive operators, where $T(x)$ in (b) is contractive.

Lemma 2.4 (Strongly smooth function [10], [54]–[57]). *For a CCP function $f : \mathbb{R}^n \to \mathbb{R}$, the following conditions are equivalent:*

(a) $f(x)$ is L-smooth: $f(x)$ is differentiable and $\partial f(x) = \nabla f(x)$ is Lipschitz with parameter L ($\forall x, y \in \mathbb{R}^n$, $\langle \partial f(x) - \partial f(y), x - y \rangle \overset{(i)}{\leq} \|\partial f(x) - \partial f(y)\| \|x - y\| \leq L \|x - y\|^2$ where (i) holds for Cauchy-Schwartz inequality).

(b) $\forall x, y \in \mathbb{R}^n$, $\forall \lambda \in [0, 1]$, $f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda) f(y) - \frac{1}{2}\lambda(1 - \lambda) L \|x - y\|^2$.

(c) $\forall y \in \mathbb{R}$, $\frac{1}{2} L \|x - y\|^2 - f(x)$ is convex.

(d) $\forall x, y \in \mathbb{R}^n$, $\partial f(x) = \nabla f(x)$, $f(x) \leq f(y) + \langle \partial f(y), x - y \rangle + \frac{1}{2} L \|x - y\|^2$.

(e) If $f(x)$ is twice continuous differentiable, $\forall x \in \mathbb{R}^n$, $LI - \nabla^2 f(x) \succeq 0$.

Remark 2.7 (Strongly convex and strongly smooth functions). We compare Lemma 2.4 with Lemma 2.3 and obtain Table 2.3. If $f(x)$ on $\mathbb{R}^n$ is both m-strongly convex

$f(x)$ is m-strongly convex.	$f(x)$ is L-smooth.
$f(x) - \frac{1}{2} \|x\|^2$ is convex.	$\frac{1}{2} L \|x\|^2 - f(x)$ is convex.
$\partial f(x)$ is m-strongly monotone.	$\partial f(x) = \nabla f(x)$ is L-Lipschitz.
If $f(x)$ is twice continuously differentiable, $\nabla^2 f(x) - mI \succeq 0$.	If $f(x)$ is twice continuously differentiable, $LI - \nabla^2 f(x) \succeq 0$.

Table 2.3: Properties of strongly convex and strongly smooth functions.

and L-smooth, $\forall x, y \in \mathbb{R}^n$ we have:

$$m \|x-y\|^2 \leq \langle \partial f(x) - \partial f(y), x-y \rangle \leq \|\partial f(x) - \partial f(y)\| \|x-y\| \leq L \|x-y\|^2, \tag{2.26}$$

and $\kappa := L/m \geq 1$ is the *condition number* of $\partial f(x)$. Moreover, with $\partial f^*(x) = \partial f(x)^{-1}$ (2.22) we have: If a CCP $f(x)$ is strongly convex with parameter m, $f^*(x)$ is strongly smooth with parameter $L = 1/m$, and vice versa.

Definition 2.8 (Averaged operators)**.** For an operator $T(x)$ on $\mathbb{R}^n$, if $\exists \alpha \in (0,1)$,

$$T(x) = \alpha N(x) + (1-\alpha)x, \tag{2.27}$$

where $N(x)$ is nonexpansive, then $T(x)$ is α-*averaged*. If $\alpha = 1/2$ we say $T(x)$ is *firmly nonexpansive*.

Definition 2.9 (Fixed-point set and fixed-point iteration)**.** For a nonexpansive operator $T(x)$ on $\mathbb{R}^n$, its closed and convex *fixed-point set* is defined as

$$\mathbf{Fix}(T) := \{x \in \mathbb{R}^n | Tx = x\}, \tag{2.28}$$

The resulting *fixed-point iteration* of $T(x)$ is defined as

$$x^{k+1} := T(x^k), \tag{2.29}$$

where $k \in \mathbb{N}$ is the step index of the iteration.

Definition 2.10 (Fejér monotone [56], [58]–[60])**.** Let $\mathcal{C}$ be a convex and closed subset of $\mathbb{R}^n$. A sequence $\{x^k\}_{k\in\mathbb{N}}$ on $\mathbb{R}^n$ is Fejér monotone if $\forall x^* \in \mathcal{C}$:

$$\|x^{k+1} - x^*\| \leq \|x^k - x^*\|. \tag{2.30}$$

From the definition (2.25) of nonexpansive operators we know that the fixed-point iteration of a nonexpansive operator is Fejér monotone. Next we investigate the convergence of the fixed-point iteration of nonexpansive operators.

Lemma 2.5 (Browder's demiclosedness principle [61])**.** *For a nonexpansive operator T on $\mathbb{R}^n$, a sequence $\{x^n\} \in \mathbb{R}^n$, if $x^n \to x$ and $(T-I)x^n \to 0$, then $(T-I)x = 0$ (i.e., $x \in \mathbf{Fix}(T)$).*

Lemma 2.6 (Banach fixed-point theorem [62]). *For a contractive operator $T(x)$ on $\mathbb{R}^n$, the fixed-point set* $\mathbf{Fix}(T)$ *is a singleton.*

Lemma 2.7 (Convergence of the fixed-point iteration of nonexpansive operators [62], [63]). *For a nonexpansive operator $T(x)$, the fix-point iteration $x^{k+1} \leftarrow T(x^k)$ converges to* $\mathbf{Fix}(T)$ *if T is contractive or averaged.*

Proof. We choose $x^* = Tx^* \in \mathbf{Fix}(T)$. If $T(x)$ is contractive, with $L < 1$ in (2.25) we have:

$$\|x^{k+1} - x^*\| = \|Tx^k - x^*\| \leq L\|x^k - x^*\| \leq \cdots \leq L^k\|x^0 - x^*\| \overset{k\to\infty}{\to} 0, \tag{2.31}$$

and this iteration was proposed together with Lemma 2.6 in [62]. If $T(x)$ is α-averaged, let $\beta := \alpha^{-1} - 1$, we combine (2.25) as $L \leq 1$ and (2.27) to obtain:

$$\left\|Tx^k - x^*\right\|^2 + \beta\left\|(T-I)x^k\right\|^2 \leq \left\|x^k - x^*\right\|^2, \tag{2.32}$$

$$\Rightarrow \beta\frac{1}{K}\sum_{k=0}^{K-1}\left\|(T-I)x^k\right\|^2 \leq \frac{1}{K}\sum_{k=0}^{K-1}\left[\left\|x^k - x^*\right\|^2 - \left\|x^{k+1} - x^*\right\|^2\right] \overset{k\to\infty}{\to} 0,$$

$$\overset{\text{Lemma } 2.5}{\Rightarrow} \left\|x^0 - x^*\right\|^2 \leq \cdots \leq \left\|x^{k-1} - x^*\right\|^2 \leq \left\|x^k - x^*\right\|^2 \overset{k\to\infty}{\to} 0, \tag{2.33}$$

and this (*Mann's iteration*) was proposed in [63]. □

The framework of the fixed-point iteration of nonexpansive operators plays an important role in analysing first-order algorithms, and a detailed view focussing on $(T-I)x^k$) is elaborated in Chapter 5.

Example 2.5 (Gradient descent method [64]). In the light of Remark 2.6, we investigate the convergence behaviour of the (sub)gradient descent method (Algorithm 2.1), namely the fixed-point iteration:

$$x^{k+1} \leftarrow T_G(x^k), \tag{2.34}$$

where $T_G(x) = (I - \alpha\partial f)(x)$, for a CCP function $f(x)$ on $\mathbb{R}^n$, with $f(x)$ L-smooth ($\partial f(x) = \nabla f(x)$ is L-Lipschitz) and m-strongly convex ($m \geq 0$ and when $m = 0$, $f(x)$ is simply a convex function). For the gradient descent operator $T_G(x) = (I - \alpha\nabla f)(x)$

with constant step size $\alpha^k = \alpha$, by combining $\nabla T_G(x) = (I - \alpha\nabla^2 f)(x)$ (we assume the existence of $\nabla^2 f$ for simpler demonstration, but the result is also valid when $f(x)$ is not twice continuously differentiable) and $mI \leq \nabla^2 f(x) \leq LI$ listed in Table 2.3 we have:

$$\begin{aligned} I - \alpha m I \succeq \nabla T_G(x) = I - \alpha\nabla^2 f \succeq I - \alpha L I \\ \Rightarrow \|\nabla T_G(x)\| \leq \max\{|1 - \alpha L|, |1 - \alpha m|\} = L_G, \end{aligned} \tag{2.35}$$

in which L_G is the Lipschitz constant of T_G. In order to make T_G averaged, we need:

$$\alpha \in (0, 2/L). \tag{2.36}$$

Moreover, if $m > 0$, T_G is contractive and L_G is minimised at $\alpha = 2/(L + m)$. For the subgradient method with non-differentiable $f(x)$ and time-varying step sizes, we refer readers to [65].

If the objective $f(x)$ is subject to equality constraints $Ax = b$, by dualising this we may also apply the gradient descent method to the Lagrangian dual:

$$g(\lambda) \overset{(2.9)}{=} \inf_x f(x) + \langle \lambda, Ax - b\rangle = -(f^*(-A^\intercal\lambda) - \langle \lambda, b\rangle), \tag{2.37}$$

where $f^*(y) = \sup_x \langle x, y\rangle - f(x)$ is the convex conjugate (Example 2.2) of the objective $f(x)$. Therefore we have the *dual ascent* (gradient descent of $-g(\lambda) = -(Ax^*(\lambda) - b)$ where $x^*(\lambda) = \arg\min_x (f(x) + \langle \lambda, Ax - b\rangle)$) method [8], [66], [67]:

$$x^{k+1} \leftarrow \arg\min_x \left(f(x) + \langle \lambda^k, Ax - b\rangle\right), \tag{2.38a}$$

$$\lambda^{k+1} \leftarrow \lambda^k + \alpha(Ax^k - b). \tag{2.38b}$$

From Remark 2.7, in order to make $-g(\lambda)$ (2.37) L-smooth ($\partial_\lambda - g(\lambda) = \nabla_\lambda - g(\lambda) = -(Ax^*(\lambda - b))$ L-Lipschitz), $f(x)$ needs to be strongly convex with parameter m. In this case $L_{-\nabla g} = \frac{\|A\|^2}{m}$, and $\alpha \in (0, 2m/\|A\|^2)$ is needed for algorithm convergence under averagedness criteria (2.36). In Section 2.3 we investigate this dual ascent algorithm from the perspective of distributed optimisation.

Definition 2.11. (Resolvent and Cayley operators) For an operator $T(x)$ on $\mathbb{R}^n$ and $\alpha \in \mathbb{R}$, the *Resolvent* operator of T is defined as:

$$R_{\alpha T} := (I + \alpha T)^{-1}. \tag{2.39}$$

The Cayley operator of T is defined as:

$$C_{\alpha T} := 2R - I. \tag{2.40}$$

Lemma 2.8 (Resolvent and Cayley operators of maximal monotone operators [10], [56], [68]–[70])**.** *For a maximal monotone operator $T(x)$ on $\mathbb{R}^n$ and $\alpha > 0$ we have:*

(a) $\mathbf{Dom}(R_{\alpha T}) = \mathbf{Dom}(C_{\alpha T}) = \mathbb{R}^n$.

(b) *$R_{\alpha T}$ and $C_{\alpha T}$ are nonexpansive operators, and $R_{\alpha T}$ is $\frac{1}{2}$-averaged (firmly nonexpansive).*

(c) $0 \in T(x)$ *if and only if* $x \in \mathbf{Fix}(R_{\alpha T}) = \mathbf{Fix}(C_{\alpha T})$.

(d) *If T is m-strongly monotone, then R_T is L-contractive where $L = \frac{1}{1+\alpha m}$.*

(e) $(I - \alpha T)(x) = C_{\alpha T}(I + \alpha T)(x)$. *When T is single-valued,* $C_{\alpha T}(x) = (I - \alpha T)(I + \alpha T)(x)$.

As mentioned in Lemma 2.2, for a CCP function $f(x)$ on $\mathbb{R}^n$ the subdifferential $\partial f(x)$ is maximal monotone. Moreover, as stated in Lemma 2.3 if $f(x)$ is m-strongly convex, $\partial f(x)$ is m-strongly monotone. We combine these with Lemma 2.8 and investigate the Resolvent of the subgradient operator of CCP functions as follows.

Example 2.6 (Proximal operator [12], [31], [71]–[74]). For a CCP function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, the proximal operator is defined as:

$$\mathbf{prox}_{\alpha f}(x) = \arg\min_{v} \left(f(v) + \frac{1}{2\alpha} \|v - x\|^2 \right). \tag{2.41}$$

To attain this minimisation, the first-order stationary condition is satisfied:

$$0 \in \partial_v f(v) + (1/\alpha)(v - x) \Rightarrow x \in (I + \alpha \partial f)(v) \overset{(2.39)}{\Rightarrow} v = R_{\alpha \partial f}(x) = \mathbf{prox}_{\alpha f}(x), \tag{2.42}$$

which is always $\frac{1}{2}$-averaged for $\alpha > 0$, and contractive when ∂f is strongly monotone (Lemma 2.8). Therefore compared with gradient descent method (Example 2.5), which requires smooth $f(x)$ and careful choice of step size α, the fixed-point iteration of proximal operator (*proximal point method*)

$$x^{k+1} \leftarrow \mathbf{prox}_{\alpha f}\left(x^k\right) \tag{2.43}$$

always converges to the optimal value for any CCP $f(x)$ and for any $\alpha > 0$. Since $\mathbf{prox}_{\alpha f}(x) = R_{\alpha\partial f}(x)$ is the inverse of $I + \alpha\partial f$, the proximal point method is also called *backward method* compared to the *forward (gradient descent) method*. For time-varying proximal point method, we may use line search [75] to find λ^k, and [76]–[80] discuss the acceleration methods applied to the proximal operator.

Similar to gradient descent (Example 2.5), for the convex problem subject to equality constraints $Ax - b = 0$, we may also interested in the proximal operator of the dual function $\mathbf{prox}_{\alpha - g}(\lambda) = R_{\alpha - g}(\lambda)$ since $-g(\lambda)$ is convex, where $g(\lambda) \overset{(2.9)}{=} \inf_x f(x) + \langle \lambda, Ax - b\rangle$. Therefore we calculate the Resolvent $R_{\partial - g}(\lambda)$:

$$\begin{aligned} &u = R_{\partial - g}(\lambda) \overset{(2.39)}{\Leftrightarrow} \lambda = (I - \alpha\partial g)(u) \Leftrightarrow u = \lambda + \alpha(Ax^* - b),\ 0 \in \partial f(x^*) + A^\intercal u \\ \Rightarrow &0 \in \partial f(x^*) + A^\intercal(\lambda + \alpha(Ax^* - b)) \Rightarrow x^* = \arg\min_x \left(f(x) + \frac{\alpha}{2}\left\| Ax - b + \frac{\lambda}{\alpha}\right\|^2 \right), \end{aligned} \tag{2.44}$$

The resulting proximal point method applied to the dual $\lambda^{k+1} \leftarrow \mathbf{prox}_{\alpha - g}\left(\lambda^k\right)$ is:

$$x^{k+1} \leftarrow \arg\min_x \left(f(x) + \frac{\alpha}{2}\left\| Ax - b + \frac{\lambda^k}{\alpha}\right\|^2 \right), \tag{2.45a}$$

$$\lambda^{k+1} \leftarrow \lambda^k + \alpha(Ax^k - b). \tag{2.45b}$$

which is called the *method of multipliers* [72], [81], [82]. Compared with dual ascent (2.38) (gradient descent of the dual $-g(\lambda)$), this method (proximal point method of the dual) does not require a Lipschitz $\partial g(x)$ (which means an m-strongly convex primal $f(x)$) as well as a limited range of the step size $\alpha \in (0, 2m/\|A\|^2)$ (Example 2.5). However it does have limitations when applying to distributed optimisation, which is discussed in Section 2.3.

Example 2.7 (Projection and the normal cone operator). For a closed convex set $\mathcal{C} \subseteq \mathbb{R}^n$ and $\forall v \in \mathbb{R}^n$, the *projection* operator is defined as $\forall \alpha > 0$:

$$\mathbf{proj}_{\mathcal{C}}(v) := \arg\min_x \left(\mathcal{I}_{\mathcal{C}}(x) + \frac{1}{2\alpha} \|x - v\|^2 \right), \tag{2.46}$$

where $\mathcal{I}_{\mathcal{C}}(x)$ is the indicator function of $\mathcal{C}$, and (2.46) is exactly the special case of proximal operator (2.41) as $f(x) = \mathcal{I}_{\mathcal{C}}(x)$ (i.e., $\mathbf{proj}_{\mathcal{C}}(v) = \mathbf{prox}_{\alpha \mathcal{I}_{\mathcal{C}}}(v)$). From (2.42) we know that $\mathbf{prox}_{\alpha \mathcal{I}_{\mathcal{C}}}(v) = R_{\alpha \partial \mathcal{I}_{\mathcal{C}}}(v)$, and we define the *normal cone* operator as:

$$N_{\mathcal{C}}(x) := \partial \mathcal{I}_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathbf{int}(\mathcal{C}), \\ \{s \in \mathbb{R}^n \mid \forall y \in \mathcal{C}, \langle s, y - x \rangle \leq 0\} & \text{if } x \in \mathbf{bdry}(\mathcal{C}). \end{cases} \tag{2.47}$$

Note that, in (2.47), if $x \in \mathbf{bdry}(\mathcal{C})$, then $N_{\mathcal{C}}(x)$ is the set of all outward normal vectors of the hyperplanes tangential to $\mathcal{C}$ passing through x. In Figure 2.7, x_1 is inside $\mathcal{C}$, hence $N_{\mathcal{C}}(x_1) = 0$, which is a singleton. Also, x_2 is on the smooth part of the boundary, hence $N_{\mathcal{C}}(x_2)$ is a convex cone comprised of vectors of the same outward direction, whereas x_3 is on the non-smooth part of the boundary, hence $N_{\mathcal{C}}(x_3)$ is a convex cone of vectors of different outward directions. In Chapter 4 we use the normal

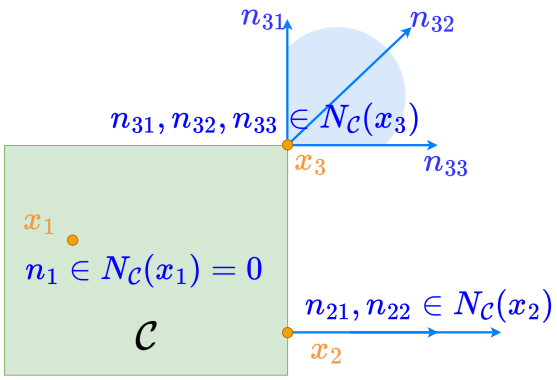


Figure 2.7: Normal cone operator $N_{\mathcal{C}}(x)$.

cone to restrict the range of optimal dual solution.

Remark 2.8. In this section we have discussed using the framework of nonexpansive operators to analyse first-order algorithms for convex optimisation problems. For a CCP $f(x)$ on $\mathbb{R}^n$, in order to find the optimal solution $0 \in F(x^*)$, we explored and applied the nonexpansiveness of the (sub)gradient descent operator $T_G = (I - \alpha \partial f)(x)$

and the proximal operator $R_{\alpha\partial f}(x)$. Table 2.4 provides a brief comparison between the two operators.

Gradient descent $T_G = (I - \alpha\partial f)(x)$	Proximal operator $\mathbf{prox}_{\alpha f}(x) = R_{\alpha\partial f}(x)$
∂f needs to be Lipschitz.	N/A
T_G is contractive when f strongly convex.	Same as T_G.
Can be applied to primal and dual.	Same as T_G.
Primal gradient has closed form.	Some have closed forms (See [12]).

Table 2.4: Comparison between gradient descent and the proximal operator.

2.2.3 Operator splitting

In practice we may have separable objectives ($f(x) = \sum_i f_i(x)$) and prefer to solve the optimisation problem in a distributed manner (See Section 2.3). Therefore, we want to find the optimal solution $0 \in \sum_i \partial f(x^*)$, which corresponds to $0 \in (\sum_i T_i)(x)$ where T_i is maximal monotone. *Operator splitting* provides methods to transform the problem into a fixed-point iteration comprised of the gradients, Resolvent and Cayley operators.

Lemma 2.9 (Composition of nonexpansive operators)**.** *For two nonexpansive operators $A(x)$ and $B(x)$ on $\mathbb{R}^n$, the composition $C(x) = A(B(x))$ is nonexpansive. Moreover*

(a) If A or B is contractive, C is contractive.

(b) If A and B are averaged, C is averaged.

Proof. We apply the definition of Lipschitz operators (2.25) and directly prove the nonexpansive and contractive case. For the averaged case see [11], [83]. □

Example 2.8 (Forward-backward splitting [84]). Suppose we want to find $x \in \mathbb{R}^n$ such that

$$0 \in (P + G)x, \tag{2.48}$$

where $P(x)$ and $G(x)$ are maximal monotone operators on $\mathbb{R}^n$. Let $\alpha > 0$, and rearrange this condition:

$$\begin{aligned} 0 \in (P+G)(x) &\Leftrightarrow 0 \in (I-I)x + \alpha(P+G)x \\ &\Leftrightarrow (I-\alpha G)x \in (I+\alpha P)x \\ &\Leftrightarrow x = (I+\alpha P)^{-1}(I-\alpha G)x = R_{\alpha P}(I-\alpha G)x. \end{aligned} \tag{2.49}$$

Thus we have reformulated (2.48) as the problem of finding the fixed-point of $Tx = R_{\alpha P}(I-\alpha G)x$. The resulting fixed-point iteration

$$x^{k+1} \leftarrow R_{\alpha P}(I-\alpha G)x^k \tag{2.50}$$

is called *forward-backward* splitting method [84], of which the name comes from the composition of a forward (gradient-like) operator and a backward (resolvent) operator. From Lemma 2.9 we know that when both operators are averaged, the composition is averaged. The resolvent $R_{\alpha P}$ is always $\frac{1}{2}$-averaged (Lemma 2.8); if $Gx = \partial g(x)$ where $g(x)$ is a CCP function on $\mathbb{R}^n$, $(I-\alpha G)$ is averaged if g is L-smooth and $\alpha \in (0, 2/L)$. If moreover $Px = \partial f(x)$ where $f(x)$ is a CCP function on $\mathbb{R}^n$, the resulting fixed-point iteration $x^{k+1} \leftarrow R_{\alpha f}(1-\alpha\nabla g)x^k = \mathbf{prox}_{\alpha f}\left((1-\alpha\nabla g)x^k\right)$:

$$z^{k+1} \leftarrow (I-\alpha\nabla g)x^k \tag{2.51a}$$

$$x^{k+1} \leftarrow \mathbf{prox}_{\alpha f}\left(z^{k+1}\right) \tag{2.51b}$$

is called the *proximal gradient method* [85]–[87]. In Section 2.3, this method is discussed from the point of view of distributed optimisation in practice.

Example 2.9 (Peaceman-Rachford and Douglas-Rachford splitting [88]–[91]). We also consider the problem of finding $x \in \mathbb{R}^n$ such that

$$0 \in (A+B)x, \tag{2.52}$$

where $A(x)$ and $B(x)$ are maximal monotone operators on $\mathbb{R}^n$. Given $\alpha > 0$ we have

$$\begin{aligned} 0 \in \partial(A+B)x &\overset{(2.48)}{\Leftrightarrow} 0 \in (I+\alpha A)x - (I-\alpha B)x \\ &\overset{\text{Lem. 2.8(e)}}{\Leftrightarrow} 0 \in (I+\alpha A)x - C_{\alpha B}(I+\alpha B)x \end{aligned}$$

$$
\begin{aligned}
&\overset{(2.39)}{\Leftrightarrow} C_{\alpha B}z \in (I+\alpha A)R_{\alpha B}z,\ x = R_{\alpha B}z \\
&\overset{(2.39)}{\Leftrightarrow} R_{\alpha A}C_{\alpha B}z = R_{\alpha B}z,\ x = R_{\alpha B}z \\
&\overset{(2.40)}{\Leftrightarrow} (\frac{1}{2}I + \frac{1}{2}C_{\alpha A})z = (\frac{1}{2}I + \frac{1}{2}C_{\alpha B})z,\ x = R_{\alpha B}z \\
&\Leftrightarrow z = C_{\alpha A}C_{\alpha B}z,\ x = R_{\alpha B}z. \qquad (2.53)
\end{aligned}
$$

From Lemma 2.9 we know $T_P = C_{\alpha A}C_{\alpha B}$ is nonexpansive. The resulting fixed-point iteration $x^{k+1} \leftarrow Tx^k$ defined by

$$x^{k+\frac{1}{2}} \leftarrow R_{\alpha B}z^k \qquad (2.54a)$$

$$z^{k+\frac{1}{2}} \leftarrow 2x^{k+\frac{1}{2}} - z^k \qquad (2.54b)$$

$$x^{k+1} \leftarrow R_{\alpha A}z^{k+\frac{1}{2}} \qquad (2.54c)$$

$$z^{k+1} \leftarrow 2x^{k+1} - z^{k+\frac{1}{2}}, \qquad (2.54d)$$

is called Peaceman-Rachford splitting (PRS) [88]–[90]. However, $C_{\alpha A}C_{\alpha B}$ is not necessarily averaged. We define the modified version,

$$T_D z := (\frac{1}{2}I + \frac{1}{2}C_{\alpha A})z,\ x = R_{\alpha B}z, \qquad (2.55)$$

and the resulting fixed-point iteration $z^{k+1} \leftarrow T_D z$:

$$x^{k+\frac{1}{2}} \leftarrow R_{\alpha B}z^k \qquad (2.56a)$$

$$z^{k+\frac{1}{2}} \leftarrow 2x^{k+\frac{1}{2}} - z^k \qquad (2.56b)$$

$$x^{k+1} \leftarrow R_{\alpha A}z^{k+\frac{1}{2}} \qquad (2.56c)$$

$$z^{k+1} \leftarrow x^{k+1} + z^k - x^{k+\frac{1}{2}}, \qquad (2.56d)$$

is the Douglas-Rachford splitting (DRS) [10], [31], [56], [89], [91]. From Lemma 2.8 we know that, when either A or B is strongly monotone, T_D is contractive, and the algorithm has linear convergence.

In the next section, we introduce the Alternating Direction Method of Multipliers (ADMM) algorithm that can be formulated as either the primal [92] or the dual [93] iteration of DRS. In Chapter 3 we propose an asynchronous ADMM algorithm with a computing-free data exchange server. In Chapter 4 we discuss the convergence of ADMM under uncertainty.

Example 2.10 (Other operator splitting methods). Examples of other operator splitting methods for general monotone operators are as follows.

- *Forward-backward-forward (FBF) splitting [56], [66], [94].* For maximal monotone operators A and B on $\mathbb{R}^n$ that A is L-Lipschitz, forward-backward-forward splitting addresses that $\forall \alpha \in (0, 1/L)$:

$$0 \in (A+B)x \Leftrightarrow x = T_{FBF}\left((I-\alpha A)R_{\alpha B}(I-\alpha A)+\alpha A\right). \tag{2.57}$$

  The resulting fixed-point iteration of FBF is:

$$x^{k+\frac{1}{2}} \leftarrow R_{\alpha B}(x^k - \alpha A x^k) \tag{2.58a}$$

$$x^{k+1} \leftarrow x^{k+\frac{1}{2}} - \alpha(Ax^{k+\frac{1}{2}} - Ax^k). \tag{2.58b}$$

- *Davis-Yin (three-operator) splitting [10], [95].* For maximal monotone operators A, B and C on $\mathbb{R}^n$ that C is single valued. $\forall \alpha > 0$ we have:

$$0 \in (A+B+C) \Leftrightarrow T_{DY} z = \left(\frac{1}{2}I + \frac{1}{2}\left(C_{\alpha A}(C_{\alpha B} - \alpha C R_{\alpha B}) - \alpha C R_{\alpha B}\right)\right) z, \tag{2.59}$$

  with $x = R_{\alpha B} z$. The resulting fixed-point iteration is:

$$x^{k+\frac{1}{2}} \leftarrow R_{\alpha B} z^k \tag{2.60a}$$

$$z^{k+\frac{1}{2}} \leftarrow 2x^{k+\frac{1}{2}} - z^k \tag{2.60b}$$

$$x^{k+1} \leftarrow R_{\alpha A}(z^{k+\frac{1}{2}} - \alpha C x^{k+\frac{1}{2}}) \tag{2.60c}$$

$$x^{k+1} \leftarrow z^k + x^{k+1} - x^{k+\frac{1}{2}}. \tag{2.60d}$$

There are other operator splitting methods focusing on convex optimisation problems applied to the automation engineering field, such as Chambolle-Pock (CP) [96], Split Bregman [97], Primal-Dual Hybrid Gradient (PDHG) [98], Condat–Vu Algorithm (Primal-Dual Splitting Method) [99], [100]. Among these, for CCP functions f, g and matrix M, to find $0 \in \partial f + M^\intercal \partial g(Mx)$, $\forall \alpha \in (0, 1/\|M\|)$ CP performs the iteration:

$$x^{k+1} \leftarrow R_{\alpha f}(x^k - \alpha M^\intercal z^k) \tag{2.61a}$$

$$z^{k+1} \leftarrow R_{\alpha g^*}(z^k + \alpha M(2x^{k+1} - x^k)). \tag{2.61b}$$

This iteration is used in image processing applications and is also studied in [98], [101], [102].

In Chapters 4 and 5 we analyse how the algorithms perform in uncertain environments via operator splitting methods.

2.3 Consensus-Oriented Distributed Optimisation

Recent advances in communication technologies and embedded systems have motivated the development of algorithms to coordinate intelligent agents in a distributed manner. Compared to centralised decision-making mechanisms, distributed optimisation algorithms [1]–[3] feature the following characteristics: participating agents iteratively solve local optimisation problems and share the necessary information with neighbouring agents, which is described via an update and communication protocol; such protocols can be designed to facilitate large-scale problems with privacy intrinsically protected. The optimal solution of the global problem is asymptotically obtained from the local iterates. However the challenges caused by limited local computational power and communication reliability may arise.

2.3.1 Formulation of the distributed optimisation problem

We formulate a convex distributed optimisation problem with the consensus decision variable $x \in \mathbb{R}^n$ by m participating processing agents as follows:

$$\begin{aligned} \text{minimise} \quad & \sum_{i=1}^{M} f_i(x), \\ \text{subject to} \quad & x \in \mathcal{C}, \end{aligned} \tag{2.62}$$

where each local cost function $f_i : \mathbb{R}^n \to \mathbb{R}$ and the constraint set $\mathcal{C}$ is convex and closed.

We also model the communication network among the agents at discrete time $k \geq k_0$ with a graph $\mathcal{G}^k = (U, E^k$, where $U = \{1, 2, \ldots, M\}$ is the set of the agents and E^k is the set of communicating directed edges at time k. We assume $(i, i) \in$

E^k, $\forall i \in U$, $\forall k$ as each agent has access to its own local data. The neighbourhood of agent i at time k is defined as the set of agents from whom incoming data are received: $\mathcal{N}_i^k := \{j | (j,i) \in E^k\}$. Further details on modeling multi-agent networks using graph theory can be found in [103].

2.3.2 Weighted Averaging

To solve the problem (2.62), we consider the distributed subgradient algorithm proposed in [5], [7]. A weighting matrix $\mathcal{W}^k \in \mathbb{R}^{M \times M}$ associated with the communication graph E^k is introduced to represent the reliability of information received, where $\mathcal{W}_{ij}^k > 0$ for all $j \in \mathcal{N}_i^k$, and $\mathcal{W}_{ij}^k = 0$ elsewhere. Each agent starts with an initial value $x_i^0 \in \mathcal{C}$ and iteratively performs the following updates:

$$v_i^k \leftarrow \sum_{j \in \mathcal{N}_i^k} \mathcal{W}_{ij}^k x_j^k, \tag{2.63a}$$

$$x_i^{k+1} \leftarrow \mathbf{proj}_{\mathcal{C}} \left(v_i^k - \alpha^k s_i^k \right), \tag{2.63b}$$

where $\mathbf{proj}_{\mathcal{C}}(\cdot)$ denotes the projection onto $\mathcal{C}$, $s_i^k \in \partial f_i(v_i^k)$ is a subgradient, and $\alpha^k > 0$ is the step size.

Theorem 2.1. *[5], [7] Under the following assumptions, the iterates generated by (2.63) converge to an optimal solution $x_i^k \to x^* \in X^*$ as $k \to +\infty$:*

(i) Bounded subgradients*: Each function $f_i : \mathbb{R} \to \mathbb{R}$ has bounded subgradients on the set $\mathcal{C}$: for all i and $x_i \in \mathcal{C}$, there exists S_i such that $\|\partial f_i(x)\| \leq S_i$.*

(ii) Weighting rule*: There exists a scalar $\eta \in (0,1)$ such that for all $i, j \in U$ and $k \geq k_0$, (a) $\mathcal{W}_{ii}^k \geq \eta$; (b) $\mathcal{W}_{ij}^k \geq \eta$ if $\mathcal{W}_{ij}^k > 0$.*

(iii) Double stochasticity*: For all $i, j \in U$, (a) $\sum_{j \in U} \mathcal{W}_{ij}^k = 1$; (b) $\sum_{i \in U} \mathcal{W}_{ij}^k = 1$.*

(iv) Step size rule*: The step size satisfies $\sum_{k=k_0}^{+\infty} \alpha^k = +\infty$ and $\sum_{k=k_0}^{+\infty} (\alpha^k)^2 < +\infty$.*

(v) Connectivity*: There exists an infinite sequence of times $\{k_0, k_1, k_2, \dots\}$, where $0 < k_l - k_{l-1} \leq B_t$, with $B_t \in \mathbb{Z}_+$, such that the union of graphs $\cup_{k_{l-1}}^{k_l - 1} \mathcal{G}^k$ is strongly connected for all $l \in \mathbb{Z}_+$.*

(vi) Existence of an optimal solution*: The optimal solution set* $X^* \neq \varnothing$.

Remark 2.9. Assumption (i) ensures the existence of bounded subgradients. Consider the convex objective:

$$f(x) = x \ln x - x, \tag{2.64}$$

subject to the constraint $\mathcal{C} = [-1, 1]$. The subgradient becomes unbounded as $x \to 0_+$. If $x^0 \in [-1, 1]$, the algorithm may fail. We will show that this behaviour is common in dual functions, making subgradient methods challenging for the *dual ascent* algorithm described in Section 2.3.4. Assumption (ii) guarantees uniform positivity of the weighting matrix. Assumption (iii) implies that, over time, each agent is equally influenced by and equally influences others, aligning with the equal weighting of f_i in (2.62). Assumption (iv), known as the *square summable but not summable* step size rule, is essential for convergence and is common in centralised subgradient algorithms [65]. Recent work [104] provides convergence analysis for more general step sizes. When the cost functions are Lipschitz smooth, constant step sizes can also be used with modified algorithms [105], [106]. Assumption (v) ensures strong connectivity within any time interval of length B_t. Finally, assumption (vi) guarantees the existence of an optimal solution.

The use of network communication to solve optimisation problems in parallel dates back to the works of [107], [108]. Subsequently, the *incremental method* was introduced, where the communication graph forms a directed cycle, and agents update sequentially. These early works are summarised in [109]. More recently, averaging consensus algorithms, which aim to achieve consensus among agents—such as averaging their initial values—have attracted significant attention. The technical primer [2] provides an overview of the evolution of weighted averaging algorithms. The works [4], [5] establish a connection between weighted averaging algorithms and the subgradient method, on which (2.63) is based. Several extensions of weighted averaging distributed optimisation methods can be found in the recent survey [3]. It is worth noting that the proposed time-varying algorithm still requires a global synchronous

clock. Asynchronous distributed optimisation algorithms will be discussed in Chapter 3.

2.3.3 Proximal Gradient Method

In Section 2.2, we proposed an operator framework to analyze first-order convex optimization algorithms. By reformulating the first-order optimality condition $0 \in \partial f(x^*)$ into the fixed-point condition of a nonexpansive operator, $T(z^*) = z^*$, the fixed-point iteration $z^{k+1} \leftarrow T(z^k)$ is performed to converge to the fixed-point set $\mathbf{Fix}(T)$, corresponding to the optimal solution. For separable objectives, splitting methods may be employed to facilitate distributed optimization. Example 2.8 introduced the forward-backward splitting method [84]. In this section, the proximal gradient method [12], [85]–[87] is applied to (2.62). Assume the objective function consists of a nonsmooth term and a smooth separable term:

$$\text{minimise} \quad h(x) + \sum_{i=1}^{M} f_i(x), \tag{2.65}$$

where $\forall i,\ f_i : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, $\sum_{i=1}^{M} f_i(x)$ is L-smooth, and $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a closed, convex, and proper (CCP) function, including the constraints in (2.62). For $\alpha \in (0, 2/L)$, the distributed proximal gradient method is given by:

$$g_i^k \leftarrow \nabla f_i(x^k), \tag{2.66a}$$

$$x^{k+1} \leftarrow \mathbf{prox}_{\alpha h}\left(x^k - \alpha \sum_{i=1}^{M} g_i^k\right). \tag{2.66b}$$

This method is widely used in distributed learning [110], [111], where $\{f_i(x)\}$ are local objectives, and $h(x)$ is the global regularization term.

If the local objectives satisfy $\forall i,\ f_i(x) = \tilde{f}_i(x_i)$ with $\tilde{f}_i : \mathbb{R}^{n_i} \to \mathbb{R}$ being L-smooth, and $x := [x_1, x_2, \ldots, x_M]^\intercal$, the algorithm becomes:

$$y_i^k \leftarrow x_i^k - \alpha \nabla \tilde{f}_i(x_i^k), \tag{2.67a}$$

$$x^{k+1} \leftarrow \mathbf{prox}_{\alpha h}\left(y^k\right), \tag{2.67b}$$

where $y := [y_1, y_2, \ldots, y_M]^\intercal$. When all local variables have the same dimension (i.e., $\forall i,\ n_i = n/M$), and $h(x) = \mathcal{I}_{\mathcal{C}}(x)$ where $\mathcal{C} = \{x \mid x_i = x_j,\ \forall i, j = 1, 2, \ldots, M\}$, with $\mathbf{prox}_{\alpha\mathcal{I}_{\mathcal{C}}}(y) = \mathbf{proj}_{\mathcal{C}}(y)$ (Example 2.7), (2.67) can be reformulated as:

$$y_i^k \leftarrow z^k - \alpha \nabla \tilde{f}_i(z^k), \tag{2.68a}$$

$$x^{k+1} \leftarrow \mathbf{proj}_{\mathcal{C}}\left(y^k\right) = \mathbf{1}_M \otimes \frac{1}{M} \sum_{i=1}^{M} y_i^k := \mathbf{1}_M \otimes z^k, \tag{2.68b}$$

where $\mathbf{1}_M \in \mathbb{R}^M$ is a vector of ones, and $\otimes$ denotes the Kronecker product. This is the projected gradient method [112], [113] for consensus-based distributed optimization, which can be viewed as a variation of Algorithm (2.63).

In practice, non-smooth local objectives may arise. The next section explores distributed algorithms leveraging dualisation techniques.

2.3.4 Dual Decomposition

Section 2.2 introduced the dual ascent method, presented as the gradient method for the dual function in Example 2.5. In this section, we explore the formulation of distributed optimisation using this approach. When the consensus decision variable x in the optimisation problem (2.62) is *separable*, we have the following:

$$\begin{aligned} \text{minimise} \quad & \sum_{i \in U} f_i(x_i), \\ \text{subject to} \quad & x_i \in X_i,\ \sum_{i \in U} A_i x_i = b, \end{aligned} \tag{2.69}$$

where $x_i \in \mathbb{R}^{n_i}$, $A_i \in \mathbb{R}^{p \times n_i}$, and $b \in \mathbb{R}^p$. We concatenate $[x_1, x_2, \ldots, x_M]^\top = x$, $\sum_{i \in U} A_i x_i = Ax$, and $\sum_{i \in U} f_i(x_i) = f(x)$. To solve this problem using averaging algorithms, each agent must maintain a local copy of all other agents' decision variables. This can lead to communication inefficiencies in large-scale problems and may raise privacy concerns. These challenges motivate the use of *dual decomposition*.

We dualise the coupling equality constraints in (2.69), extending the local cost functions by adding indicator functions for local constraints, as reformulated in (2.2), i.e., $\tilde{f}_i \leftarrow f_i + \mathcal{I}_{X_i}$ and $\tilde{f}(x) \leftarrow f(x) + \mathcal{I}_X$. The resulting Lagrangian is given by (2.8):

$$\mathcal{L}(x, \lambda) := \tilde{f}(x) + \lambda^\top (Ax - b) = \sum_{i \in U} \tilde{f}_i(x_i) + \lambda^\top \left(\sum_{i \in U} A_i x_i - b \right). \tag{2.70}$$

Minimising over the primal decision variables yields the (concave) dual function (2.9):

$$g(\lambda) := \inf_x \mathcal{L}(x,\lambda) = \sum_{i\in U} \inf_{x_i} \Bigg[\underbrace{\tilde{f}_i(x_i) + \lambda^\top (A_i x_i - \frac{b}{M})}_{:=\mathcal{L}_i(x_i,\lambda)} \Bigg] := \sum_{i\in U} g_i(\lambda). \tag{2.71}$$

Since $g(\lambda)$ can be computed distributively, this approach is termed *dual decomposition*. The corresponding dual problem is:

$$\text{maximise} \quad \sum_{i\in U} g_i(\lambda). \tag{2.72}$$

We derive the subgradient conditions as follows:

$$\text{If } x_i^* \text{ is found:} \quad 0 \in \partial_{x_i} \mathcal{L}_i(x_i^*, \lambda), \tag{2.73a}$$

$$\text{Then:} \quad A_i x_i^* - \frac{b}{M} \in \partial g_i(\lambda). \tag{2.73b}$$

The existence of x_i^* is not always guaranteed, as discussed in Remark 2.10. If *strong duality* holds (typically true for dualised affine constraints), a *centralised aggregator* can perform dual updates using the distributed *dual ascent* algorithm [8], [66], [67]:

$$x_i^{k+1} \leftarrow \arg\min_{x_i} \mathcal{L}_i(x_i, \lambda^k), \quad \forall i, \tag{2.74a}$$

$$\lambda^{k+1} \leftarrow \lambda^k + \alpha^k \left(\sum_{i\in U} A_i x_i^{k+1} - b \right), \tag{2.74b}$$

where α^k is the step size.

Remark 2.10. The subgradient must be bounded for convergence, meaning (2.74a) must be *solvable* and $A_i x_i^k$ must remain *bounded*. This is challenging for arbitrary dual variables, as $g(\lambda) = -\infty$ if λ is infeasible. For example, if $\tilde{f}(x) = e^x$ and $A = -1$, $-g(\lambda)$ takes the form of (2.64) ,and (2.74a) becomes unbounded for $\lambda \leq 0$. Similarly, if $\tilde{f}(x) = c^\top x$ (in linear programming), (2.74a) is typically unbounded for most λ values.

Remark 2.11. The step size in the dual ascent algorithm must be carefully chosen, since the dual function is not always Lipschitz. As shown in Example 2.6, the gradient descent algorithm for a non-Lipschitz objective with constant step sizes generally

yields convergence to a suboptimal solution ($\|x^k - x^*\| \leq \delta$). To have an exact optimal solution, *diminishing* step sizes, such as *square summable but not summable* sequences [65], are recommended to ensure convergence. Even then, the convergence rate may be as low as $\mathcal{O}(1/\sqrt{k})$.

Remark 2.12. To address the challenges discussed earlier, we impose *strong convexity* on $\tilde{f}_i(x_i)$ with modulus $\sigma > 0$. This guarantees a unique solution for (2.74a) for all $\lambda \in \mathbb{R}^p$. By replacing subgradients with gradients for $g(\lambda)$, we define $s^*(\cdot)$ as the subgradient satisfying (2.73a) and rewrite (2.73) for distinct λ^1 and λ^2:

$$0 \in s^*(\tilde{f}(x^{*2})) - s^*(\tilde{f}(x^{*1})) + A^\top(\lambda^2 - \lambda^1), \tag{2.75a}$$

$$\nabla g(\lambda^2) - \nabla g(\lambda^1) = A(x^{*2} - x^{*1}). \tag{2.75b}$$

With strong convexity, we obtain the following Lipschitz smoothness condition for $g(\lambda)$:

$$\begin{aligned} &\|\nabla g(\lambda^2) - \nabla g(\lambda^1)\|_2 = \|A(x^{*2} - x^{*1})\|_2 \leq \|A\|_2 \|x^{*2} - x^{*1}\|_2 \\ \leq &\frac{\|A\|_2}{\sigma} \|s^*(\tilde{f}(x^{*2})) - s^*(\tilde{f}(x^{*1}))\|_2 = \frac{\|A\|_2}{\sigma} \|A^\intercal(\lambda^2 - \lambda^1)\|_2 \leq \frac{\|A\|_2^2}{\sigma} \|\lambda^2 - \lambda^1\|_2, \end{aligned} \tag{2.76}$$

where $g(\lambda)$ is *Lipschitz smooth* with parameter $L_g = \|A\|^2/\sigma$. From Example 2.5 we know $I - \alpha L_{\nabla -g}$ is *nonexpansive* for $\alpha \in (0, 2/L_{\nabla -g}]$, and *averaged* for $\alpha \in (0, 2/L_{\nabla -g})$ (*firmly nonexpansive* when $\alpha \in (0, 1/L_{\nabla -g}]$) The gradient method achieves a convergence rate of $\mathcal{O}(1/k)$ for constant step sizes (See (2.32)), with potential acceleration to $\mathcal{O}(1/k^2)$ using Nesterov's method [78].

2.3.5 Method of Multipliers

We now pose the question: Can we smooth the dual function while ensuring that (2.74a) remains bounded, without requiring the strong convexity of $f(x)$? One promising approach is the *method of multipliers*, also known as the proximal point method for the dual, as described in Example 2.6.

We define the augmented Lagrangian for (2.70) with a penalty parameter $\theta > 0$:

$$\mathcal{L}_\theta(x, \lambda) \coloneqq \mathcal{L}(x, \lambda) + \frac{\theta}{2}\|Ax - b\|_2^2, \tag{2.77}$$

and the corresponding dual function:

$$g_\theta(\lambda) \coloneqq \inf_x \mathcal{L}_\theta(x, \lambda). \tag{2.78}$$

Remark 2.13. It can be verified that the primal and dual solutions remain unchanged compared to the original problem (2.69). The motivation for adding the penalty term in (2.77) is to smooth the dual function without requiring the strong convexity assumption in Remark 2.12. Furthermore, $-g_\theta(\lambda)$ is Lipschitz smooth with a parameter $L_{\nabla -g_\theta} = 1/(\theta + \sigma/\|A\|_2^2)$, where $\sigma \geq 0$. When $\sigma > 0$, it represents the modulus of $\tilde{f}(x)$, as noted in Remark 2.12. However, $\tilde{f}(x)$ is not necessarily strongly convex since σ can be zero.

Applying dual ascent to the modified dual problem gives:

$$x^{k+1} \leftarrow \arg\min_x \mathcal{L}_\theta(x, \lambda^k), \tag{2.79a}$$

$$\lambda^{k+1} \leftarrow \lambda^k + \theta\big(Ax^{k+1} - b\big), \tag{2.79b}$$

which leads to the *method of multipliers* [81], [82], [114], where a fixed step size θ is adopted.

Remark 2.14. From Example 2.6, we know this method is equivalent to the *proximal point method* for the dual, given by $\lambda^{k+1} \leftarrow \mathbf{prox}_{\theta g}(\lambda^k)$, which achieves an $\mathcal{O}(1/k)$ convergence rate. Here, the proximal operator $\mathbf{prox}_{\theta g}(\lambda) = R_{\theta g}(\lambda)$ is always $\frac{1}{2}$-averaged. From (2.79a), we observe:

$$\begin{aligned} 0 \in \partial_x \mathcal{L}_\theta(x^{k+1}, \lambda^k) &= \partial \tilde{f}(x^{k+1}) + A^\top \big(\lambda^k + \theta(Ax^{k+1} - b)\big) \\ &= \partial \tilde{f}(x^{k+1}) + A^\top \lambda^{k+1} = \partial_x \mathcal{L}(x^{k+1}, \lambda^{k+1}), \end{aligned} \tag{2.80}$$

which demonstrates that the method of multipliers is a *backward* method for the non-augmented dual problem. The update x^{k+1} ensures that λ^{k+1} is feasible for $g(\lambda)$.[3]

Remark 2.15. One might argue that solving (2.79a) could still be challenging since the augmentation term does not fully condition $x \notin \mathbf{null}(A)^\perp = \mathbf{row}(A)$. To address this, we can add a regularisation term $\frac{\gamma}{2}\|x - x^k\|_2^2$ to (2.77), resulting in the *proximal method of multipliers* [72], [115], which is the resolvent of the KKT operator $T(x, \lambda) = [\partial_x \mathcal{L}(x, \lambda), \partial_\lambda(-\mathcal{L}(x, \lambda))]^\top$ [10].

[3] In other words, x^{k+1} ensures $\lambda(k)$ is feasible for $g_\theta(\lambda)$ in (2.79a).

Remark 2.16. The main drawback of the method of multipliers is that it sacrifices distributed solvability of (2.79a) because the additional penalty term couples local decision variables. To address this, [116] proposed the *diagonal quadratic approximation* (DQA) method with a nested loop to solve (2.79a). More recently, [117] introduced the *Accelerated Distributed Augmented Lagrangian* (ADAL) method, which avoids the nested loop but requires a small constant step size constrained by the problem's degree of constraints. In the next section, we will present the *alternating direction method of multipliers* (ADMM) [1], [9], which adapts the augmented Lagrangian to distributed optimisation.

2.3.6 Alternating Direction Method of Multipliers (ADMM)

The ADMM algorithm addresses the following convex optimisation problem:

$$\min_{x,y} \quad h_1(x) + h_2(y), \tag{2.81a}$$

$$\text{subject to} \quad Ax + By - c = 0, \tag{2.81b}$$

where $h_1 : \mathbb{R}^{n_1} \to \mathbb{R} \cup \{+\infty\}$ and $h_2 : \mathbb{R}^{n_2} \to \mathbb{R} \cup \{+\infty\}$ are convex, closed, and proper (CCP) functions, and A, B are matrices of appropriate dimensions.

The augmented Lagrangian (2.77) for this problem is given by:

$$\mathcal{L}_\theta(x, y, \lambda) \coloneqq h_1(x) + h_2(y) + \lambda^\top (Ax + By - c) + \frac{\theta}{2}\|Ax + By - c\|_2^2, \tag{2.82}$$

where $\theta > 0$ is a penalty parameter.

To solve the problem, ADMM iteratively performs the following updates:

$$x \leftarrow \arg\min_x \mathcal{L}_\theta(x, y, \lambda), \tag{2.83a}$$

$$y \leftarrow \arg\min_y \mathcal{L}_\theta(x, y, \lambda), \tag{2.83b}$$

$$\lambda \leftarrow \lambda + \theta(Ax + By - c), \tag{2.83c}$$

where we omit the iteration index k for brevity.

ADMM guarantees convergence under certain conditions [1]. If problem (2.81) has at least one saddle point (x^*, y^*, λ^*), the following hold:

(i) The objective value at the iterates (x^k, y^k) converges to the optimal value.

(ii) The primal residual $Ax^k + By^k - c$ converges to zero.

(iii) The dual variable λ^k converges to a saddle point.

Remark 2.17. ADMM can be interpreted as the method of multipliers with inexact minimisations (compare (2.79a) with (2.83a) and (2.83b)), where the updates alternate in a Gauss-Seidel fashion. As discussed in Section 2.2, it can also be viewed as an application of Douglas-Rachford splitting [91], an averaged version of Peaceman-Rachford splitting [88], to either the primal [92] or dual [93] problem. However, even if the assumptions in [1] are satisfied, examples exist where ADMM fails due to unsolvability of (2.83a) or (2.83b) [118]. In general, averaged algorithms achieve an $\mathcal{O}(1/k)$ convergence rate, with linear convergence rates possible under additional assumptions. For more on convergence rates, see [119]–[122]. Although ADMM converges for any penalty parameter $\rho > 0$, the choice of ρ affects the convergence pattern. Adaptive penalty parameter and step size strategies have been proposed in [118], [123], [124].

Several extensions of the ADMM algorithm address various challenges. The work in [125] proposes an iterative algorithm for problems with additional global inequality constraints by dualising them. In [126], [127], the authors apply ADMM to the dual problem after dualising polyhedral constraints, allowing for inexact updates and time-varying networks. The authors of [21] combine the averaging algorithm from Section 2.3.2 with ADMM to achieve dual consensus tracking. For cost functions with coupling terms, convergence analysis is provided in [128]. The convergence behaviour of ADMM under pathological cases is studied in [129].

An interesting question is whether ADMM can be extended from two alternating blocks to multiple blocks. Convergence analyses for this question are found in [122], [130]–[133], with applications such as [134].

ADMM is well-suited for distributed optimisation since it can be designed for problems with block-separable quadratic terms $\|Ax + By - c\|_2^2$, provided $A^\top A$ and/or

$B^\top B$ are block-diagonal. For instance, many optimisation problems naturally split into two groups of objectives, as seen in [135], [136] for decentralised power distribution systems and [137] for mobile data offloading problems. In traffic control applications, multiple roles can be assigned to the same physical agent, such as vehicles serving as both motion planners and traffic mediators [138]–[141]. For non-convex collision avoidance constraints, [138] uses convexification via formation control, while [139]–[141] employ successive linearisation within either the model predictive control horizon or ADMM iterations.

When the functions h_1 or h_2 and the augmented Lagrangian possess a block-separable structure, ADMM can be applied to distributed optimisation. For Problem 2.69, we reformulate the problem as:

$$\begin{aligned} \text{minimise} \quad & \sum_{i\in U} \left(f_i(x_i) + \mathcal{I}_{X_i}(x_i)\right) + \mathcal{I}_Y(y), \\ \text{subject to} \quad & x - y = 0, \end{aligned} \tag{2.84}$$

where $Y = \left\{y \mid \sum_{i\in U} A_i y_i = b\right\}$. By dualising the coupling $x - y = 0$ the augmented Lagrangian for any $\theta > 0$ is given by:

$$\mathcal{L}_\theta(x, y, \lambda) \coloneqq \sum_{i\in U} \left(f_i(x_i) + \mathcal{I}_{X_i}(x_i)\right) + \mathcal{I}_Y(y) + \sum_{i\in U} \left(\lambda_i^\top (x_i - y_i) + \frac{\theta}{2}\|x_i - y_i\|_2^2\right), \tag{2.85}$$

where $y = [y_1, y_2, \ldots, y_M]$ and $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_M]$. The resulting ADMM algorithm (2.83) for distributed optimisation is:

$$x_i \leftarrow \arg\min_{x_i} \left(f_i(x_i) + \mathcal{I}_{X_i}(x_i) + \lambda_i^\top x_i + \frac{\theta}{2}\|x_i - y_i\|_2^2\right), \quad \forall i \in U, \tag{2.86a}$$

$$y \leftarrow \arg\min_{y} \left(\mathcal{I}_Y(y) - \lambda^\top y + \frac{\theta}{2}\|x - y\|_2^2\right), \tag{2.86b}$$

$$\lambda \leftarrow \lambda + \theta(x - y). \tag{2.86c}$$

Compared to (2.77), the augmented Lagrangian (2.85) is block-separable, enabling the development of a distributed optimisation algorithm with a centralised aggregator[4] via ADMM with greater solvability than the dual ascent method in (2.74).

[4] Agents perform (2.86a) distributively and the aggregator performs (2.86b) to enforce coupling constraints.

In Chapter 3, we provide a formulation for distributed optimisation with local coupling consensus and address the challenges of asynchronicity and privacy through a computation-free data exchange server. Chapter 4 discusses extensions of ADMM to manage parametric uncertainty.

2.4 Recursive Estimation and Optimal Control

Remark 2.18 (Engineering notation of probability). In this thesis, we adopt an engineering approach to probability, balancing simplicity and rigour, rather than relying on measure-theoretic formalism. We use x to denote both a random variable and a specific realisation of that variable, with the distinction made clear from the context. For example, we write $x \in \Omega$ to represent a random variable with sample space Ω, and $y = g(x)$ to denote a random variable defined as a function of x. Unless explicitly required, we do not differentiate between the event space and the sample space. Additionally with proper discretisation assumed, we adopt the engineering convention of treating the discrete sum $\sum \rho\,[x]$ as equivalent to the continuous integral $\int \rho\,[x]\,dx$ with the respective discrete or continuous probability density function (PDF) $\rho\,[x]$.

2.4.1 Recursive system identification and state estimation

In reality, there is always a gap between what we measure and what we estimate. Therefore, we use probability theory to analyse problems related to measurement and estimation. Suppose that we have a dynamic system with:

- $x^k \in \mathcal{X}$, a random vector of the state we want to estimate,
- $z^k \in \mathcal{Z}$, a vector-valued measurement that we observe.

The system dynamics is modelled as:

$$x^k = f^k(x^{k-1}, v^k), \tag{2.87a}$$

$$z^k = h^k(x^k, w^k), \tag{2.87b}$$

where $v^k \in \mathcal{V}$ and $w^k \in \mathcal{W}$ are independent noise, f^k in (2.87a) is the process dynamics, and h^k in (2.87b) is the measurement dynamics. Here the control input u^k is implicitly modelled in f^k and h^k for brevity.

We assume that the following are known before the iteration:

- f^k and h^k, the process and measurement dynamics.
- x^0, the initial state, which is independent of $v^k, w^k, \forall k$.
- $\rho\left[v^k\right]$ and $\rho\left[w^k\right]$, the PDF of independent noise.

Bayesian recursive tracking

At time k, we want to calculate $\rho\left[x^k|z^{1:k}\right]$ in a recursive way, assuming that $\rho\left[x^{k-1}|z^{1:k-1}\right]$ has been known at time $k-1$, where $z^{1:k} = \{z^1, z^2, \ldots, z^k\}$. From the total probability theorem we have $\forall x^k \in \mathcal{X}$:

$$\textbf{A priori update:} \quad \rho\left[x^k|z^{1:k-1}\right] \leftarrow \sum_{x^{k-1}\in\mathcal{X}} \rho\left[x^k|x^{k-1}\right] \rho\left[x^{k-1}|z^{1:k-1}\right], \tag{2.88}$$

where $\rho\left[x^k|x^{k-1}\right]$ can be calculated from the sum rule as $\forall x^k \in \mathcal{X}$:

$$\rho\left[x^k|x^{k-1}\right] = \sum_{\{v^k\in\mathcal{V}|f^k(x^{k-1},v^k)=x^k\}} \rho\left[v^k\right]. \tag{2.89}$$

In the next step, we calculate $\rho\left[x^k|z^{1:k}\right]$ via Bayes' law as $\forall x \in \mathcal{X}$:

$$\begin{aligned}\textbf{A posteriori update:} \quad \rho\left[x^k|z^{1:k}\right] &\leftarrow \rho\left[x^k|z^k, z^{1:k-1}\right] \\ &= \frac{\rho\left[z^k|x^k\right]\rho\left[x^k|z^{1:k-1}\right]}{\rho\left[z^k|z^{1:k-1}\right]},\end{aligned} \tag{2.90}$$

where since z^k and $z^{1:k-1}$ are independent given x^k, we calculate $\rho\left[z^k|z^{1:k-1}\right]$ from the total probability theorem:

$$\rho\left[z^k\right] = \sum_{x^k\in\mathcal{X}} \rho\left[z^k|x^k\right]\rho\left[x^k|z^{1:k-1}\right], \tag{2.91}$$

in which, similar to (2.89), $\rho\left[z^k|x^k\right]$ can be calculated from the sum rule as $\forall z \in \mathcal{Z}$:

$$\rho\left[z^k|x^k\right] = \sum_{w^k\in\mathcal{W}|z^k=h^k(x^k,w^k)} \rho\left[w^k\right]. \tag{2.92}$$

This is the *Baysian tracking* algorithm to recursively calculate $\rho\left[x^k|z^{1:k}\right]$.

In engineering applications, we usually only need an estimation of a random variable instead of the PDF. In the following, we discuss how to obtain meaningful and efficient estimations from PDFs.

Maximum likelihood estimation (MLE)

Suppose we have directly obtained the *likelihood* $\mathcal{L}\left[x^k; z^k\right] = \rho\left[z^k|x^k\right]$ as in (2.92). The *maximum likelihood* estimation of x^k is:

$$\hat{x}^k_{\text{MLE}} := \arg\max_{x^k \in \mathcal{X}} \rho\left[z^k|x^k\right]. \tag{2.93}$$

Example 2.11. If we have $z^k = Hx^k + w^k$ with $x^k \in \mathbb{R}^n,\ z^k, w^k \in \mathbb{R}^m,\ m > n,\ w^k \sim N(0, \Sigma_{w^k})$, the we have:

$$\begin{aligned} &\rho\left[z^k|x^k\right] \propto \exp\left(-\frac{1}{2}(z^k - Hx^k)^\intercal {\Sigma_{w^k}}^{-1}(z^k - Hx^k)\right) \\ \Rightarrow\ &\hat{x}^k_{\text{MLE}} = \hat{x}^k_{\text{MLE}} := \arg\max_{x^k \in \mathcal{X}} \rho\left[z^k|x^k\right] = (H^\intercal {\Sigma_{w^k}}^{-1} H)^{-1} H^\intercal {\Sigma_{w^k}}^{-1} z^k. \end{aligned} \tag{2.94}$$

MLE depends on the precise distribution of the measurement model, and if we have a linear measurement model and Gaussian noise as shown in Example 2.11, it has a closed linear form. However, MLE is vulnerable to non-identifiability with lack of data (in (2.93) the assumption $m > n$ is not always true), imperfect model and sharp outlier peaks. As shown in Figure 2.8, $\hat{x}^k_{MLE} = \rho\left[z^k|x^k = 5\right]$ but it is not a reasonable estimate of x^k.

Maximum a posteriori estimation (MAP)

If we have some prior knowledge of $\rho\left[x^k|z^{1:k-1}\right]$ as in (2.88) at time k, we can have the *maximum a posteriori estimation (MAP)* of x^k:

$$\hat{x}^k_{\text{MAP}} := \arg\max_{x^k \in \mathcal{X}} \rho\left[x^k|z^{1:k}\right] \overset{(2.90)}{=} \arg\max_{x^k \in \mathcal{X}} \rho\left[z^k|x^k\right] \rho\left[x^k|z^{1:k-1}\right]. \tag{2.95}$$

MAP heavily depends on the prior knowledge and clearly, if $\rho\left[x^k|z^{1:k-1}\right]$ is constant, $\hat{x}^k_{\text{MAP}} = \hat{x}^k_{\text{MLE}}$.

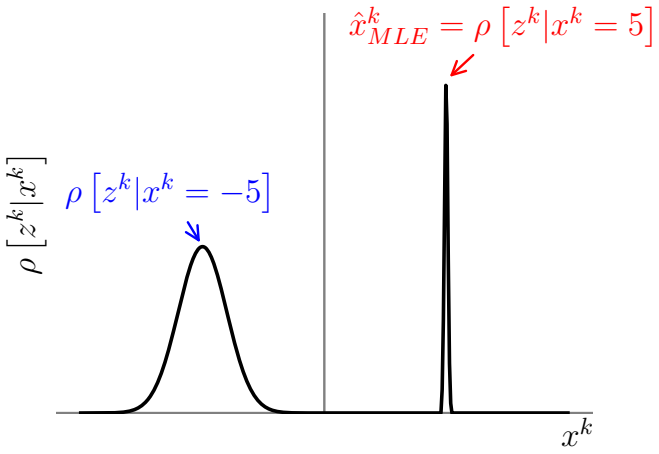


Figure 2.8: MLE is vulnerable to sharp outlier peaks: $\hat{x}^k_{MLE} = \rho\left[z^k|x^k = 5\right]$ however it is not a proper estimate of x^k.

Minimum mean squared error (MMSE) estimation

Instead of searching for the maximum point of the PDF of a posteriori as in MAP (2.95), the *minimum mean squared error (MMSE)* minimises the squared error and is defined as:

$$\begin{aligned}\hat{x}^k_{\text{MMSE}} &:= \arg\min_{\hat{x}^k} J^k(\hat{x}^k) = \arg\min_{\hat{x}^k} \mathbb{E}_{x^k|z^{1:k}}\left[\left\|\hat{x}^k - x^k\right\|^2\right] \\ &= \arg\min_{\hat{x}^k} \mathbb{E}_{x^k|z^{1:k}}\left[\left\|\hat{x}^k\right\|^2 - 2\langle \hat{x}^k, \mathbb{E}_{x^k|z^{1:k}}\left[x^k\right]\rangle + \left\|\mathbb{E}_{x^k|z^{1:k}}\left[\left\|x^k\right\|^2\right]\right\|^2\right] \\ &\overset{\text{(i)}}{=} \mathbb{E}_{x^k|z^{1:k}}\left[x^k\right], \end{aligned} \tag{2.96}$$

where (i) comes by setting $\frac{\partial J^k(\hat{x}^k)}{\partial \hat{x}^k} = 0$. Then we ask a question: is there a way to efficiently calculate the MMSE without calculating the PDFs?

Recursive least squares (RLS) for parameter identification

Suppose we have a static system (i.e. $f^k = I, \forall k$ in (2.87a)) with a linear measurement model (i.e. (2.87b) is linear as in Example 2.11:

$$z^k = H^k x + w^k, \tag{2.97}$$

where $x \in \mathbb{R}^n$ with $\mathbb{E}[x] = \mu_x$ and $\mathbf{Var}[x] = \Sigma_x$, $w^k \in \mathbb{R}^m$ with $\mathbb{E}\left[w^k\right] = 0$ and $\mathbf{Var}\left[x^k\right] = \Sigma_{w^k}$. Compared to Example 2.11 we do not assume $m > n$, instead we

assume $[H^1, H^2, \dots, H^k]^\intercal$ has full rank (i.e. the static system is observable), as is more common in practice. We want to design a linear estimator,

$$\hat{x}^k \leftarrow \hat{x}^{k-1} + K^k(z^k - H^k\hat{x}^{k-1}) \tag{2.98}$$

that recursively minimises the squared error, defined as for MMSE in (2.96) as

$$J^k(\hat{x}^k) := \mathbb{E}_{x|z^k}\left[\left\|x - \hat{x}^k\right\|^2\right] := \mathbb{E}_{x|z^k}\left[\left\|e^k\right\|^2\right] = \mathbb{E}_{x|z^k}\left[\mathbf{tr}\left(e^k(e^k)^\intercal\right)\right] := \mathbf{tr}\left(P^k\right), \tag{2.99}$$

where the error is $e^k := x - \hat{x}^k$ and its covariance is $P^k := \mathbb{E}_{x|z^{1:k}}\left[(e^k(e^k)^\intercal\right]$. Similar to the Luenberger observer [142], the error dynamics of (2.98) are

$$\mathbb{E}\left[e^k\right] = \left(I - K^kH^k\right)\mathbb{E}\left[e^{k-1}\right], \tag{2.100a}$$

$$P^k = \mathbb{E}\left[e^k(e^k)^\intercal\right] = \left(I - K^kH^k\right)P^{k-1}\left(I - K^kH^k\right)^\intercal + K^k\Sigma_{w^k}(K^k)^\intercal. \tag{2.100b}$$

If we set $\hat{x}^0 = \mu_x$ then e^k will remain zero, hence RLS is an unbiased estimator. Similar to (2.96), we solve $\partial J^k/\partial K^k = \partial\mathbf{tr}\left(P^k\right)/\partial K^k = \partial(2.100\text{b})/\partial K^k = 0$ for K^k and obtain

$$K^k = P^{k-1}H^k\left(H^kP^{k-1}(H^k)^\intercal + \Sigma_{w^k}\right)^{-1}. \tag{2.101}$$

We summarise above, set $\hat{x}^0, P^0$ as the guesses of μ_x, Σ_x, and obtain the *recursive least squares (RLS) [45], [143], [144]* (Algorithm 2.2) for parameter identification.

Algorithm 2.2 Recursive least squares (RLS) [45], [143], [144]

Input: $\hat{x}^0 = \mu_x, P^0 = \Sigma_x$

Repeat:

1: $K^k \leftarrow P^{k-1}(H^k)^\intercal\left(H^kP^{k-1}(H^k)^\intercal + \Sigma_{w^k}\right)^{-1}$ (2.101) (can be computed offline)

2: $\hat{x}^k \leftarrow \hat{x}^{k-1} + K^k(z^k - H^k\hat{x}^{k-1})$ (2.98) (to be computed online)

3: $P^k \leftarrow \left(I - K^kH^k\right)P^{k-1}\left(I - K^kH^k\right)^\intercal + K^k\Sigma_{w^k}(K^k)^\intercal$ (2.100b) (offline)

4: Output: $\hat{x}^k$ (update the estimates online)

As shown in Algorithm 2.2 RLS efficiently tracks $\hat{x}^k$, the MMSE of x^k, with only one online linear step (for $\hat{x}^k$) and two offline steps (for K^k and P^k) involving matrix inverses and quadratic forms. We output the estimate $\hat{x}^k$ recursively since each update

is meaningful. For RSL P^k usually converges to a finite bound around the true value of x instead of zero, due to the existence of measurement noise and unmodeled dynamics of the parameter as well as the measurement. In the sequel, we will discuss an MMSE filter that upgrades RLS from estimating static parameter states to the states with linear dynamics, namely the Kalman filter.

Kalman filter (KF) for state estimation

Suppose in (2.87) we have a linear system with noise defined as:

$$x^k = A^k x^{k-1} + B^k u^k + v^k, \tag{2.102a}$$

$$z^k = H^k x^k + w^k, \tag{2.102b}$$

where $x^k \in \mathbb{R}^n$ is the system state, $u^k \in \mathbb{R}^p$ is the control input, $z^k \in \mathbb{R}^m$ is the measurement, v^k, w^k are the process noise and measurement noise, and A^k, B^k, H^k are the time-varying matrices of system dynamics. We assume the following:

(1) $\{x^0, v^1, \ldots, v^k, w^1, \ldots, w^k\}$ are independent random variables.

(2) $x^0 \sim N(\mu_{x^0}, \Sigma_{x^0})$, and $v^k \sim N(0, \Sigma_{v^k})$, $\Sigma_{w^k} \sim N(0, \Sigma_{w^k})$ are random variables with normal distribution.

We define auxiliary variables x_p^k, x_m^k, and z_m^k ("p" for "prediction/prior" and "m" for "measurement") as follows. We set $x_m^0 = x^0 \sim N(\mu_{x^0}, \Sigma_{x^0})$ and $\forall k$:

$$x_p^k := A^k x_m^{k-1} + B^k u^k + v^k, \tag{2.103a}$$

$$z_m^k := H^k x_p^k + w^k, \tag{2.103b}$$

$$\rho_{x_m^k}[y] := \rho_{x_p^k | z_m^k}\left[y | z^k\right], \forall y, \tag{2.103c}$$

With similar approach as for (2.88) we can prove by induction [144]–[146] that $\forall k, \forall y$, x_p^k and x_m^k have the following distribution:

$$\rho_{x_p^k}[y] = \rho_{x^k | z^{1:k-1}}\left[y | z^{1:k-1}\right], \tag{2.104}$$

$$\rho_{x_m^k}[y] = \rho_{x^k | z^{1:k}}\left[y | z^{1:k}\right]. \tag{2.105}$$

We define two MMSE estimators $\hat{x}_p^k = \mathbb{E}\left[x_p^k\right]$, $\hat{x}_m^k = \mathbb{E}\left[x_m^k\right]$ as well as the variance $P_p^k = \mathbf{Var}\left[x_p^k\right], P_m^k = \mathbf{Var}\left[x_m^k\right]$. With the closure property under linear transformation of normal distribution, we use the following Kalman filter (KF) [45], [144] (Algorithm 2.3) to track $\hat{x}_p^k$, and $\hat{x}_m^k$ with iteration:

Algorithm 2.3 Kalman Filter (KF) [45], [144]

Input: $\hat{x}_m^0 = \mu_{x^0}, P^0 = \Sigma_{x^0}$

Repeat:

I: Prediction/A priori update

1: $\hat{x}_p^k \leftarrow A^k \hat{x}_m^k + B^k u^k$ (to be computed online)

2: $P_p^k \leftarrow A^k P_m^{k-1} (A^k)^\intercal + \Sigma_{v^k}$ (can be done offline)

3: Output: $\hat{x}_p^k$ (update the estimates online)

II: Measurement/A posteriori update

4: $K^k \leftarrow P_p^k (H^k)^\intercal \left(H^k P_p^k (H^k)^\intercal + \Sigma_{w^k}\right)^{-1}$ (can be done offline)

5: $\hat{x}_m^k \leftarrow \hat{x}_p^k + K^k (z^k - H^k \hat{x}_p^k)$ (to be computed online)

6: $P_m^k \leftarrow \left(I - K^k H^k\right) P_p^k \left(I - K^k H^k\right)^\intercal + K^k \Sigma_{w^k} (K^k)^\intercal$ (can be done offline)

7: Output: $\hat{x}_m^k$ (update the estimates online)

Similar to RLS, the Kalman filter has only two online steps (for $\hat{x}_p^k$ and $\hat{x}_m^k$) and other three linear algebra steps (for P_p^k, K^k, and P_m^k) involving the computing of matrix inverses and quadratic forms. In the 1960s, this enabled the space craft of the Apollo project to perform efficient online state estimates with the onboard computer that had only 32KB of RAM [147]. Comparing Algorithm 2.3 with Algorithm 2.2, we find that by limiting the states to static (setting $A^k = I, B^k = 0, \Sigma_{v^k} = 0$) in (2.102), the Kalman filter (with $x_p^k = x_m^{k-1}, P_p^k = P_m^{k-1}$) will reduce to RLS. Similar to RLS, the estimation error:

$$e^k := x^k - \hat{x}_m^k = \left(I - K^k H^k\right) A^{k-1} e^{k-1} + \left(I - K^k H^k\right) v^{k-1} - K^k w^k \tag{2.106}$$

is unbiased and $\mathbf{Var}\left[e^k\right] = P_m^k$.

If the system dynamics and the process and measurement noise statistics are time-invariant, so that $\{A^k, B^k, H^k, \Sigma_{v^k}, \Sigma_{w^k}\} = \{A, B, H, \Sigma_v, \Sigma_w\}$, then we may assume $P_p^k, P_m^k \overset{k\to\infty}{\to} P^\infty$ and the resulting $K^k \overset{k\to\infty}{\to} K^\infty$. By setting $P^k = P^{k-1} = P^\infty$ and

$K^k = K^{k-1} = K^\infty$ in (2.100b)(2.101), and solve the algebraic Ricatti equation:

$$P^\infty = AP^\infty A^\intercal + \Sigma_v - AP^\infty H^\intercal (HP^\infty H^\intercal + \Sigma_w)^{-1}, \tag{2.107a}$$

$$K^\infty = P^\infty H^\intercal (HP^\infty H^\intercal + \Sigma_w)^{-1} \tag{2.107b}$$

for the steady-state P^∞, K^∞. If we have stabilisable $(A, E), \Sigma_v = EE^\intercal$ and detectable (A, H), these equations have a unique positive semidefinite solution $P^\infty \succeq 0$, and $(I - K^\infty H)A$ (which defines the dynamics of the error as in (2.106)) is stable and $P^k \overset{k\to\infty}{\to} P^\infty$ [148].

2.4.2 Optimal control and Model Predictive Control (MPC)

In the previous section, we discussed the inductive-thinking part of a feedback controller, which uses available measurements (system outputs) to estimate model components (system parameters and states). In this section we focus on the deductive-thinking part, namely the controller.

Suppose we want to solve an optimal control problem described by:

$$\underset{u^{0:N}}{\text{minimise}} \quad J^0 = \sum_{0\le k\le N} \phi^k(x^k, u^k), \tag{2.108a}$$

$$\text{subject to} \quad x^{k+1} = f^k(x^k, u^k)\ \forall k \tag{2.108b}$$

In contrast to (2.1), the stagewise structure of this problem implies that it is a dynamic programming problem. In particular, its solution satisfies the *Principle of Optimality* [149], which states that, regardless of the initial state x^0 and decision u^0, the remaining decisions must constitute an optimal policy with regard to the state x^1 resulting from the first decision. This property is the basis of a stagewise solution method known as Dynamic Programming (DP) [149], which determines the optimal solution for u^k successively as the discrete time index counts backwards from $k = N$ to $k = 0$. Under mild conditions on ϕ^k and f^k, it can be shown that the optimal solution of (2.108) is a feedback control law of the form

$$u^k = \kappa^k(x^k). \tag{2.109}$$

Apart from certain special cases (e.g. f^k linear, ϕ^k quadratic, and no inequality constraints on x^k or u^k), the computation required for Dynamic Programming grows exponentially with the state dimension, a feature often referred to as the *curse of dimensionality*, making DP computationally intractable in general. This motivates an alternative solution method known as Model Predictive Control (MPC) [150].

In MPC, the optimal control is computed for a specific initial state, which we denote here as x. This allows the optimal control problem to be re-formulated as

$$\begin{aligned}
\underset{u^{0:N}}{\text{minimise}} \quad & J^0 = \sum_{0\le k\le N} \phi^k(x^k, u^k), && \text{(2.110a)}\\
\text{subject to} \quad & x^{k+1} = f^k(x^k, u^k)\ \forall k && \text{(2.110b)}\\
& x^k \in \mathcal{X}^k\ \forall k && \text{(2.110c)}\\
& u^k \in \mathcal{U}^k\ \forall k && \text{(2.110d)}\\
& x^0 = x && \text{(2.110e)}
\end{aligned}$$

(where for future reference we have explicitly included the state and input inequality constraints $(x^k, u^k) \in \mathcal{X}^k \times \mathcal{U}^k$). A consequence of this reformulation is that the optimal control law $u^0 = \kappa^0(x)$ can be computed by solving directly a numerical optimisation problem in the form of (2.1), thus potentially avoiding the curse of dimensionality. However the price to pay for this simplification is that in MPC the optimisation (2.110) must be solved online at each discrete time step given the current state x. The *prediction horizon* N is often fixed, in which case MPC is also known as receding horizon control. The stability properties of MPC can be established (under mild conditions on f^k, ϕ^k, $\mathcal{X}^k$, $\mathcal{U}^k$ [e.g. 150]) by considering the optimal value $J^0(x)$ of (2.110) as a Lyapunov (or Lyapunov-like) function of the system state x.

Model Predictive Control (MPC) without uncertainty

In classical MPC, which considers the case in which the model dynamics, system state, inequality constraints, and cost function are all known exactly, problem (2.110) is convex if f^k is linear, and ϕ^k and $\mathcal{X}^k$, $\mathcal{U}^k$ are respectively convex functions and convex sets for all k. We note that the more general case that f^k is nonlinear and

Lipschitz continuous can be handled using convex optimisation methods by making use of successive local linear model approximations with appropriate bounds $\mathcal{X}^k, \mathcal{U}^k$ and an appropriate min-max reformulation of the cost [e.g. 150].

Consider the case in which we have a linear dynamic system,

$$x^{k+1} = A^k x^k + B^k u^k, \tag{2.111}$$

quadratic stage cost,

$$\phi^k(x,u) = \tfrac{1}{2}\begin{bmatrix} x \\ u \end{bmatrix}^\top \begin{bmatrix} H^k_{xx} & H^k_{xu} \\ H^k_{ux} & H^k_{uu} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \begin{bmatrix} c^k_x & c^k_u \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}, \tag{2.112}$$

and polytopic (i.e. closed, convex polyhedral) constraint sets,

$$\mathcal{X}^k = \{x \in \mathbb{R}^{n_x} : F^k_x x \leq 1\}, \tag{2.113}$$

$$\mathcal{U}^k = \{u \in \mathbb{R}^{n_u} : F^k_u u \leq 1\}. \tag{2.114}$$

Then (2.110) can be expressed as a convex Quadratic Program taking the form

$$\underset{z}{\text{minimise}} \quad \tfrac{1}{2} z^\top Q z + c^\top z, \tag{2.115a}$$

$$\text{subject to} \quad Az \leq b, \tag{2.115b}$$

where $z = (u^0, u^1, \ldots, u^{N-1}) \in \mathbb{R}^{Nn_u}$, and the problem formulation has been *condensed* by eliminating the linear equality constraints (2.111) for $k = 0, 1, \ldots, N-1$.

MPC in practice: state and parameter estimation

In practice we typically only have estimates instead of the true parameters of the MPC optimisation problem. A case of particular importance is that in which the state x is not measured, but must be estimated. Moreover, the constraint sets $\mathcal{X}^k, \mathcal{U}^k$, stage cost ϕ^k, and model dynamics f^k may all contain unknown parameters that have to be estimated online using available information. As a result, the MPC optimisation (2.110) contains parameters that need to be estimated online, typically using filters derived from Recursive Least Squares or Kalman Filtering operations.

For example, consider the very common situation in which the state x is to be estimated, and suppose that the model dynamics are linear, the state and input

constraints are polytopic, and the cost is quadratic. Then the right hand side of the inequality constraint (b) and the linear term in the cost (c) in the MPC optimisation (2.115) both depend linearly on the state estimate.

2.4.3 Coupled decision-making system as the recursive estimation of the optimal solution

If the MPC optimisation problem contains parameters that need estimating using an iterative scheme that converges only asymptotically, a question of interest is whether an iterative solver can be employed *simultaneously* with an iterative estimator, as this would obviate the need to wait for the estimator to provide sufficiently accurate estimates before initiating the iterations of a solver for (2.115). In particular, for a certainty-equivalent approach in which the most recent estimate is used directly in the solver iterations, how does the estimator convergence rate and estimation uncertainty due to noise affect the solver convergence and its asymptotic solution accuracy? Given that first-order solvers are widely used in MPC due to their low per-iteration computational burden, this is one of the motivations behind the analysis performed in Chapters 4 and 5.

A second question concerns the stability of the combined estimator, MPC law and controlled system when noisy measurements and incomplete estimates are employed. The analysis of Chapters 4 and 5 allows one to consider the combination of estimator and solver as coupled dynamic systems, and thus provides an important avenue for answering this question.

MPC with parametric uncertainty

For example, consider the following optimisation problem:

$$\min_{y,z} \quad f^k(y) + g(z), \tag{2.116a}$$

$$\text{subject to} \quad y - z = 0, \tag{2.116b}$$

where

$$f^k(y) = \bar{f}(y) + \mathcal{I}_{\mathcal{F}_f^k}(y) = \frac{1}{2} y^\intercal Q y + c^\intercal y + \mathcal{I}_{\mathcal{F}_f^k}(y), \tag{2.116c}$$

$$\mathcal{F}_f^k: \quad Ay \leq b + D\hat{p}^k. \tag{2.116d}$$

This formulation can be interpreted as a model predictive control (MPC) problem with inexact estimates of polytopic constraint parameters—such as electricity load updated in real-time in load balancing problems [151]—modelled by a random variable $\hat{p}^k$, where k denotes the time index of recursive estimation (see [150, Ch. 2] for details on this MPC formulation). Any non-quadratic objectives or non-polytopic constraints may be embedded in $g(z)$, making (2.116) compatible with general optimal control problems.

Note that (2.116) involves two potentially non-Lipschitz[5] objective terms; hence, the proximal gradient method is inapplicable. Instead, we adopt ADMM to solve this problem. The central question becomes: can we combine the ADMM iteration with recursive estimation? As illustrated in Figure 2.9, k indexes both recursive estimation and ADMM iterations. The analysis in Chapter 4 outlines the conditions under which this combined algorithm converges in probability.

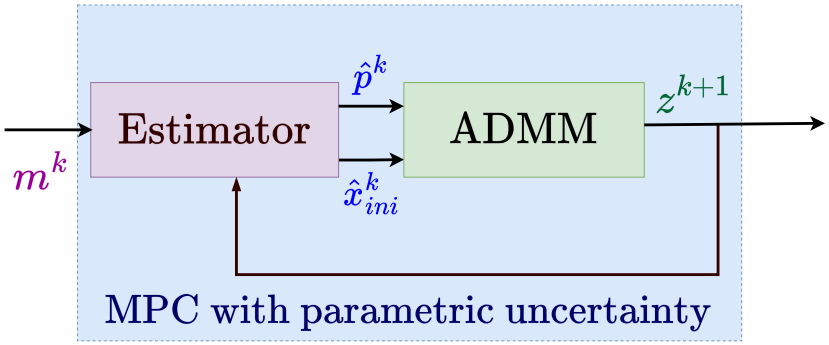


Figure 2.9: MPC with parametric uncertainty.

Recursive estimation of the optimal solution of a decision-making system

A natural follow-up question is: what if we replace problem (2.116) with a generic convex optimisation problem solved using any first-order algorithm? Suppose we consider:

$$\min_x \quad f(x, \hat{\theta}^k), \tag{2.117}$$

[5]The indicator functions of local constraints are non-Lipschitz.

where $\hat{\theta}^k$ is a stochastic estimate of problem parameters (e.g. state estimates, system identification parameters, or the objective/constraint parameters of an optimal controller), with k indicating the recursive update index. To solve this problem, we employ a general first-order algorithm of the form:

$$x^{k+1} \leftarrow T(x^k, \hat{\theta}^k),$$

as specified in Algorithm 2.4. The key question is whether Algorithm 2.4 converges. Chapter 5 analyses its convergence properties under different asymptotic behaviours of $\mathbf{Var}\left[\hat{\theta}^k\right]$—whether it vanishes or converges to a non-zero constant—within a stochastic operator framework.

Algorithm 2.4 Recursive fixed-point method

Input: x^0

Repeat:

1: Receive the latest parameter estimate $\hat{\theta}^k$
2: $x^{k+1} \leftarrow T(x^k, \hat{\theta}^k)$
3: $k \leftarrow k + 1$
4: Output x^{k+1}

We thus combine the recursive estimator of $\hat{\theta}^k$ with the first-order optimiser into a unified decision-making system, illustrated in Figure 2.10. This coupled system may be interpreted as a recursive estimator of the optimal solution to the coupled problem. Note that x^k and $\hat{\theta}^k$ are conditionally independent given the historical estimates $\{\hat{\theta}^1, \hat{\theta}^2, \ldots, \hat{\theta}^{k-1}\}$. Therefore, we conduct a convergence-in-probability analysis similar to that used in recursive estimation, using a stochastic operator framework.

In each clock cycle, the coupled system performs prior updates—such as the Kalman filter prior for $\hat{\theta}^k$ and parts of the optimisation step independent of $\hat{\theta}^k$—before the measurement m^k arrives. Upon receiving m^k, the remaining steps are executed as posterior updates. In Chapter 5, we introduce the concept of convex-combination invariance (CCI), and show that the estimator of the optimal solution to the coupled problem is asymptotically unbiased, i.e.,

$$\mathbf{dist}^2\left(\mathbb{E}\left[x^k\right], \mathbf{Fix}(\mathbb{E}_{x|\hat{\theta}}\left[T(x, \hat{\theta})\right])\right) \overset{k\to\infty}{\to} 0,$$

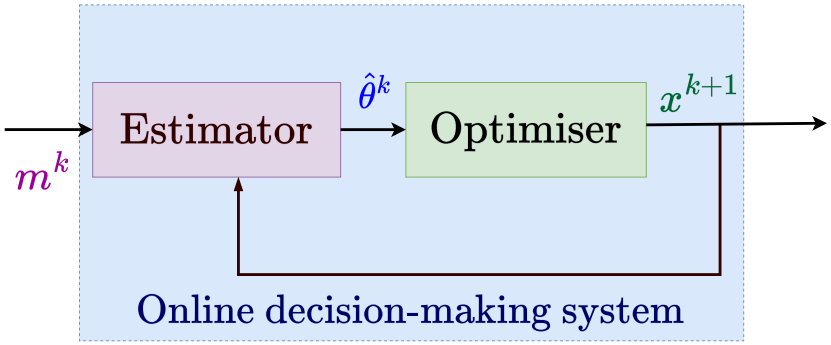


Figure 2.10: Online decision-making system.

if the optimiser is CCI. Here, $\mathbb{E}_{x|\hat{\theta}}\left[T(x,\hat{\theta})\right]$ denotes the overall mean operator under the conditional distribution of $\hat{\theta}$.

Chapter 3

Asynchronous ADMM via a Data Exchange Server

3.1 Introduction

Recent advances in communication technologies and embedded systems have motivated the development of algorithms to coordinate intelligent agents in a distributed fashion. In contrast to centralised decision-making mechanisms, distributed optimisation algorithms [1]–[3] feature participating agents iteratively solving local optimisation problems and sharing information with neighbouring agents, as specified by an update and communication protocol. Such protocols can be designed to facilitate the solution of large-scale problems with privacy intrinsically protected. The optimal solution of the global problem is asymptotically obtained from the local iterates.

However challenges may arise as a result of limited local computational power and communication reliability.

In general, distributed optimisation algorithms are designed to iteratively approach the optimal solution either by means of primal or primal-dual iterations. Motivated by [4], a class of distributed subgradient methods collaboratively estimates the common primal consensus via weighted averaging of local objective subgradient updates over a possibly time-varying network, with the capability to have global or local constraints [5], delays [7] and asymmetrical communication [6]. As the problem scales up, to keep a local copy of the entire consensus vector for every agent becomes prohibitive, hence dual decomposition algorithms [8] which allow agents to share only local variables become more advantageous. The alternating direction method of multipliers (ADMM) [1], [9] empowers the dual decomposition with an augmented Lagrangian to enhance the class of problems that can be tackled. For a convergence rate analysis we refer to [119]–[122].

As the number of agents increases, with delays becoming more prevalent due to distance, packet congestion and limited availability or capability of processors, synchronous algorithms may be challenged by a single straggler. In [13], the effectiveness of distributed machine learning over a stale synchronous server was discussed. This motivates us to explore distributed optimisation algorithms with delays. Instead of waiting until all agents are synchronised at each iterative step, each agent uses the most recent information at hand to compute the next update, hence achieving better efficiency with respect to the diminishing overall waiting time [14], [15]. However such optimisation algorithms have a natural limitation since the outdated data employed at each update step may cause an undesirable accumulation of error in solution estimates. The work of [14] shows that in general fixed-point algorithms such a trade-off can be favourable.

3.1.1 Related Work

The works [108], [109] investigate asynchronous optimisation algorithms applied to a collection of gradient-like methods. In [152] the authors focus on the delayed sub-

gradient method performed by a centralised coordinator, and in the later work [153] it is extended with an averaging consensus algorithm. The work of [154] extends these developments to stochastic convex optimisation problems. The dual gradient method for asynchronous distributed optimisation is explored in [155]. A framework for the convergence analyses of asynchronous fixed-point distributed optimisation algorithms is provided by [14], [156]. Addressing the need for parallel computing algorithms in the field of machine learning, [157], [158] study delay-tolerant gradient algorithms for distributed learning.

In this chapter we focus on asynchronous distributed optimisation via ADMM [15]–[20], [159]–[165]. The work of [16] studies ADMM with asynchronous updates and relates the almost sure convergence property to the case of synchronous ADMM. Randomised ADMM is introduced in [17] with randomised Gauss-Seidel iterations and convergence analysis via non-expansiveness. In [159] the authors propose an asynchronous ADMM algorithm with a centralised aggregator, and also provide an intuitive explanation for the convergence of the respective expected values provided that the agents have equal probability delivering the updates to the aggregator. Based on similar theoretical analyses, the works [18], [160], [162] propose hierarchical communication strategies for asynchronous ADMM. The works [15], [20], [163], [164] explore asynchronous ADMM with a centralised aggregator, and propose three algorithms whose convergence analyses are based on worst case bounded delay scenarios, to which our proposed algorithm is closely related. In [161] the authors propose a proximal and majorized approximation variant of ADMM, while the work [165] presents an incremental delayed-gradient variant, to enable the aggregator to cope with asynchrony and nonconvexity.

Recent studies [21]–[25] investigate the application of distributed optimization algorithms through ADMM, thus eliminating the need for a centralized aggregator. In [21], an averaging algorithm is used to achieve a consensus of the global primal residual and dual variable, thereby replacing the centralized aggregator. [24] presents a different approach, introducing a pairwise structure that is employed to compute

the bridging copies of local variables for relaxed ADMM. This method address problems that have a common global decision variable while also permitting asynchronous updates with probabilistic convergence. Further, [22] integrates an inner loop of a directed averaging algorithm. This strategy allows the distributed computation of the ϵ-consensus of the global decision variable. Similarly, [23] introduces an inner loop but for computing the finite-time exact ratio consensus. Finally, [25] tackles a bipartite optimal power flow problem with asynchrony via ADMM, utilizing learning algorithms to create replacements for missing updates.

3.1.2 Contribution

In this chapter we propose a decentralised asynchronous communication and update protocol that uses ADMM to solve a convex optimisation problem comprising two groups of local cost functions and constraints with local coupling consensus. The most closely related approach to our algorithm is [15, Algorithm 4], with the following main difference: we propose a data server working at its own clock cycles that handles asynchronous data exchange among agents with no computation involved, while in [15] the authors use a centralised aggregator to take charge of data exchange, a part of the primal variable update and the dual variable update. We also introduce local consensus blocks instead of a common consensus, as well as a vectorised augmentation parameter instead of a scalar one.

The chapter is organised as follows. Section 3.2 describes a distributed optimisation problem with local consensus constraints and a synchronous ADMM algorithm for its solution. Section 3.3 introduces the concept of a data exchange server in this context, explains the proposed asynchronous algorithm, and derives sufficient conditions on the problem and solver parameters for convergence. Section 3.4 presents a numerical study illustrating the theoretical results and provides a comparison with an alternative approach. Concluding remarks and directions for future work are provided in Section 3.5. Relevant proofs are included in Section 3.6.

3.1.3 Notation

The $n \times n$ identity matrix and the n-dimensional column vector with all elements taking the value 1 are denoted by I_n and $\mathbf{1}_n$, respectively. A symmetric positive definite (or positive semidefinite) matrix is denoted $A \succ 0$ (or $A \succeq 0$, respectively). We define $\|x\|_Q^2 := x^\intercal Q x$ for $Q \succeq 0$. The indicator function of a nonempty closed convex set $\mathcal{C}$ is denoted $\mathcal{I}_\mathcal{C}(x)$, where $\mathcal{I}_\mathcal{C}(x) = 0$ for $x \in \mathcal{C}$ and $\mathcal{I}_\mathcal{C}(x) = +\infty$ otherwise. $\partial F(x)$ indicates the subdifferential of function F evaluated at x. Rounding to the nearest integer is denoted $\lfloor \cdot \rceil$. $N_{tr}(\mu, \sigma^2, a, b)$ indicates the truncated normal distribution[1].

3.2 Problem Statement

3.2.1 ADMM Formulation

ADMM considers the following convex optimisation problem:

$$\min_{x,y} \quad h_1(x) + h_2(y), \tag{3.1a}$$

$$\text{subject to} \quad Ax + By - c = 0. \tag{3.1b}$$

where $h_1 : \mathbb{R}^{n_1} \to \mathbb{R} \cup \{+\infty\}$, $h_2 : \mathbb{R}^{n_2} \to \mathbb{R} \cup \{+\infty\}$ are convex, closed and proper functions; A, B are matrices of appropriate dimension. We construct the augmented Lagrangian:

$$\mathcal{L}_\rho := h_1(x) + h_2(y) + \lambda^\intercal (Ax + By - c) + \frac{1}{2}\|Ax + By - c\|_\rho^2, \tag{3.2}$$

in which $\rho \succ 0$ is a penalty parameter.

In order to solve the problem, ADMM iteratively performs the following updates:

$$x \leftarrow \min_x \mathcal{L}_\rho, \tag{3.3a}$$

$$y \leftarrow \min_y \mathcal{L}_\rho, \tag{3.3b}$$

$$\lambda \leftarrow \lambda + \rho(Ax + By - c). \tag{3.3c}$$

[1] If a random variable x has the normal distribution $N(\mu, \sigma^2)$ and $a < b$, then x conditional on $a \le x \le b$ follows $N_{tr}(\mu, \sigma^2, a, b)$. We specifically define $x \sim N_{tr}(\mu, \sigma^2, a, a)$ as $\mathbb{P}(x = a) = 1$.

ADMM guarantees [1] that if the problem (3.1) has a saddle point (x^*, y^*, λ^*), (i) the objective evaluated at the iterates of the primal variables (x, y) converge to its optimal value, (ii) the iterates of the primal residual $(Ax + By - c)$ converge to zero, and (iii) the iterates of the dual variable λ will converge to a saddle point.

3.2.2 Optimisation with Local Consensus

When the problems (3.3a) and (3.3b) are separable, they may be solved in a distributed manner. Here we propose a splitting scheme for distributed optimisation with local consensus. We consider a network of processing agents grouped by (i) $U := \{1, 2, 3 \dots M_U\}$ that solve separate problems in the form of (3.3a), and (ii) $V := \{1, 2, 3 \dots M_V\}$ that solve separate problems in the form of (3.3b). We assume (3.1b) has the special form of local coupling consensus constraints between the two groups, and use an undirected *bipartite graph* (bigraph) $\mathcal{G} = (U, V, E)$ to denote these relationships (see for example the illustration in Fig. 3.1). Thus the edge set E represents the local consensus couplings specified by the constraints (3.1b), which are formulated as the constraints (3.4c) below. We refer readers to [103] for a detailed description of modeling multi-agent networks using graph theory.

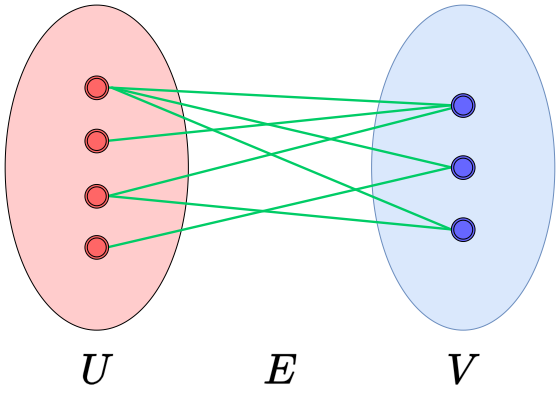


Figure 3.1: Problem bigraph $\mathcal{G} = (U, V, E)$.

We reformulate (3.1) to obtain the main problem $\mathcal{P}$ considered in this chapter:

$$\mathcal{P}: \min_{\substack{\{z_{ij},w_{ij}\}_{(i,j)\in E},\\ \{u_i\}_{i\in U},\,\{v_j\}_{j\in V}}} \sum_{(i,j)\in E} \Big(F_{ij}(z_{ij}) + G_{ij}(w_{ij})\Big) \tag{3.4a}$$

$$+ \sum_{i\in U} f_i(u_i) + \sum_{j\in V} g_j(v_j), \tag{3.4b}$$

subject to

$$z_{ij} = w_{ij}, \quad \forall (i,j) \in E, \tag{3.4c}$$

$$\big(u_i, \{z_{ij}\}_{j\in\mathcal{N}_i}\big) \in \mathcal{C}_i, \quad \forall i \in U, \tag{3.4d}$$

$$\big(v_j, \{w_{ij}\}_{i\in\mathcal{N}_j}\big) \in \mathcal{C}_j, \quad \forall j \in V, \tag{3.4e}$$

where $\mathcal{N}_i$ and $\mathcal{N}_j$ denote the sets of neighbours connected to agents i and j respectively. For each $i \in U$, $u_i \in \mathbb{R}^{p_i}$ is the local (private) decision variable and $z_{ij} \in \mathbb{R}^{m_{ij}}$ is the local consensus decision variable to be shared with $j \in \mathcal{N}_i$. Similarly, for each $j \in V$, $v_j \in \mathbb{R}^{p_j}$ is the local (private) decision variable and $w_{ij} \in \mathbb{R}^{m_{ij}}$ is to be shared with $i \in \mathcal{N}_j$. Constraint sets $\mathcal{C}_i$ and $\mathcal{C}_j$ represent inequality constraints that apply to u_i and v_i and their local consensus variables $\{z_{ij}\}_{j\in\mathcal{N}_i}$ and $\{w_{ij}\}_{i\in\mathcal{N}_j}$, respectively. Since the graph $\mathcal{G}$ is undirected, $j \in \mathcal{N}_i$ if and only if $i \in \mathcal{N}_j$.

Assumption 3.1. Problem $\mathcal{P}$ has the following properties:

(a). $\{F_{ij}, G_{ij} : \mathbb{R}^{m_{ij}} \to \mathbb{R}\}_{(i,j)\in E}$, $\{f_i : \mathbb{R}^{p_i} \to \mathbb{R}\}_{i\in U}$, and $\{g_j : \mathbb{R}^{p_j} \to \mathbb{R}\}_{j\in V}$ are convex objectives.

(b). $\mathcal{C}_i$ and $\mathcal{C}_j$ are convex local inequality constraint sets.

Remark 3.1. Several applications have the same structure as problem $\mathcal{P}$. For example, supply-demand pairs [151] in the scenario of market behaviour and individual-regulator pairs [141] in the scenario of resource allocation. We notice that ADMM [1] solves the problems which can be modelled in this bi-partite form of non-Lipschitz local objectives, and compared to [15] this formulation does not require a centralised aggregator hence end-to-end encryption can be established between agents.

Problem $\mathcal{P}$ in (3.4) is equivalent to (3.1) under the assignments:

- $h_1 = \sum_{i \in U} \big(\sum_{j \in \mathcal{N}_i} F_{ij}(z_{ij}) + f_i(u_i) + \mathcal{I}_{\mathcal{C}_i}(u_i, \{z_{ij}\}_{j \in \mathcal{N}_i}) \big)$
- $h_2 = \sum_{j \in V} \big(\sum_{i \in \mathcal{N}_j} G_{ij}(w_{ij}) + g_j(v_j) + \mathcal{I}_{\mathcal{C}_j}(v_i, \{w_{ij}\}_{i \in \mathcal{N}_j}) \big)$
- (3.1b) is equivalent to (3.4c).

A local sub-problem $\mathcal{P}_i$ is defined for each $i \in U$ as

$$\mathcal{P}_i : \min_{u_i, \{z_{ij}\}_{j \in \mathcal{N}_i}} f_i(u_i) + \sum_{j \in \mathcal{N}_i} F_{ij}(z_{ij}), \tag{3.5a}$$

$$\text{subject to: } \big(u_i, \{z_{ij}\}_{j \in \mathcal{N}_i}\big) \in \mathcal{C}_i, \tag{3.5b}$$

and likewise $\mathcal{P}_j$ is defined for each agent $j \in V$ as

$$\mathcal{P}_j : \min_{v_j, \{w_{ij}\}_{i \in \mathcal{N}_j}} g_j(v_j) + \sum_{i \in \mathcal{N}_j} G_{ij}(w_{ij}), \tag{3.6a}$$

$$\text{subject to: } \big(v_j, \{w_{ij}\}_{i \in \mathcal{N}_j}\big) \in \mathcal{C}_j. \tag{3.6b}$$

By dualising the coupling constraints (3.4c), we obtain the augmented Lagrangian of problem $\mathcal{P}$:

$$\begin{aligned} \mathcal{L}_\Theta := & \sum_{i \in U} \Big(f_i(u_i) + \sum_{j \in \mathcal{N}_i} F_{ij}(z_{ij}) \Big) \\ & + \sum_{j \in V} \Big(g_j(v_j) + \sum_{i \in \mathcal{N}_j} G_{ij}(w_{ij}) \Big) \\ & + \sum_{(i,j) \in E} \Big(\lambda_{ij}^\intercal (z_{ij} - w_{ij}) + \frac{1}{2} \| z_{ij} - w_{ij} \|_{\Theta_{ij}}^2 \Big). \end{aligned} \tag{3.7}$$

Here $\{\Theta_{ij}\}_{(i,j) \in E}$ is a set of penalty parameters that control the step size, and hence the convergence rate, of the Method of Multipliers applied to $\mathcal{P}$ (see e.g. [1] Sec. 3). Conditions on $\{\Theta_{ij}\}_{(i,j) \in E}$ to ensure convergence of the proposed asynchronous ADMM are identified in Section 3.3 and investigated numerically in Section 3.4. In this section we simply make the following assumption.

Assumption 3.2. $\Theta_{ij} \succ 0, \ \forall (i,j) \in E$.

We define $\mathcal{L}_\Theta^i, \forall i \in U$, and $\mathcal{L}_\Theta^j, \forall j \in V$ as:

$$\begin{aligned}\mathcal{L}_\Theta^i := f_i(u_i) + \sum_{j\in\mathcal{N}_i} \Big(&F_{ij}(z_{ij}) \\ &+ \lambda_{ij}^\intercal(z_{ij} - w_{ij}) + \frac{1}{2}\|z_{ij} - w_{ij}\|^2_{\Theta_{ij}}\Big),\end{aligned} \tag{3.8}$$

$$\begin{aligned}\mathcal{L}_\Theta^j := g_j(v_j) + \sum_{i\in\mathcal{N}_j} \Big(&G_{ij}(w_{ij}) \\ &+ \lambda_{ij}^\intercal(z_{ij} - w_{ij}) + \frac{1}{2}\|z_{ij} - w_{ij}\|^2_{\Theta_{ij}}\Big).\end{aligned} \tag{3.9}$$

Applying synchronous ADMM (3.3) to this problem results in Algorithm 3.1. Each iteration of this algorithm involves the following steps:

- In Step 1, each agent $i \in U$ solves problem $\mathcal{P}_i$ using $\{w_{ij}, \lambda_{ij}\}_{i\in\mathcal{N}_j}$ computed at the previous iteration and, for each $j \in \mathcal{N}_i$, sends the updated local consensus variable z_{ij} to agent j.
- Similarly, in Step 2, each agent $j \in V$ solves problem $\mathcal{P}_j$ using $\{z_{ij}, \lambda_{ij}\}_{i\in\mathcal{N}_j}$ computed at the previous iteration and sends the updated w_{ij} to agent i, for each $i \in \mathcal{N}_j$.
- In Step 3, all agents cooperatively update the Lagrange multipliers $\{\lambda_{ij}\}_{(i,j)\in E}$; hence the local iterates λ_{ij} of i and j are identical for all $(i, j) \in E$ at each iteration.

Assumption 3.3. Assume that:

(a). The Lagrangian (3.7) has at least one saddle point $\{u_i^*\}_{i\in U}$, $\{v_j^*\}_{j\in V}$, $\{z_{ij}^*, w_{ij}^*, \lambda_{ij}^*\}_{(i,j)\in E}$.

(b). All the U and V updates in Algorithm 3.1 have solutions for any inputs.

Remark 3.2. Assumption 3.3(b) is easily achieved since, for all $\forall i \in U$, $\mathcal{L}_\Theta^i$ is strongly convex in z_{ij} under Assumption 3.2, and the same reasoning applies to $\mathcal{L}_\Theta^j$, $\forall j \in V$.

Theorem 3.1. *Under Assumptions 3.1, 3.2, and 3.3, the iterates u_i, $\{z_{ij}, \lambda_{ij}\}_{j\in\mathcal{N}_i}$, $\forall i \in U$, and v_j, $\{w_{ij}, \lambda_{ij}\}_{i\in\mathcal{N}_j}$, $\forall j \in V$ of Algorithm 3.1 have the following convergence properties:*

Algorithm 3.1 Solve $\mathcal{P}$ via Synchronous ADMM

Initialise: z_{ij}, w_{ij}, λ_{ij}, Θ_{ij} $(\forall (i,j) \in E)$

Repeat:

1: U Update $(\forall i \in U$ in parallel$)$

Input: $\{w_{ij}, \lambda_{ij}\}_{j \in \mathcal{N}_i}$

Output:

$$u_i, \{z_{ij}\}_{j \in \mathcal{N}_i} \leftarrow \begin{array}{rl} \underset{u_i, \{z_{ij}\}_{j \in \mathcal{N}_i}}{\arg\min} & \mathcal{L}^i_\Theta\big(u_i, \{z_{ij}, w_{ij}, \lambda_{ij}\}_{j \in \mathcal{N}_i}\big) \\ \text{s.t.} & \big(u_i, \{z_{ij}\}_{j \in \mathcal{N}_i}\big) \in \mathcal{C}_i \end{array}$$

Each $j \in \mathcal{N}_i$ communicates $\{z_{ij}\}_{j \in \mathcal{N}_i}$ to the respective j

---

2: V Update $(\forall j \in V$ in parallel$)$

Input: $\{z_{ij}, \lambda_{ij}\}_{j \in \mathcal{N}_i}$

Output:

$$v_j, \{w_{ij}\}_{i \in \mathcal{N}_j} \leftarrow \begin{array}{rl} \underset{v_j, \{w_{ij}\}_{i \in \mathcal{N}_j}}{\arg\min} & \mathcal{L}^j_\Theta\big(v_j, \{w_{ij}, z_{ij}, \lambda_{ij}\}_{i \in \mathcal{N}_j}\big) \\ \text{s.t.} & \big(v_j, \{w_{ij}\}_{i \in \mathcal{N}_j}\big) \in \mathcal{C}_j \end{array}$$

Each $i \in \mathcal{N}_j$ communicates $\{w_{ij}\}_{i \in \mathcal{N}_j}$ to the respective i

---

3: E Update $(\forall i \in U$ and $\forall j \in V$ in parallel$)$

Input: $\{z_{ij}, w_{ij}, \lambda_{ij}\}_{(i,j) \in E}$

$$\lambda_{ij} \leftarrow \lambda_{ij} + \Theta_{ij}(z_{ij} - w_{ij})$$

Until satisfaction of the stopping criterion

Output:

4:

$$\begin{array}{ll} u_i, \{z_{ij}\}_{j \in \mathcal{N}_i} & \forall i \in U \\ v_j, \{w_{ij}\}_{i \in \mathcal{N}_j} & \forall j \in V \end{array}$$

(1) the objective of $\mathcal{P}$ evaluated at the local iterates converges to the optimal value;

(2) the primal residual $\sum_{\forall(i,j)\in E}(z_{ij} - w_{ij})$ evaluated at the local iterates converges to zero.

Remark 3.3. Algorithm 3.1 is required to synchronise communication between agents twice within each ADMM iteration. Therefore any unreliable peer-to-peer connections $(i, j) \in E$ will increase the waiting time needed per iteration. To avoid this problem and allow more flexible inter-agent communications, an asynchronous ADMM algorithm with a data exchange server is proposed in the following section.

3.3 Network Model and Proposed Asynchronous ADMM Algorithm

We introduce a *Data Exchange Server* to handle the shared data among the participating agents. Each agent is directly connected to the server via a communication link with a different round-trip time, as illustrated in Fig. 3.2.

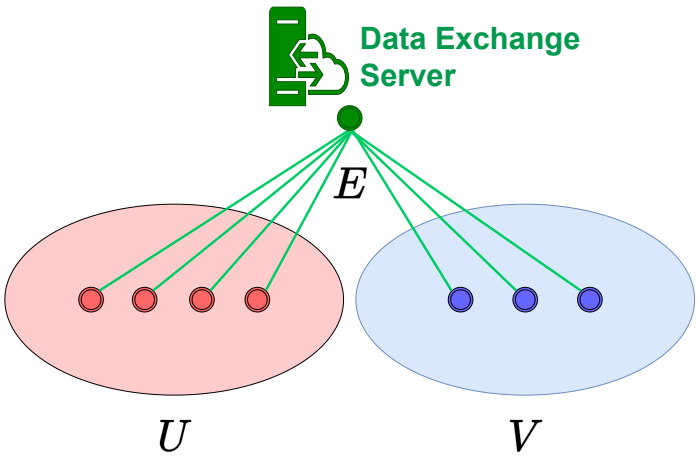


Figure 3.2: Network graph with a data exchange server.

The clock cycles of the data server are indexed by $k \in \{k_0, k_0+1, \ldots, -1, 0, 1, \ldots, K\}$, where $k_0 < 0$. During each clock cycle the server receives the data from an arbitrary set of agents. At the end of the clock cycle, the data server sends to each agent from which it received data in that cycle a set of data that it has received from the

Algorithm 3.2 Decentralised Asynchronous ADMM - (1/3) Data Exchange Server

Initialise: z_{ij}, w_{ij}, Θ_{ij} $\quad \forall (i,j) \in E$

$k = k_0 \leq -\max(\{\tau_i\} \cup \{\tau_j\})$

1: Send the initial data:

$\{\{w_{ij}^{\text{ini}}\}_{j \in \mathcal{N}_i}\} \quad \rightarrow \forall i \in U$

$\{\{z_{ij}^{\text{ini}}\}_{i \in \mathcal{N}_j}\} \quad \rightarrow \forall j \in V$

Repeat:

2: During clock cycle k:

Receive data from inbound agents:

$\{z_{ij}^k\}_{j \in \mathcal{N}_i} \leftarrow \{z_{ij}^{\text{in}}\}_{j \in \mathcal{N}_i} \qquad \forall i \in U \text{ s.t. } a_i^1(k) = k$

$\{w_{ij}^k\}_{i \in \mathcal{N}_j} \leftarrow \{w_{ij}^{\text{in}}\}_{i \in \mathcal{N}_j} \qquad \forall j \in V \text{ s.t. } b_j^1(k) = k$

For non-inbound agents:

$\{z_{ij}^k\}_{j \in \mathcal{N}_i} \leftarrow \{z_{ij}^{k-1}\}_{j \in \mathcal{N}_i} \qquad \forall i \in U \text{ s.t. } a_i^1(k) < k$

$\{w_{ij}^k\}_{i \in \mathcal{N}_j} \leftarrow \{w_{ij}^{k-1}\}_{i \in \mathcal{N}_j} \qquad \forall j \in V \text{ s.t. } b_j^1(k) < k$

3: At the end of clock cycle k:

Respond to the inbound agents with the data:

$\{\{w_{ij}^l\}_{j \in \mathcal{N}_i}\}_{l=a_i^2(k)+1}^{k} \rightarrow i \qquad \forall i \in U \text{ s.t. } a_i^1(k) = k$

$\{\{z_{ij}^l\}_{i \in \mathcal{N}_j}\}_{l=b_j^2(k)+1}^{k} \rightarrow j \qquad \forall j \in V \text{ s.t. } b_j^1(k) = k$

4: $k \leftarrow k + 1$

Until $k = K$, send terminating signal to all agents.

respective coupling agents. We refer the reader to Algorithm 3.2 for the details of how the data exchange server operates. To understand this algorithm:

- As shown in Fig. 3.3, $a_i^1(k)$ denotes the most recent clock cycle before the end of cycle k in which data from agent $i \in U$ arrived at the server, and $a_i^2(k)$ the next most recent one. Similarly, $b_j^1(k)$ denotes the most recent cycle before the end of cycle k in which data from agent $j \in V$ arrived at the server, and $b_j^2(k)$ the next most recent one. During the first few cycles when $a_i^2(k)$ and $b_j^2(k)$ are not defined, $a_i^2(k)$ and $b_j^2(k)$ are set to k_0, the initial clock cycle.
- The clock cycle counter is initialised as $k = k_0 \leq -\max(\{\tau_i\} \cup \{\tau_j\})$, where τ_i and τ_j are defined in Assumption 3.4 and represent available bounds on communication delays. This choice ensures that all variables have been updated at least once before $k = 1$ (see Algorithm 3.5) and also allows the window of a

running average output to be adjusted by tuning k_0 and K.

- The algorithm starts when the initial data is sent to the respective agents in Step 1.
- During each clock cycle k, the server passively collects the consensus updates from the agents as shown in Step 2. The server receives consensus updates $\{z_{ij}^{\text{in}}\}, \{w_{ij}^{\text{in}}\}$ from inbound agents, namely $i \in U$ such that $a_i^1(k) = k$ and $j \in V$ such that $b_j^1(k) = k$. These saved as the recorded updates $\{z_{ij}^k\}, \{w_{ij}^k\}$. For the rest of the (non-inbound) agents whose consensus updates have not arrived at the server during clock cycle k, represented by $a_i^1(k) < k$ or $b_j^1(k) < k$, the server saves duplicates of the consensus data of the previous cycle as the recorded updates.
- At the end of clock cycle k, the server responds to all the inbound agents with all the historical recorded updates of their counterparts, to which they are connected by the respective edge in E, since the last communication, as shown in Step 3.

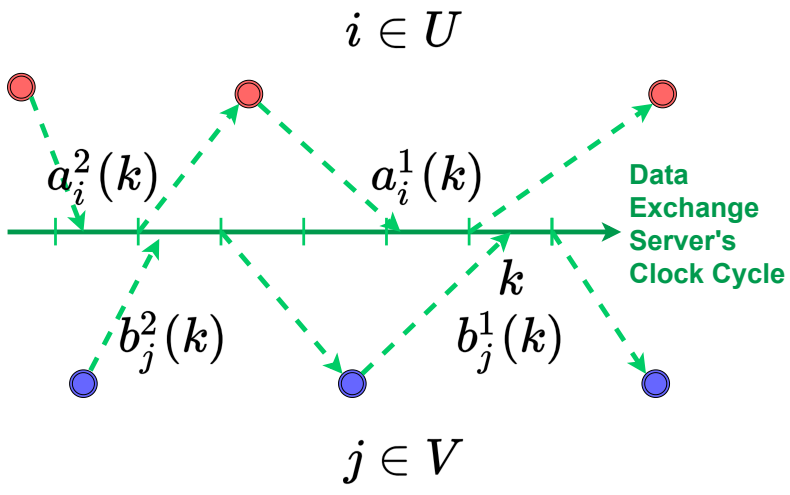


Figure 3.3: Clock cycles of the data exchange server.

Algorithm 3.3 Decentralised Asynchronous ADMM - (2/3) $\forall i \in U$ in parallel

Repeat:

1: Receive data from server: $\{\{w_{ij}^l\}_{j\in\mathcal{N}_i}\}_{l=1}^L$

2: $\lambda_{ij} \leftarrow \lambda_{ij} + \sum_{l=1}^{L-1} \Theta_{ij}(z_{ij}^{c-1} - w_{ij}^l)$

$$+ \ \Theta_{ij}(z_{ij}^c - w_{ij}^L) \qquad \forall j \in \mathcal{N}_i$$

3: $u_i, \{z_{ij}\}_{j\in\mathcal{N}_i} \leftarrow$

$$\underset{u_i,\{z_{ij}\}_{j\in\mathcal{N}_i}}{\arg\min} \ \mathcal{L}_\Theta^i\big(u_i, \{z_{ij}, w_{ij}^L, \lambda_{ij}\}_{j\in\mathcal{N}_i}\big)$$

$$\text{s.t.} \ \big(u_i, \{z_{ij}\}_{j\in\mathcal{N}_i}\big) \in \mathcal{C}_i$$

4: Send data to server: $\{z_{ij}\}_{j\in\mathcal{N}_i}$

5: $d^c \leftarrow L := \text{length}(\{\{w_{ij}\}_{j\in\mathcal{N}_i}\})$

$c \leftarrow c + 1$

$u_i^c, \{z_{ij}^c\} \leftarrow u_i, \{z_{ij}\}_{j\in\mathcal{N}_i}$

Until receive the terminating signal from the server.

Output:

6:

$$\bar{u}_i, \{\bar{z}_{ij}\}_{j\in\mathcal{N}_i} := \frac{\sum_{c\geq C_i} d^c\{u_i^c, \{z_{ij}^c\}\}}{\sum_{c\geq C_i} d^c} \tag{3.10}$$

Every agent works responsively and asynchronously. When it receives data from the server, the agent computes the update and replies to the server according to Algorithm 3.3 or 3.4. In particular:

- In Step 1, the agent passively receives from the server a time sequence of recorded consensus updates from its connected counterparts, which was sent in Step 3 in Algorithm 3.2.
- The agent then reconstructs the λ_{ij} by adding the primal residuals as shown in Step 2. Note that for $\forall i \in U$ and $\forall j \in V$ there is a slight difference in such additions.
- In Step 3, the agent updates in the same way as Step 1 or 2 in Algorithm 3.1.
- The agent responds to the server with the updated consensus data in Step 4.

Algorithm 3.4 Decentralised Asynchronous ADMM - (3/3) $\forall j \in V$ in parallel

Repeat:

1: Receive data from server: $\{\{z_{ij}^l\}_{i\in\mathcal{N}_j}\}_{l=1}^L$

2: $\lambda_{ij} \leftarrow \lambda_{ij} + \sum_{l=1}^L \Theta_{ij}(z_{ij}^l - w_{ij}^{c-1})$ $\qquad \forall i \in \mathcal{N}_j$

3: $v_j, \{w_{ij}\}_{i\in\mathcal{N}_j} \leftarrow$

$$\underset{v_j,\{w_{ij}\}_{i\in\mathcal{N}_j}}{\arg\min}\ \mathcal{L}_\Theta^j\big(v_j, \{w_{ij}, z_{ij}^L, \lambda_{ij}\}_{i\in\mathcal{N}_j}\big)$$

$$\text{s.t. } \big(v_j, \{w_{ij}\}_{i\in\mathcal{N}_j}\big) \in \mathcal{C}_j$$

4: Send data to server: $\{w_{ij}\}_{i\in\mathcal{N}_j}$

5: $d^c \leftarrow L := \text{length}(\{\{z_{ij}\}_{i\in\mathcal{N}_j}\})$

$c \leftarrow c+1$

$v_j^c, \{w_{ij}^c\} \leftarrow v_j, \{w_{ij}\}_{i\in\mathcal{N}_j}$

Until receive the terminating signal from the server.

Output:

6:

$$\bar{v}_j, \{\bar{w}_{ij}\}_{i\in\mathcal{N}_j} := \frac{\sum_{c\geq C_j} d^c\{v_j^c, \{w_{ij}^c\}\}}{\sum_{c\geq C_j} d^c} \tag{3.11}$$

- In Step 5, the agent records the weights of its historical iterates, making preparations for the running-average output in Step 6.
- When the algorithm is terminated, the agent outputs the weighted running average as in Step 6. The window sizes C_i and C_j of the running average output are adjustable and could be either set to k_0 or set independently (these choices being equivalent in the limit as $K \to \infty$).

Algorithm 3.5 provides a summary of Algorithms 3.2, 3.3 and 3.4, after simplification by removing the detailed description of the information that passes through the data exchange server. To understand this:

- Steps 1 and 2 of Algorithm 3.5 resemble Steps 1 and 2 of the synchronous Algorithm 3.1, but with historical data.
- The local reconstructions of λ_{ij} (in Step 2 of Algorithms 3.3 and 3.4) are equivalent to Step 3 of Algorithm 3.5.

Algorithm 3.5 Decentralised Asynchronous ADMM - Complete Picture

Initialise: z_{ij}, w_{ij}, λ_{ij}, Θ_{ij} $\forall (i,j) \in E$

Repeat:

$\forall i \in U$, $\forall j \in V$ at server's clock cycle $k \geq 1$:

1: U Update $\forall i \in U$

$$u_i^k, \{z_{ij}^k\}_{j \in \mathcal{N}_i} \leftarrow$$

$$\underset{u_i, \{z_{ij}\}_{j \in \mathcal{N}_i}}{\arg\min} \; \mathcal{L}_\Theta^i \left(u_i, \{z_{ij}, w_{ij}^{a_i^2(k)}, \lambda_{ij}^{a_i^2(k)}\}_{j \in \mathcal{N}_i} \right) \tag{3.12}$$

$$\text{s.t.} \quad \left(u_i, \{z_{ij}\}_{j \in \mathcal{N}_i}\right) \in \mathcal{C}_i$$

2: V Update $\forall j \in V$

$$v_j^k, \{w_{ij}^k\}_{i \in \mathcal{N}_j} \leftarrow$$

$$\underset{v_j, \{w_{ij}\}_{i \in \mathcal{N}_j}}{\arg\min} \; \mathcal{L}_\Theta^j \left(v_j, \{w_{ij}, z_{ij}^{b_j^2(k)}, \lambda_{ij}^{b_j^2(k)-1}\}_{i \in \mathcal{N}_j} \right) \tag{3.13}$$

$$\text{s.t.} \; \left(v_j, \{w_{ij}\}_{i \in \mathcal{N}_j}\right) \in \mathcal{C}_j$$

The local reconstruction of λ_{ij} is equivalent to:

3: E Update $\forall (i,j) \in E$, at server's clock cycle k

$$\lambda_{ij}^k \leftarrow \lambda_{ij}^{k-1} + \Theta_{ij}(z_{ij}^k - w_{ij}^k) \tag{3.14}$$

4: $k \leftarrow k+1$

Until $k = K$

Output:

5:

$$\bar{u}_i^K, \{\bar{z}_{ij}^K\}_{j \in \mathcal{N}_i} \quad \forall i \in U$$
$$\bar{v}_j^K, \{\bar{w}_{ij}^K\}_{i \in \mathcal{N}_j} \quad \forall j \in V$$

in which the running average $\bar{x}^K := \frac{1}{K} \sum_{k=1}^K x^k$

- The weighted running averages in Step 6 of Algorithms 3.4 and 3.5 are equivalent to the arithmetic average in Step 5 of Algorithm 3.5 since duplicates are recorded in clock cycles in which no data is received by the data exchange server.

Assumption 3.4. We assume the following conditions:

(a). Bounded delay and postponed conflicts $\forall k$:

$$1 \leq a_i^1(k) - a_i^2(k) \leq \tau_i, \quad \forall i \in U, \tag{3.15}$$

$$1 \leq b_j^1(k) - b_j^2(k) \leq \tau_j, \quad \forall j \in V. \tag{3.16}$$

(b). For all $(i,j) \in E$ with $\tau_i \neq 1$, F_{ij} is strongly convex with a generalised modulus $\Sigma_{ij}^U \succ 0$ defined as:

$$\begin{aligned} &\partial F_{ij}(z_{ij}^\dagger)^\intercal (z_{ij} - z_{ij}^\dagger) + \frac{1}{2}\|z_{ij} - z_{ij}^\dagger\|_{\Sigma_{ij}^U}^2 \\ \leq &F_{ij}(z_{ij}) - F_{ij}(z_{ij}^\dagger), \quad \forall z_{ij}^\dagger, z_{ij} \in \mathbb{R}^{m_{ij}}. \end{aligned} \tag{3.17}$$

(c). For all $(i,j) \in E$, G_{ij} is strongly convex with a generalised modulus $\Sigma_{ij}^V \succ 0$ defined as:

$$\begin{aligned} &\partial G_{ij}(w_{ij}^\dagger)^\intercal (w_{ij} - w_{ij}^\dagger) + \frac{1}{2}\|w_{ij} - w_{ij}^\dagger\|_{\Sigma_{ij}^V}^2 \\ \leq\ &G_{ij}(w_{ij}) - G_{ij}(w_{ij}^\dagger), \quad \forall w_{ij}^\dagger, w_{ij} \in \mathbb{R}^{m_{ij}}. \end{aligned} \tag{3.18}$$

$\forall (i,j) \in E$, we define τ_{ij} and α_{ij} as

$$\tau_{ij} := 2\tau_i + 2\tau_j - 4, \tag{3.19}$$

$$\alpha_{ij} := 1 + \frac{1}{2}(3\tau_{ij} + \sqrt{5\tau_{ij}^2 + 8\tau_{ij} + 4}). \tag{3.20}$$

(d). $\forall (i,j) \in E$ such that $\tau_i \neq 1$:

$$\frac{\Sigma_{ij}^U}{\alpha_{ij}(4\tau_i - 4)} - \Theta_{ij} \succeq 0. \tag{3.21}$$

(e). $\forall (i,j) \in E$:

$$\frac{\Sigma_{ij}^V}{\alpha_{ij}(4\tau_j - 3)} - \Theta_{ij} \succeq 0. \tag{3.22}$$

The convergence of the proposed asynchronous ADMM can be stated as follows (a proof is provided in Section 3.6).

Theorem 3.2. *Let Assumptions 3.1, 3.3 and 3.4 hold. Then Algorithms 3.2, 3.3 and 3.4 (or equivalently Algorithm 3.5 in the limit as $K \to \infty$) have the following asymptotic properties:*

1. *The reconstructed local running averages $\bar{u}_i$, $\{\bar{z}_{ij}\}_{j\in\mathcal{N}_i}$, $\forall i \in U$ in Algorithm 3.3 and $\bar{v}_j$, $\{\bar{w}_{ij}\}_{i\in\mathcal{N}_j}$, $\forall j \in V$ in Algorithm 3.4 converge as $K \to \infty$ to a saddle point $\{u_i^*\}, \{v_j^*\}, \{z_{ij}^*, w_{ij}^*\}_{(i,j)\in E}$ of the Lagrangian (3.7).*

2. *Equivalently, $\{\bar{u}_i^K\}$, $\{\bar{v}_j^K\}$, $\{\bar{z}_{ij}^K, \bar{w}_{ij}^K\}_{(i,j)\in E}$ in Algorithm 3.5 converge to $\{u_i^*\}$, $\{v_j^*\}$, $\{z_{ij}^*, w_{ij}^*\}_{(i,j)\in E}$ as $K \to \infty$.*

3.4 Numerical Analysis and Comparison

This section investigates the convergence properties of the proposed algorithm through numerical simulations. The example considered is the following modified Ridge regression problem (linear regression with ℓ_2 regularisation):

$$\begin{aligned}\min_{\{z_i\}_{i\in U}} \quad & \sum_{i\in U} r_i \|z_i\|_2^2 \\ & + \sum_{j\in V} \Big(\sum_{i\in U} \big(\|A_{ij} z_i - b_{ij}\|_2^2 + \sum_{k\in U} c_j \|z_i - z_k\|_2^2 \big) \Big)\end{aligned} \tag{3.23a}$$

subject to

$$\text{s.t. } \underline{z}\mathbf{1}_n \le z_i \le \bar{z}\mathbf{1}_n, \ \forall i \in U. \tag{3.23b}$$

We assume $z_i \in \mathbb{R}^n$, $r_i > 0$, $\forall i \in U$; $c_j > 0$, $\forall j \in V$; $A_{ij} \in \mathbb{R}^{m\times n}$, $b_{ij} \in \mathbb{R}^m$, $\forall (i,j) \in U \times V$. This problem can be viewed as $|U|$ independent learning nodes that identify their respective parameters $\{z_i\}$ via the local data $\{A_{ij},\ b_{ij}\}$ stored in the $|V|$ data centres, with $\{r_i\}$ being the penalty terms for ℓ_2 regularisation. Data centres may have prior knowledge that some parameters are related, and this motivates the inclusion of the penalty terms $\{c_j\}$. The vectors containing the elements of the matrices A_{ij}

are each drawn from the normal distribution $N(0, I_{mn})$; each b_{ij} is generated using $b_{ij} = A_{ij}\hat{z}_i + d_{ij}$, where the noise vector d_{ij} is drawn from $N(0, 0.01I_m)$, and $\hat{z}_i = z^{\text{ref}} + e_i$ where each element of z^{ref} is zero with probability 0.5 and otherwise is drawn from $N(0, 1)$, and the noise e_i is drawn from $N(0, 0.01I_n)$. The remaining coefficients are $|U| = 4$, $|V| = 4$, $n = 10$, $\underline{z} = -2$, $\bar{z} = 2$, $\{c_j = 10\}_{\forall j \in V}$. We reformulate Problem (3.23) equivalently as

$$\min_{\{z_i\}_{i\in U}, \{w_{ij}\}_{(i,j)\in U\times V}} \sum_{i\in U} r_i \|z_i\|_2^2 \tag{3.24a}$$

$$+ \sum_{j\in V} \Big(\sum_{i\in U} (\|A_{ij}w_{ij} - b_{ij}\|_2^2 + \sum_{k\in U} c_j \|w_{ij} - w_{kj}\|_2^2) \Big) \tag{3.24b}$$

subject to:

$$\underline{z}\mathbf{1}_n \leq z_i \leq \bar{z}\mathbf{1}_n, \ \forall i \in U, \tag{3.24c}$$

$$z_i = w_{ij}, \ \forall (i,j) \in U \times V. \tag{3.24d}$$

To see the equivalence of this with problem $\mathcal{P}$, note that each $i \in U$ has the decision variable $z_i = z_{ij}, \forall j \in V$ with the local cost function (3.24a) and the local constraint set (3.24c), whereas each $j \in V$ has local decision variables $\{w_{ij}\}_{\forall i \in U}$ with the local objective (3.24b). The realisation of the delay t_i, $i \in U$, is modelled as: $t_i \sim \lfloor N_{tr}(\frac{\tau_i+1}{2}, (\frac{\tau_i-1}{4})^2, 1, \tau_i) \rceil$, and t_j, $j \in V$, is modelled analogously. The delay upper bounds are identical for all $i \in U$ or $j \in V$, namely $\{\tau_i = \tau_U\}_{\forall i \in U}$ and $\{\tau_j = \tau_V\}_{\forall j \in V}$.

Fig. 3.4 displays the convergence behaviour of the proposed asynchronous ADMM algorithm. We define the residual of the objective value $\mathrm{R}^{\text{obj}}(k) := \frac{|\text{obj}^k - \text{obj}^*|}{|\text{obj}^*|}$ where obj^* is the optimal objective value obtained with a centralised solver. The parameter vector $\mathrm{p}_s := [\theta, \tau_U, \tau_V]$ summarises the simulation parameters, where θ is a scalar defining $\{\Theta_{ij} = \theta I_n\}_{\forall (i,j) \in E}$. We also compute $\theta_r := [\hat{\theta}, \bar{\theta}]$, in which $\hat{\theta}$ is the step size computed using (3.21) and (3.22) in Assumption 3.4, evaluated using the upper bounds on delays, and $\bar{\theta}$ is the corresponding step size evaluated at the expected values of the delays.

When the local objective functions of all the agents are strongly convex, we observe from Fig. 3.4(a) that the iterations converge until θ has increased to a critical value,

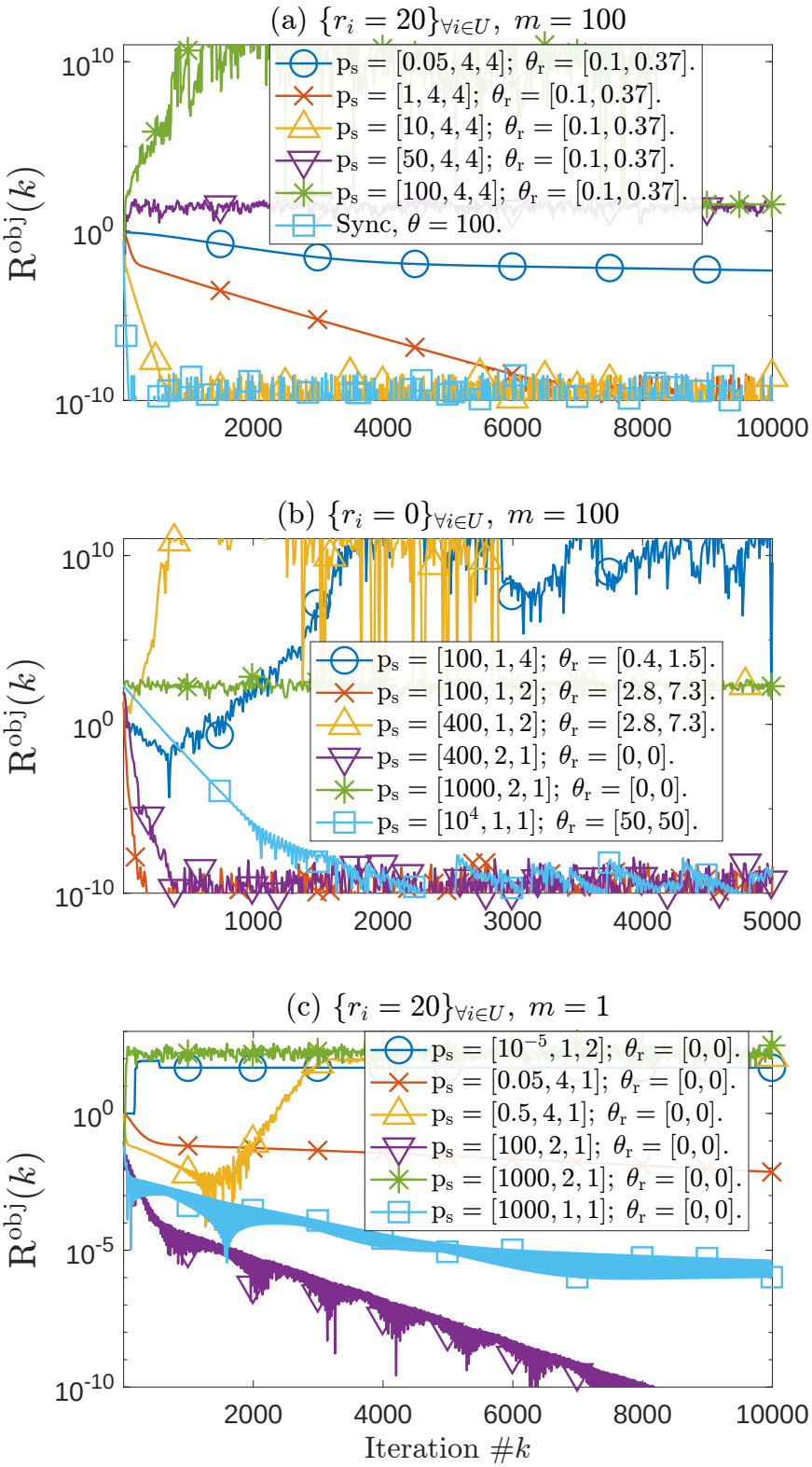


Figure 3.4: Convergence with the local objective functions of the agents (a) all being strongly convex, (b) only being strongly convex in group V, (c) only being strongly convex in group U.

above which the iterations diverge. In this specific example, the threshold from the simulation is a factor of 10^3 higher than $\hat{\theta}$ and 10^2 higher than $\bar{\theta}$. If the local objectives of the agents in U are not strongly convex, the simulation result in Fig. 3.4(b) shows that (i) the critical value of θ increases as the maximum delays decrease; (ii) when we interchange the value of τ_U and τ_V, the case that $\tau_U > \tau_V$ results in larger critical value of θ, even when θ_r diminishes to zero due to the loss of strong convexity; (iii) when $\tau_U = \tau_V = 1$, the iteration converges even when θ is increased to 10^4. Fig. 3.4(c) presents the results when local objectives of V are not strongly convex. We observe: (i) when τ_V is greater than 1, the iterations diverge no matter how small θ is; (ii) Lower τ_U implies higher critical value for θ; (iii) when $\tau_U = \tau_V = 1$, similar to (b)(iii), the iteration converges at high θ. To summarise the numerical analysis: the processing agents in V are more intolerant both to non-strongly-convex local objectives and larger delays (that require lower θ values for convergence). These observations are consistent with Theorem 3.2 but they also indicate that the sufficient conditions provided by (3.21) and (3.22) in Assumption 3.4 for the critical value of θ are conservative.

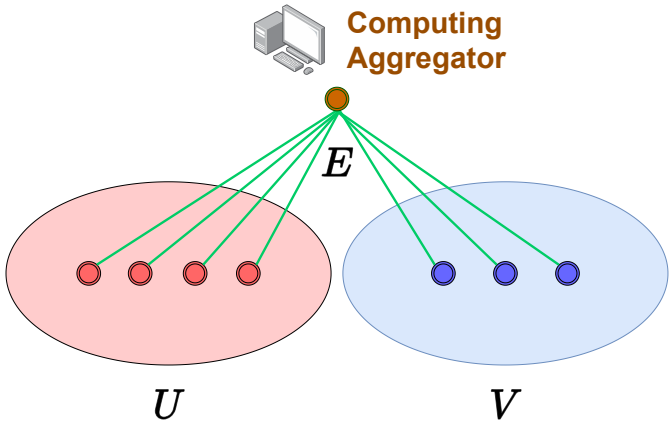


Figure 3.5: Replacing the data exchange server with an aggregator.

Problem (3.23) was further explored using the distributed ADMM approach of [15, Algorithm 4]. In this approach, a computing aggregator (see Fig. 3.5) replaces the data exchange server previously shown in Fig. 3.2. This aggregator is configured to

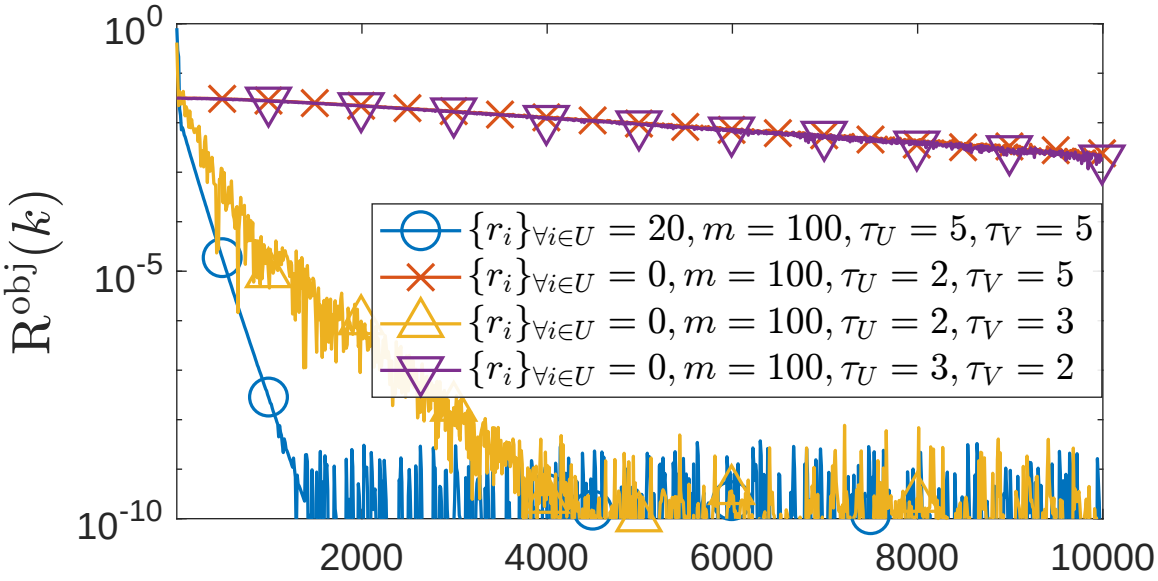


Figure 3.6: Convergence when $\theta \to \theta^{\mathrm{lim}}$ with [15, Algorithm 4].

maintain local copies of all the consensus decision variables $\{z_i\}_{\forall i\in U}, \{w_{ij}\}_{\forall (i,j)\in U\times V}$, executing updates in concordance with the form in [1, Sec. 7.2]. Owing to the substantial alterations made to the ADMM structure, the convergence rates of the two algorithms are not directly comparable. However Fig. 3.6 provides an indication of the performance of [15, Algorithm 4] for several different delay bounds when its penalty parameter θ is set to the empirical upper limit θ^{lim} that has the highest convergence rate.

The plots in Fig. 3.6 are analogous to those in Fig. 3.4(a) (for $\{r_i\}_{\forall i\in U} = 20$) and Fig. 3.4 (for $\{r_i\}_{\forall i\in U} = 0$). In this comparison, an extra clock cycle must be included in the delay for [15, Algorithm 4] due to the change from a data exchange server to a computing aggregator, as depicted in Fig. 3.7. From qualitative comparison of Fig. 3.6 with Fig. 3.4, we conclude that the proposed asynchronous ADMM converges rapidly when applied to the problems for which [15, Algorithm 4] converges within 5000 iterations, and moreover the proposed algorithm also converges in one of the two cases shown in Fig. 3.6 in which the convergence of [15, Algorithm 4] is impractically slow. Since it does not require a communication system with OSI (Open Systems Interconnection [166]) Layer 6 (Presentation) and Layer 7 (Application), the proposed method using a computation-free server considerably reduces processing time,

resulting in smaller delays and/or allowing faster clock cycles. This characteristic potentially facilitates the server's integration into existing communication infrastructure.

Moreover, the absence of an encryption/decryption process in the proposed data exchange server inherently safeguards peer-to-peer privacy, thereby strengthening its appeal as a viable alternative to a computing aggregator. The data exchange server necessitates larger memory allocation to cache historical data. Quantitatively, this is approximately p times the amount employed by the aggregator, where p is proportional to the average of $\{\frac{1}{2}(\frac{\tau_i}{\tau_j} + \frac{\tau_j}{\tau_i})\}_{(i,j)\in E}$. However, it is crucial to recognise that when computational aspects are taken into account, the aggregator's memory requirements may significantly exceed those of the data exchange server. Regarding the communication cost, the exchange server maintains a data transfer rate equivalent to that of the synchronous case (albeit with fluctuations due to asynchrony), and when compared with a computing aggregator, it offers bandwidth savings by eliminating the need to transfer data for dual variable updates.

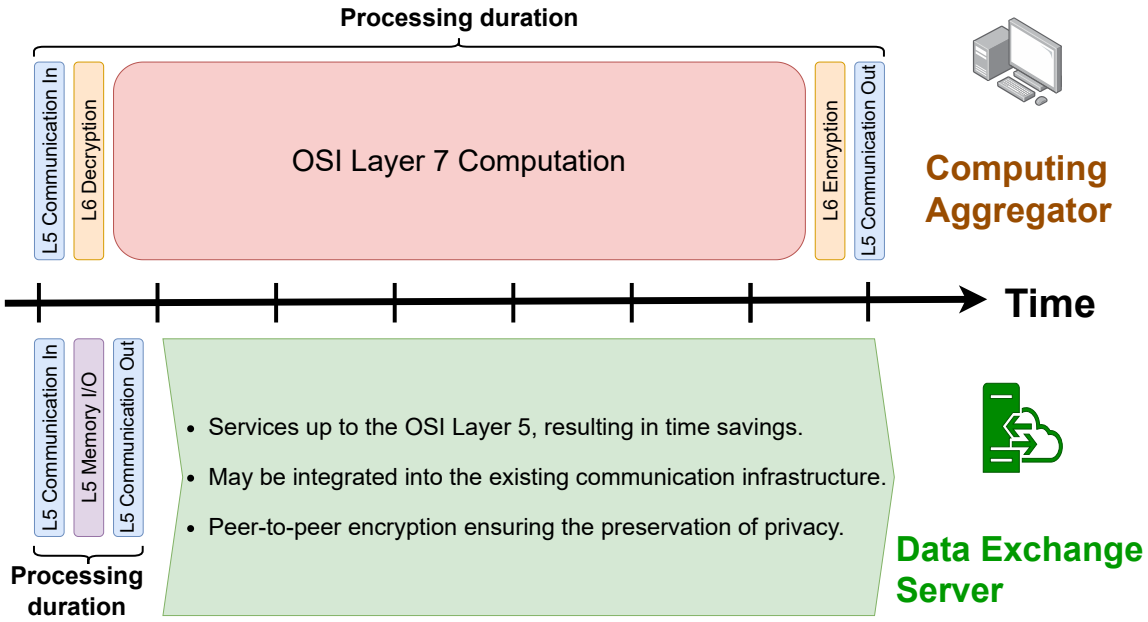


Figure 3.7: Key differences between a computing aggregator and a data exchange server.

One may argue if this data-exchange server adds vulnerability to the entire system.

However, this logical server can be physically realised by multiple synchronised, peer-to-peer oriented servers with redundancies.

3.5 Conclusion

This chapter proposes an asynchronous, distributed ADMM optimisation algorithm for problems with local consensus coupling constraints, in which a computation-free data exchange server handles the communication between agents with delays. Under assumptions of strongly convex local objectives and upper limits on communication delays, sufficient conditions are derived on the penalty parameters in the augmented Lagrangian formulation in order to ensure that the solver iterations converge asymptotically. In numerical experiments we observe that the sufficient conditions are conservative and in practice the algorithm may tolerate delays even when local objectives are not non-strongly convex.

Future work will involve: (i) enabling inter-agent communication within group U or V via virtual agents (with affine local constraint sets and zero local objectives), as described for example in [1, Sec. 7], with the help of the data exchange server; (ii) investigating acceleration methods for improving the linear convergence rates that are observed in simulations; (iii) tightening the sufficient conditions for algorithm convergence.

3.6 Convergence Analysis

Lemma 3.1. *Similar to [109, Lemma 4.1], consider $h_1(x)$ and $h_2(x)$ are convex functions over the convex domain $x \in \mathcal{X}$. We define $\Sigma \succeq 0$ such that $\partial h_1(x_0)^\intercal(x - x_0) + \frac{1}{2}\|x - x_0\|_\Sigma^2 \leq h_1(x) - h_1(x_0), \forall x, x_0 \in \mathcal{X}$. If $\Sigma \succ 0$, this implies $h_1(x)$ is strongly convex. We also define $\hat{x} := \arg\min_{x\in\mathcal{X}} h_1(x) + h_2(x)$.*

Therefore we have $\forall x \in \mathcal{X}$:

$$h_1(\hat{x}) - h_1(x) + \frac{1}{2}\|\hat{x} - x\|_\Sigma^2 + \partial h_2(\hat{x})^\intercal(\hat{x} - x) \leq 0. \tag{3.25}$$

Proof. Since $\hat{x} := \arg\min_{x\in\mathcal{X}} h_1(x) + h_2(x)$, $h_1, h_2, \mathcal{X}$ being convex, we have:

$$h_1(\hat{x}) - h_1(x) - \partial h_1(\hat{x})^\intercal(\hat{x} - x) + \frac{1}{2}\|\hat{x} - x\|_\Sigma^2 \leq 0,$$
$$(\partial h_1(\hat{x}) + \partial h_2(\hat{x}))^\intercal(\hat{x} - x) \leq 0.$$

By combining the two equations we obtain (3.25). □

Proof of Theorem 3.2. Since the results 1) and 2) stated in the theorem are equivalent, we explicitly prove only 2).

Part 0: we note that a saddle point of our Lagrangian (3.7): $\{u_i^*\}, \{v_j^*\}, \{z_{ij}^*, w_{ij}^*, \lambda_{ij}^*\}_{(i,j)\in E}$ has the following properties:

$$z_{ij}^* = w_{ij}^* \quad \forall (i,j) \in E \tag{3.26}$$

$$\begin{aligned}&\mathcal{L}_\Theta(\{u_i^*\}, \{v_j^*\}, \{z_{ij}^*, w_{ij}^*, \lambda_{ij}^*\}_{(i,j)\in E}) \\ &\quad \leq \mathcal{L}_\Theta(\{u_i\}, \{v_j\}, \{z_{ij}, w_{ij}, \lambda_{ij}^*\}_{(i,j)\in E}) \\ &\qquad \forall \{u_i\}, \{v_j\}, \{z_{ij}\}, \{w_{ij}\} \text{ s.t. } \bigcap \forall(3.4d) \forall(3.4e)\end{aligned} \tag{3.27}$$

Part 1: $\forall i \in U$, $\forall k \geq 1$, from (3.12) and Lemma 3.1 we have:

$$\begin{aligned}&f_i(u_i^k) - f_i(u_i^*) + \sum_{j\in\mathcal{N}_i} \Big[F_{ij}(z_{ij}^k) - F_{ij}(z_{ij}^*) \\ &\quad + \Big(\lambda_{ij}^{a_i^2(k)} + \Theta_{ij}(z_{ij}^k - w_{ij}^{a_i^2(k)})\Big)^\intercal (z_{ij}^k - z_{ij}^*) \\ &\quad + \frac{1}{2}\|z_{ij}^k - z_{ij}^*\|_{\Sigma_{ij}^U}^2\Big] \leq 0,\end{aligned} \tag{3.28}$$

in which with the arbitrarily chosen $\{\lambda_{ij}\}$:

$$\begin{aligned}&\lambda_{ij}^{a_i^2(k)} + \Theta_{ij}(z_{ij}^k - w_{ij}^{a_i^2(k)}) \\ &\quad = \lambda_{ij}^{a_i^2(k)} + \Theta_{ij}(z_{ij}^k - w_{ij}^k) + \Theta_{ij}(w_{ij}^k - w_{ij}^{a_i^2(k)}) \\ &\quad \overset{(3.14)}{=} \lambda_{ij}^{a_i^2(k)} + \lambda_{ij}^k - \lambda_{ij}^{k-1} + \Theta_{ij}(w_{ij}^k - w_{ij}^{a_i^2(k)}) \\ &\quad = \lambda_{ij} + (\lambda_{ij}^{a_i^2(k)} - \lambda_{ij}^{k-1}) + (\lambda_{ij}^k - \lambda_{ij}) \\ &\qquad + \Theta_{ij}(w_{ij}^k - w_{ij}^{a_i^2(k)}),\end{aligned} \tag{3.29}$$

and

$$
\begin{aligned}
&\left(\Theta_{ij}(w_{ij}^k - w_{ij}^{a_i^2(k)})\right)^\intercal (z_{ij}^k - z_{ij}^*) \\
&\quad = (w_{ij}^k - w_{ij}^{a_i^2(k)} + w_{ij}^{k-1} - w_{ij}^{k-1})^\intercal \\
&\qquad \Theta_{ij}(z_{ij}^k - z_{ij}^* + w_{ij}^k - w_{ij}^k) \\
&\quad = (w_{ij}^k - w_{ij}^{k-1})^\intercal \Theta_{ij}(w_{ij}^k - z_{ij}^*) \\
&\qquad + (w_{ij}^k - w_{ij}^{k-1})^\intercal \Theta_{ij}(z_{ij}^k - w_{ij}^k) \\
&\qquad + (w_{ij}^{k-1} - w_{ij}^{a_i^2(k)})^\intercal \Theta_{ij}(z_{ij}^k - z_{ij}^*) \\
&\overset{(3.26)(3.14)}{=} (w_{ij}^k - w_{ij}^{k-1})^\intercal \Theta_{ij}(w_{ij}^k - w_{ij}^*) \\
&\qquad + (w_{ij}^k - w_{ij}^{k-1})^\intercal (\lambda_{ij}^k - \lambda_{ij}^{k-1}) \\
&\qquad + (w_{ij}^{k-1} - w_{ij}^{a_i^2(k)})^\intercal \Theta_{ij}(z_{ij}^k - z_{ij}^*).
\end{aligned} \tag{3.30}
$$

Part 2: similar to Part 1, $\forall j \in V$, $\forall k$, from (3.13) and Lemma 3.1 we have:

$$
\begin{aligned}
&g_j(v_j^k) - g_j(v_j^*) + \sum_{i \in \mathcal{N}_j} \Bigg[G_{ij}(w_{ij}^k) - G_{ij}(w_{ij}^*) \\
&\quad - \left(\lambda_{ij}^{b_j^2(k)-1} + \Theta_{ij}(z_{ij}^{b_j^2(k)} - w_{ij}^k)\right)^\intercal (w_{ij}^k - w_{ij}^*) \\
&\quad + \frac{1}{2}\|w_{ij}^k - w_{ij}^*\|_{\Sigma_{ij}^V}^2 \Bigg] \leq 0,
\end{aligned} \tag{3.31}
$$

in which:

$$
\begin{aligned}
&\lambda_{ij}^{b_j^2(k)-1} + \Theta_{ij}(z_{ij}^{b_j^2(k)} - w_{ij}^k) \\
&\quad = \lambda_{ij}^{b_j^2(k)-1} + \Theta_{ij}(z_{ij}^{b_j^2(k)} - w_{ij}^{b_j^2(k)}) + \Theta_{ij}(w_{ij}^{b_j^2(k)} - w_{ij}^k) \\
&\overset{(3.14)}{=} \lambda_{ij}^{b_j^2(k)} + \Theta_{ij}(w_{ij}^{b_j^2(k)} - w_{ij}^k) \\
&\quad = (\lambda_{ij}^{b_j^2(k)} - \lambda_{ij}^k) + (\lambda_{ij}^k - \lambda_{ij}) + \lambda_{ij} \\
&\qquad + \Theta_{ij}(w_{ij}^{b_j^2(k)} - w_{ij}^{k-1}) + \Theta_{ij}(w_{ij}^{k-1} - w_{ij}^k).
\end{aligned} \tag{3.32}
$$

Part 3: we combine the equations above as well as (3.26), (3.14), sum $\forall i \in U$ $\forall j \in V$, and take the average over K steps:

$$
\begin{aligned}
&\frac{1}{K}\sum_{k=1}^{K} \Bigg[\Delta p^k + \sum_{(i,j)\in E} \Bigg(\lambda_{ij}^\intercal (z_{ij}^k - w_{ij}^k) \\
&\qquad + (\lambda_{ij}^k - \lambda_{ij})^\intercal \Theta_{ij}^{-1} (\lambda_{ij}^k - \lambda_{ij}^{k-1})
\end{aligned} \tag{3.33a}
$$

$$
\begin{aligned}
&+ 2(w_{ij}^k - w_{ij}^{k-1})^\intercal \Theta_{ij}(w_{ij}^k - w_{ij}^*) && \text{(3.33b)}\\
&+ \frac{1}{2}\|z_{ij}^k - z_{ij}^*\|_{\Sigma_{ij}^U}^2 + \frac{1}{2}\|w_{ij}^k - w_{ij}^*\|_{\Sigma_{ij}^V}^2 \\
&+ (\lambda_{ij}^{a_i^2(k)} - \lambda_{ij}^{k-1})^\intercal (z_{ij}^k - z_{ij}^*) && \text{(3.33c)}\\
&+ (w_{ij}^{k-1} - w_{ij}^{a_i^2(k)})^\intercal \Theta_{ij}(z_{ij}^k - z_{ij}^*) && \text{(3.33d)}\\
&+ (\lambda_{ij}^k - \lambda_{ij}^{b_j^2(k)})^\intercal (w_{ij}^k - w_{ij}^*) && \text{(3.33e)}\\
&+ (w_{ij}^{k-1} - w_{ij}^{b_j^2(k)})^\intercal \Theta_{ij}(w_{ij}^k - w_{ij}^*) && \text{(3.33f)}\\
&+ (\lambda_{ij}^k - \lambda_{ij}^{k-1})^\intercal (w_{ij}^k - w_{ij}^{k-1}) && \text{(3.33g)}\\
&\quad \Bigg)\Bigg] \leq 0,
\end{aligned}
$$

in which:

$$
\begin{aligned}
\Delta p^k := &\sum_{i\in U}\Big(f_i(u_i^k) - f_i(u_i^*)\Big) \\
&+ \sum_{j\in V}\Big(g_j(v_j^k) - g_j(v_j^*)\Big) \\
&+ \sum_{(i,j)\in E}\Big(F_{ij}(z_{ij}^k) - F_{ij}(z_{ij}^*) \\
&+ G_{ij}(w_{ij}^k) - G_{ij}(w_{ij}^*)\Big).
\end{aligned} \tag{3.34}
$$

We note that $\|\lambda_{ij}^a - \lambda_{ij}^b\|_{\Theta_{ij}^{-1}}^2 = \|\lambda_{ij}^a\|_{\Theta_{ij}^{-1}}^2 + \|\lambda_{ij}^b\|_{\Theta_{ij}^{-1}}^2 - 2(\lambda_{ij}^a)^\intercal \Theta_{ij}^{-1}\lambda_{ij}^b$, therefore:

$$
\begin{aligned}
\sum_{k=1}^K \text{(3.33a)} &= \frac{1}{2}\sum_{k=1}^K \Big(\|\lambda_{ij}^k - \lambda_{ij}\|_{\Theta_{ij}^{-1}}^2 + \|\lambda_{ij}^k - \lambda_{ij}^{k-1}\|_{\Theta_{ij}^{-1}}^2 \\
&\quad - \|\lambda_{ij}^{k-1} - \lambda_{ij}\|_{\Theta_{ij}^{-1}}^2\Big) \\
&= \frac{1}{2}\Big(\|\lambda_{ij}^K - \lambda_{ij}\|_{\Theta_{ij}^{-1}}^2 - \|\lambda_{ij}^1 - \lambda_{ij}\|_{\Theta_{ij}^{-1}}^2 \\
&\quad + \sum_{k=1}^K \|\lambda_{ij}^k - \lambda_{ij}^{k-1}\|_{\Theta_{ij}^{-1}}^2\Big).
\end{aligned} \tag{3.35}
$$

Similarly,

$$
\begin{aligned}
\sum_{k=1}^K \text{(3.33b)} &= \|w_{ij}^K - w_{ij}^*\|_{\Theta_{ij}}^2 - \|w_{ij}^1 - w_{ij}^*\|_{\Theta_{ij}}^2 \\
&\quad + \sum_{k=1}^K \|w_{ij}^k - w_{ij}^{k-1}\|_{\Theta_{ij}}^2.
\end{aligned} \tag{3.36}
$$

We bound the following term:

$$\begin{aligned}\sum_{k=1}^{K}(3.33c) &= \sum_{k=1}^{K}\sum_{l=a_i^2(k)}^{k-2}(\lambda_{ij}^{l}-\lambda_{ij}^{l+1})^{\intercal}(z_{ij}^{k}-z_{ij}^{*}) \\ &\leq \sum_{k=1}^{K}\sum_{l=a_i^2(k)}^{k-2}\big(\frac{1}{2\alpha_{ij}}\|\lambda_{ij}^{l}-\lambda_{ij}^{l+1}\|^2_{\Theta_{ij}^{-1}}+\frac{\alpha_{ij}}{2}\|z_{ij}^{k}-z_{ij}^{*}\|^2_{\Theta_{ij}}\big) \\ &\leq (\tau_i-1)\sum_{k=1}^{K}\big(\frac{1}{\alpha_{ij}}\|\lambda_{ij}^{k}-\lambda_{ij}^{k-1}\|^2_{\Theta_{ij}^{-1}}+\alpha_{ij}\|z_{ij}^{k}-z_{ij}^{*}\|^2_{\Theta_{ij}}\big),\end{aligned} \tag{3.37}$$

where $\alpha_{ij}>0$. To see the 2nd inequality in (3.37), we count the maximum possible number of duplicates $(\lambda_{ij}^{k}-\lambda_{ij}^{k+1})\ \forall k$ illustrated in Fig. 3.8, which is $2\tau_i-2$.

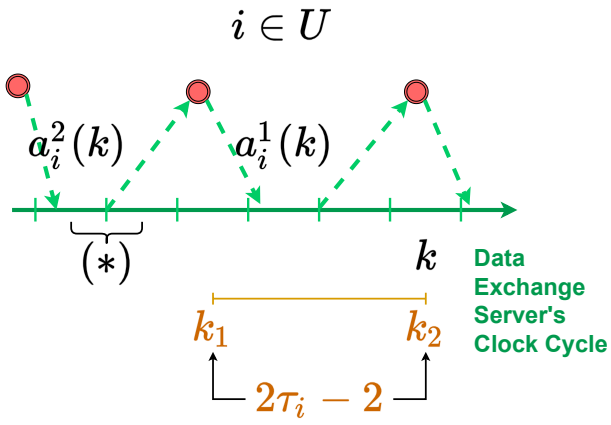


Figure 3.8: The $l \to l+1$ couple $(*)$ is duplicated up to $2\tau_i-2$ times as $k_1 \leq k \leq k_2$ when deriving the 2nd inequality in (3.37).

Similarly:

$$\begin{aligned}\sum_{k=1}^{K}(3.33d) &\leq (\tau_i-1)\sum_{k=1}^{K}\big(\frac{1}{\alpha_{ij}}\|w_{ij}^{k}-w_{ij}^{k-1}\|^2_{\Theta_{ij}} \\ &\quad+\alpha_{ij}\|z_{ij}^{k}-z_{ij}^{*}\|^2_{\Theta_{ij}}\big),\end{aligned} \tag{3.38}$$

$$\begin{aligned}\sum_{k=1}^{K}(3.33e) &\leq \frac{2\tau_j-1}{2}\sum_{k=1}^{K}\big(\frac{1}{\alpha_{ij}}\|\lambda_{ij}^{k}-\lambda_{ij}^{k-1}\|^2_{\Theta_{ij}^{-1}} \\ &\quad+\alpha_{ij}\|w_{ij}^{k}-w_{ij}^{*}\|^2_{\Theta_{ij}}\big),\end{aligned} \tag{3.39}$$

$$\sum_{k=1}^{K}(3.33f) \leq (\tau_j-1)\sum_{k=1}^{K}\big(\frac{1}{\alpha_{ij}}\|w_{ij}^{k}-w_{ij}^{k-1}\|^2_{\Theta_{ij}}$$

$$+ \alpha_{ij}\|w_{ij}^k - w_{ij}^*\|_{\Theta_{ij}}^2), \tag{3.40}$$

$$\sum_{k=1}^{K}(3.33g) \le \frac{1}{2}\sum_{k=1}^{K}(\frac{\beta_{ij}}{\alpha_{ij}}\|\lambda_{ij}^k - \lambda_{ij}^{k-1}\|_{\Theta_{ij}^{-1}}^2 + \frac{\alpha_{ij}}{\beta_{ij}}\|w_{ij}^k - w_{ij}^{k-1}\|_{\Theta_{ij}}^2). \tag{3.41}$$

where $\beta_{ij} > 0$.

Part 4: we rearrange Part 3 after having substituted (3.35), (3.36), (3.37), (3.38), (3.39), (3.40) and (3.41) into (3.33):

$$\begin{aligned}
&\frac{1}{K}\sum_{k=1}^{K}\Bigg[\Delta p^k + \sum_{(i,j)\in E}\Bigg(\lambda_{ij}^\intercal(z_{ij}^k - w_{ij}^k) \\
&+ \frac{2\tau_i + 2\tau_j - 3 + \beta_{ij} - \alpha_{ij}}{2\alpha_{ij}}\|\lambda_{ij}^k - \lambda_{ij}^{k-1}\|_{\Theta_{ij}^{-1}}^2 && \text{(3.42a)} \\
&+ \frac{2\tau_i + 2\tau_j - 4 + \alpha_{ij}^2/\beta_{ij} - 2\alpha_{ij}}{2\alpha_{ij}}\|w_{ij}^k - w_{ij}^{k-1}\|_{\Theta_{ij}}^2 && \text{(3.42b)} \\
&+ \frac{1}{2}(z_{ij}^k - z_{ij}^*)^\intercal\left[(4\tau_i - 4)\alpha_{ij}\Theta_{ij} - \Sigma_{ij}^U\right](z_{ij}^k - z_{ij}^*) && \text{(3.42c)} \\
&+ \frac{1}{2}(w_{ij}^k - w_{ij}^*)^\intercal\left[(4\tau_j - 3)\alpha_{ij}\Theta_{ij} - \Sigma_{ij}^V\right](w_{ij}^k - w_{ij}^*) && \text{(3.42d)} \\
&\Bigg)\Bigg] + \frac{1}{2K}\sum_{(i,j)\in E}\Bigg(\|\lambda_{ij}^K - \lambda_{ij}\|_{\Theta_{ij}^{-1}}^2 - \|\lambda_{ij}^1 - \lambda_{ij}\|_{\Theta_{ij}^{-1}}^2 \\
&+ \|w_{ij}^K - w_{ij}^*\|_{\Theta_{ij}}^2 - \|w_{ij}^1 - w_{ij}^*\|_{\Theta_{ij}}^2\Bigg) \le 0.
\end{aligned}$$

We choose $\beta_{ij} = \alpha_{ij} - (2\tau_i + 2\tau_j - 3) \ge 0$ to make (3.42a) $= 0$. Solve (3.42b) ≤ 0 and $\alpha_{ij} - (2\tau_i + 2\tau_j - 3) \ge 0$ for α_{ij}:

$$\alpha_{ij} \ge 1 + \frac{1}{2}(3\tau_{ij} + \sqrt{5\tau_{ij}^2 + 8\tau_{ij} + 4}), \tag{3.43}$$

where $\tau_{ij} := 2\tau_i + 2\tau_j - 4$.

Let $\alpha_{ij} = 1 + \frac{1}{2}(3\tau_{ij} + \sqrt{5\tau_{ij}^2 + 8\tau_{ij} + 4})$. Solve (3.42c) ≤ 0, (3.42d) ≤ 0 for Θ_{ij}, and we obtain the conditions (a)-(e) in Assumption 3.4.

Part 5: with Assumption 3.4 being imposed, condition (3.42) is therefore reduced to:

$$\begin{aligned}
&\frac{1}{K}\sum_{k=1}^{K}\left[\Delta p^k + \sum_{(i,j)\in E}\lambda_{ij}^{\intercal}(z_{ij}^k - w_{ij}^k)\right] \\
&\quad - \frac{1}{2K}\sum_{(i,j)\in E}\left(\|\lambda_{ij}^1 - \lambda_{ij}\|^2_{\Theta_{ij}^{-1}} + \|w_{ij}^1 - w_{ij}^*\|^2_{\Theta_{ij}}\right) \\
&= \frac{1}{K}\sum_{k=1}^{K}\Delta p^k + \sum_{(i,j)\in E}\left(\lambda_{ij}^{\intercal}(\bar{z}_{ij}^K - \bar{w}_{ij}^K)\right. \\
&\quad \left. - \frac{1}{2K}\left(\|\lambda_{ij}^1 - \lambda_{ij}\|^2_{\Theta_{ij}^{-1}} + \|w_{ij}^1 - w_{ij}^*\|^2_{\Theta_{ij}}\right)\right) \\
&\leq 0,
\end{aligned} \tag{3.44}$$

in which $\bar{x}^K$ denotes the running average: $\sum_k x^k$.

Since we have convex cost functions,

$$\begin{aligned}
\frac{1}{K}\sum_{k=1}^{K}\Delta p^k \geq & \sum_{i\in U}\left(f_i(\bar{u}_i^K) - f_i(u_i^*)\right) \\
& + \sum_{j\in V}\left(g_j(\bar{v}_j^K) - g_j(v_j^*)\right) \\
& + \sum_{(i,j)\in E}\left(F_{ij}(\bar{z}_{ij}^K) - F_{ij}(z_{ij}^*)\right. \\
& \left. + G_{ij}(\bar{w}_{ij}^K) - G_{ij}(w_{ij}^*)\right) := \Delta\bar{p}^K.
\end{aligned} \tag{3.45}$$

From (3.27), we have:

$$\Delta\bar{p}^K + \sum_{(i,j)\in E}\lambda_{ij}^{*\intercal}(\bar{z}_{ij}^K - \bar{w}_{ij}^K) \geq 0. \tag{3.46}$$

We combine (3.44) and (3.45). From the result we subtract (3.46) to get:

$$\begin{aligned}
&\sum_{(i,j)\in E}\left((\lambda_{ij} - \lambda_{ij}^*)^{\intercal}(\bar{z}_{ij}^K - \bar{w}_{ij}^K)\right. \\
&\quad \left. - \frac{1}{2K}\left(\|\lambda_{ij}^1 - \lambda_{ij}\|^2_{\Theta_{ij}^{-1}} + \|w_{ij}^1 - w_{ij}^*\|^2_{\Theta_{ij}}\right)\right) \leq 0.
\end{aligned} \tag{3.47}$$

Since $\{\lambda_{ij}\}$ are arbitrarily chosen, we choose $\lambda_{ij} = \lambda_{ij}^* + e_{ij}$, $e_{ij} = \frac{\bar{z}_{ij}^K - \bar{w}_{ij}^K}{\|\bar{z}_{ij}^K - \bar{w}_{ij}^K\|_2}$ and

substitute into (3.47) to obtain:

$$\begin{aligned}
&\sum_{(i,j)\in E} \|\bar{z}_{ij}^K - \bar{w}_{ij}^K\|_2 \\
&\quad \le \frac{1}{K} \sum_{(i,j)\in E} \left(\frac{1}{2} \max_{\|e_{ij}\|_2=1} \|\lambda_{ij}^1 - \lambda_{ij}^* - e_{ij}\|_{\Theta_{ij}^{-1}}^2 \right. \\
&\qquad \left. + \|w_{ij}^1 - w_{ij}^*\|_{\Theta_{ij}}^2 \right) \\
&\quad := \frac{C_1}{K}.
\end{aligned} \tag{3.48}$$

We also have:

$$\begin{aligned}
&\Delta\bar{p}^K + \sum_{(i,j)\in E} \lambda_{ij}^{*\intercal}(\bar{z}_{ij}^K - \bar{w}_{ij}^K) \\
&\overset{(3.46)}{=} |\Delta\bar{p}^K + \sum_{(i,j)\in E} \lambda_{ij}^{*\intercal}(\bar{z}_{ij}^K - \bar{w}_{ij}^K)| \\
&\ge |\Delta\bar{p}^K| - \sum_{(i,j)\in E} |\lambda_{ij}^{*\intercal}(\bar{z}_{ij}^K - \bar{w}_{ij}^K)| \\
&\ge |\Delta\bar{p}^K| - \sum_{(i,j)\in E} \|\lambda_{ij}^*\|_\infty \|\bar{z}_{ij}^K - \bar{w}_{ij}^K\|_1 \\
&\overset{(*)}{\ge} |\Delta\bar{p}^K| - \sum_{(i,j)\in E} \|\lambda_{ij}^*\|_\infty \sqrt{\dim(\lambda_{ij})} \|\bar{z}_{ij}^K - \bar{w}_{ij}^K\|_2 \\
&\overset{(3.48)}{\ge} |\Delta\bar{p}^K| - \frac{C_1 C_2}{K},
\end{aligned} \tag{3.49}$$

in which $(*)$ is due to norm equivalence.

Finally from (3.49),

$$\begin{aligned}
|\Delta\bar{p}^K| &\le \frac{C_1 C_2}{K} + \Delta\bar{p}^K + \sum_{(i,j)\in E} \lambda_{ij}^{*\intercal}(\bar{z}_{ij}^K - \bar{w}_{ij}^K) \\
&\overset{(3.44)(3.45)}{\le} \frac{1}{2K}\big(C_3 + \|\lambda_{ij}^1 - \lambda_{ij}^*\|_{\Theta_{ij}^{-1}}^2 + \|w_{ij}^1 - w_{ij}^*\|_{\Theta_{ij}}^2\big) \\
&= \frac{C}{K}.
\end{aligned} \tag{3.50}$$

We output the following results: $\{\bar{u}_i^K\}, \{\bar{v}_j^K\}, \{\bar{z}_{ij}^K, \bar{w}_{ij}^K\}$.

Feasibility check: Note that (3.48) shows as $K \to \infty$ the output satisfies dualised constraints (3.4c). Since taking the running average is a convex combination, it also satisfies all the the local constraints (3.4d)(3.4e).

Optimality: (3.50) shows as $K \to \infty$ the output minimises the cost function of $\mathcal{P}$, and hence converges to a minimiser of our problem. □

Chapter 4

Optimisation with Parametric Uncertainty: an ADMM Approach

4.1 Introduction

Iterative algorithms are widely used to solve convex optimisation problems. Many problems of interest contain parameters that are uncertain and require estimation. For example, consider an optimal control problem in which some parameters of the controlled system are not known exactly but can be estimated from noisy measurements. In this situation it is common to use system identification to estimate unknown plant parameters [e.g. 167], and to wait for the estimated parameters to converge to a sufficiently high level of accuracy before passing the identified model to the decision-making process to compute an optimal control law. In this chapter we are concerned with the question of whether we can combine the estimation and optimisation processes in order to accelerate the solution of such problems.

We consider the following optimisation problem:

$$\min_{x,z} \quad f(x) + g(z), \tag{4.1a}$$

$$\text{subject to} \quad x - z = 0, \tag{4.1b}$$

in which $f, g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are convex, closed, proper functions. Let $f(x) = \hat{f}(x) + \mathcal{I}_{\mathcal{F}_f}(x)$, $g(z) = \hat{g}(z) + \mathcal{I}_{\mathcal{F}_g}(z)$, where $\hat{f}, \hat{g} : \mathbb{R}^n \to \mathbb{R}$ are convex and continuous, $\mathcal{F}_f, \mathcal{F}_g \subset \mathbb{R}^n$ are convex constraint sets, and $\mathcal{I}_{\mathcal{C}}(x)$ is the indicator function of a set $\mathcal{C}$. We assume that some parameters of this problem are unknown, but may be estimated from available measurements. Thus we replace f in (4.1a) at discrete time instant k by a time-varying function f^k:

$$\min_{x,z} \quad f^k(x) + g(z), \tag{4.2a}$$

$$\text{subject to} \quad x - z = 0, \tag{4.2b}$$

where

$$f^k(x) = \bar{f}(x) + \mathcal{I}_{\mathcal{F}_f^k}(x) = \frac{1}{2} x^\intercal Q x + c^\intercal x + \mathcal{I}_{\mathcal{F}_f^k}(x), \tag{4.2c}$$

$$\mathcal{F}_f^k : \quad Ax \leq b + Dp^k. \tag{4.2d}$$

Here Q and c are respectively a given matrix and vector, and $p^k \in \mathcal{P} \subseteq \mathbb{R}^m$ is a random variable representing an estimate of an uncertain parameter in problem (4.1). In practice, an MPC problem with uncertain polytopic constraints can be modelled in this form (See in Sec. 2.4.3 and [150]). We assume that the parameter set $\mathcal{P}$ is compact and that p^k can be expressed in terms of a constant but unknown parameter $\bar{p}$ and a random variable δ^k as $p^k = \bar{p} + \delta^k$. We consider the convergence properties of an approach based on the Alternating Direction Method of Multipliers [ADMM 1], with $\bar{p}$ approximated by p^k.

The recent work [28] solves a semidefinite programming problem using an approximate ADMM solver. However, rather than considering problems with uncertain parameters, in [28] approximations of the exact ADMM iteration are used in order to reduce the computational requirements of individual iterations, and this introduces

errors into the solution estimates. The authors show that convergence is guaranteed provided that the errors introduced in the ADMM iteration are summable across the iterations.

In [168], a Model Predictive Control (MPC) optimisation problem is considered using an ADMM solver with a finite number of iterations. The authors construct an invariant set containing the target state for the combination of the control problem and the ADMM solver, such that the dynamics of the solver iterations become linear within this set. It is shown that this simplifies the analysis of the stability of the combined system.

A weighted average consensus algorithm is considered in [169] as the basis of a distributed solution for problems involving networked systems with locally time-varying quadratic cost functions. This is motivated by the problem of tracking the optimal solutions of resource allocation problems with time-varying local demand or resource constraints. The approach is able to track the optimal solution in a distributed manner with bounded tracking error, with bounds that are dependent on the cost parameters and network topology.

In each of Rontsis, Goulart, and Nakatsukasa [28], Schulze Darup and Book [168], and Esteki, Kia, and Member [169], the inherent robustness of solver iterations is exploited to derive guarantees of convergence despite bounded uncertainty or random errors in problem data. In this chapter we use similar robustness properties to devise a solver for (4.1). At each iteration, our approach uses the most recent estimate p^k to replace the unknown parameter $\bar{p}$ by replacing the uncertain function f with its corresponding estimate f^k. Our approach is applicable more generally to first order methods that can be expressed as Douglas-Rachford splitting (DRS) methods. We show that, if the solver iteration can be expressed in the form of a DRS operator, then, under the assumption of convergent estimation errors, the solution estimate necessarily converges to the solution of (4.1) corresponding to $p^k = \bar{p}$.

4.1.1 Notation

$\mathbb{R}^n$ denotes the n-dimensional real space. I represents the identity mapping. $Q \succ 0$ and $R \succeq 0$ represent real symmetric positive definite and positive semidefinite matrices. $\mathcal{I}_{\mathcal{C}}(x)$ denotes the indicator function of a closed non-empty set $\mathcal{C}$, so that $\mathcal{I}_{\mathcal{C}}(x) = 0$ for $x \in \mathcal{C}$ and $\mathcal{I}_{\mathcal{C}}(x) = \infty$ otherwise. $\partial F(x)$ indicates the subdifferential of function F evaluated at x. The n-dimensional column vector with all elements equal to 1 is $\mathbf{1}_n$. The truncated normal distribution [1] is denoted $N_{tr}(\mu, \sigma^2, a, b)$. The Euclidean distance between y and $\mathcal{X}$ is denoted $\mathbf{dist}(y, \mathcal{X}) := \inf_{x \in \mathcal{X}} \|x - y\|_2$ and $B_r(q) = \{x \mid \|x - q\|_2 < r\}$ is the open ball of radius r centred on q. $N_{\mathcal{C}}(y) = \partial \mathcal{I}_{\mathcal{C}}(y)$ denotes the normal cone of the feasible set $\mathcal{C}$. $\mathbb{E}[\cdot]$ denotes the expected value of a random variable. $x^k \xrightarrow{\mathrm{P}} x_0$ denotes convergence in probability, i.e., $\forall \epsilon > 0$, $\lim_{k \to \infty} \mathbb{P}\left[\|x^k - x_0\| > \epsilon\right] = 0$.

4.2 Problem Formulation

We first consider the solution of the deterministic problem defined by

$$\min_{x,z} \quad \bar{f}(x) + g(z), \tag{4.3a}$$

$$\text{subject to} \quad x - z = 0, \tag{4.3b}$$

where $\bar{f}(x) := \frac{1}{2} x^\intercal Q x + c^\intercal x + \mathcal{I}_{\bar{\mathcal{F}}_f}(x)$ with $\bar{\mathcal{F}}_f : Ax \leq b + D\bar{p}$. This is simply problem (4.1) with the true parameter value $\bar{p}$. We construct the augmented Lagrangian for (4.3):

$$\bar{\mathcal{L}}_\gamma := \bar{f}(x) + g(z) + \lambda^\intercal (x - z) + \frac{1}{2\gamma} \|x - z\|_2^2, \tag{4.4}$$

in which $\gamma > 0$ is a penalty parameter. By defining the scaled multiplier $u := \gamma \lambda$, problem (4.3) can be solved [e.g. 1] via the following ADMM iteration:

$$x^{k+1} \leftarrow \min_x \bar{\mathcal{L}}_\gamma(x, z^k, \lambda^k) := \mathbf{prox}_{\gamma \bar{f}}(z^k - u^k), \tag{4.5a}$$

$$z^{k+1} \leftarrow \min_z \bar{\mathcal{L}}_\gamma(x^{k+1}, z, \lambda^k) := \mathbf{prox}_{\gamma g}(x^{k+1} + u^k), \tag{4.5b}$$

[1] If a random variable x has the normal distribution $N(\mu, \sigma^2)$ and $a < b$, then the distribution of x conditional on $a \leq x \leq b$ is denoted $N_{tr}(\mu, \sigma^2, a, b)$. We specifically define $x \sim N_{tr}(\mu, \sigma^2, a, a)$ as $\mathbb{P}(x = a) = 1$.

$$u^{k+1} \leftarrow u^k + x^{k+1} - z^{k+1}, \tag{4.5c}$$

where $\mathbf{prox}_{\gamma h}(v) := \arg\min_x \left(h(x) + \frac{1}{2\gamma}\|x - v\|_2^2 \right)$ denotes the *proximal operator* and k is an iteration counter. ADMM is well-suited for distributed optimisation when $\bar{f}(x)$ and/or $g(z)$ are separable functions.

By introducing the *reflected proximal operator* $R_{\gamma h} := 2\mathbf{prox}_{\gamma h} - I$, the ADMM iteration (4.5) can be expressed using *Douglas-Rachford Splitting* (DRS) in terms of an operator $\bar{T} : \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$v^{k+1} = \bar{T}v^k := \left(\frac{1}{2}I + \frac{1}{2}R_{\gamma\bar{f}}R_{\gamma g} \right) v^k, \tag{4.6}$$

where $v^k := x^k + u^{k-1}$, and x, z, u in (4.5) can be obtained given the value of v (see [28] and [170], Appendix B) as:

$$x^k = \mathbf{prox}_{\gamma\bar{f}} R_{\gamma g} v^{k-1}, \tag{4.7a}$$

$$z^k = \mathbf{prox}_{\gamma g} v^k, \tag{4.7b}$$

$$u^k = \left(I - \mathbf{prox}_{\gamma g} \right) v^k. \tag{4.7c}$$

Note that, as a result of DRS, $\bar{T}$ is a *firmly non-expansive* operator [see e.g. 171] that satisfies, $\forall v, w \in \mathbb{R}^n$,

$$\|\bar{T}v - \bar{T}w\|_2^2 + \|(I - \bar{T})v - (I - \bar{T})w\|_2^2 \leq \|v - w\|_2^2. \tag{4.8}$$

4.3 ADMM with Parametric Uncertainty

We suppose p^k in (4.2) is a real-time estimate of $\bar{p}$ in (4.3), with

$$p^k = \bar{p} + \delta^k. \tag{4.9}$$

Convergent estimators typically ensure that $\|\delta^k\|_2 \to 0$ almost surely (a.s.) as $k \to \infty$, and we therefore expect to obtain an accurate estimate of $\bar{p}$ by running the estimator for a sufficiently long time. By subsequently passing this estimate to the ADMM solver and iterating for another sufficiently large number of iterations we expect to obtain an approximate solution to problem (4.3). In this section we consider how

to combine the two processes to reliably accelerate the overall solution process for real-world applications with estimated problem data.

By integrating p^k into the ADMM iteration (4.5) we propose the following ADMM algorithm with parametric uncertainty (ADMM-PU):

$$x^{k+1} \leftarrow \mathbf{prox}_{\gamma f^k}(z^k - u^k), \tag{4.10a}$$

$$z^{k+1} \leftarrow \mathbf{prox}_{\gamma g}(x^{k+1} + u^k), \tag{4.10b}$$

$$u^{k+1} \leftarrow u^k + x^{k+1} - z^{k+1}, \tag{4.10c}$$

in which $f^k(x)$ is defined in terms of p^k via (4.2c)-(4.2d). We assume for convenience that the solver and estimator share a common iteration index k; more generally p^k denotes the most recently available estimate of $\bar{p}$.

Similar to (4.6)-(4.7), the ADMM-PU iteration (4.10) can be expressed as a time-varying DRS operator $T^k : \mathbb{R}^n \to \mathbb{R}^n$:

$$v^{k+1} = T^k v^k := \left(\frac{1}{2}I + \frac{1}{2}R_{\gamma f^k} R_{\gamma g}\right) v^k, \tag{4.11}$$

where $v^k := x^k + u^{k-1}$, and x, z, u can be obtained from:

$$x^k = \mathbf{prox}_{\gamma f^k} R_{\gamma g} v^{k-1}, \tag{4.12a}$$

$$z^k = \mathbf{prox}_{\gamma g} v^k, \tag{4.12b}$$

$$u^k = \left(I - \mathbf{prox}_{\gamma g}\right) v^k. \tag{4.12c}$$

The following lemma shows that T^k can be expressed in terms of a bounded perturbation of the operator $\bar{T}$ defined in (4.6).

Lemma 4.1. *The iteration (4.10) can be represented as*

$$v^{k+1} = \bar{T} v^k + E^k \delta^k,$$

where $E^k \in \mathbb{R}^{n \times m}$ satisfies $\|E^k\|_2 \leq \bar{e}$ for some finite $\bar{e}$.

Proof. Since (4.10) can be expressed as (4.11), we have

$$\begin{aligned} v^{k+1} = T^k v^k &= \frac{1}{2}v^k + \frac{1}{2}R_{\gamma f^k} R_{\gamma g} v^k \\ &= \frac{1}{2}v^k + \frac{1}{2}(2\mathbf{prox}_{\gamma f^k} - I) R_{\gamma g} v^k. \end{aligned} \tag{4.13}$$

For all $y \in \mathbb{R}^n$, $\mathbf{prox}_{\gamma f^k}(y)$ solves the following problem

$$\begin{aligned} \min_x \quad & \frac{1}{2}x^\intercal Q x + c^\intercal x + \frac{1}{2\gamma}\|x - y\|_2^2, && \text{(4.14a)} \\ \text{subject to} \quad & Ax \leq b + Dp(x^k) = b + D\bar{p} + D\delta,\ \delta = \delta^k. && \text{(4.14b)} \end{aligned}$$

Define a matrix H, vector w and linear functional s as follows,

$$H := Q + \gamma^{-1} I \quad (Q \succeq 0 \text{ and } \gamma > 0 \Rightarrow\ H \succ 0), \tag{4.15a}$$

$$w := x + H^{-1}(c - \gamma^{-1} y), \tag{4.15b}$$

$$s(y) := AH^{-1}(c - \gamma^{-1} y) + b + D\bar{p}. \tag{4.15c}$$

Then, by completing the square, solving (4.14) is equivalent to solving

$$\begin{aligned} w^*(\delta) := \arg\min_w \quad & \frac{1}{2} w^\intercal H w, && \text{(4.16a)} \\ \text{subject to} \quad & Aw \leq s(y) + D\delta, && \text{(4.16b)} \end{aligned}$$

since $\mathbf{prox}_{\gamma f^k}(y) = w^*(\delta^k) - H^{-1}(c - \gamma^{-1} y)$. Problem (4.16) takes the standard form of a right-hand side *multi-parametric quadratic program* (mp-QP), and the optimiser $w^*(\delta) : \mathbb{R}^m \to \mathbb{R}^n$ is therefore continuous and piecewise affine, and can be expressed in the following form [see 29]:

$$w^*(\delta) = H^{-1}\tilde{A}^\intercal (\tilde{A} H^{-1} \tilde{A}^\intercal)^{-1} (\tilde{s}(y) + \tilde{D}\delta). \tag{4.17}$$

Here $\tilde{A}$, $\tilde{s}(y)$ and $\tilde{D}$ correspond to the set of active constraints that depends on δ, and we assume the rows of $\tilde{A}$ are linearly independent. Since $w^*(\delta)$ is piecewise affine, $w^*(\delta^k)$ can be expressed as:

$$w^*(\delta^k) = w^*(0) + E^k \delta^k, \tag{4.18}$$

where

$$w^*(0) = \mathbf{prox}_{\gamma \bar{f}}(y) + H^{-1}(c - \gamma^{-1} y), \tag{4.19}$$

and bounds on E^k can be determined by considering the value of $\partial w^*(\delta)$ along the line segment L from $\delta = 0$ to $\delta = \delta^k$. Without loss of generality we impose $E^k = 0$

when $\delta^k = 0$. For $\delta^k \neq 0$, the mean value theorem [52] implies that $E^k = \partial w^*(\bar{\delta})$, for some $\bar{\delta} \in L$. Since $\partial w^*(\delta) \ni H^{-1}\tilde{A}^\intercal(\tilde{A}H^{-1}\tilde{A}^\intercal)^{-1}\tilde{D}$ is independent of y we have

$$\|E^k\|_2 \leq \max_{j \in J} \|H^{-1}\tilde{A}(j)^\intercal(\tilde{A}(j)H^{-1}\tilde{A}(j)^\intercal)^{-1}\tilde{D}(j)\|_2 := \bar{e}$$

where index j indicates a given set of active constraints, and J is the set of indices of all possible active constraint sets for problem (4.16). Note that $\bar{e}$ is necessarily finite since H in (4.15) is positive definite.

Combining (4.15b) and (4.18)-(4.19) we obtain

$$\begin{aligned} \mathbf{prox}_{\gamma f^k}(y) &= w^*(0) - H^{-1}(c - \gamma^{-1}y) + E^k\delta^k \\ &= \mathbf{prox}_{\gamma \bar{f}}(y) + E^k\delta^k. \end{aligned} \tag{4.20}$$

Substituting (4.20) into (4.13) then gives the result

$$T^k v^k = \frac{1}{2}v^k + \frac{1}{2}R_{\gamma\bar{f}}R_{\gamma g}v^k + E^k\delta^k = \bar{T}v^k + E^k\delta^k, \tag{4.21}$$

where $\|E^k\|_2 \leq \bar{e}$. □

Our assumptions that the feasible set of problem (4.2) is non-empty and compact, and that the error in the estimate of $\bar{p}$ in problem (4.3) is bounded and convergent, can be stated as follows.

Assumption 4.1. For all k we require that:

(a) $\mathcal{F}_f^k$ is non-empty and satisfies, for finite r,

$$\mathcal{F}_f^k = \{x \mid Ax \leq b + Dp^k\} \subset B_r(0),$$

(b) $\delta^k \in B_\rho(0)$ and $\sum_{i=0}^{k} \|\delta^i\|_2 \leq \Delta$ a.s., for finite ρ and Δ,

(c) $\forall y \in \bar{\mathcal{F}}_f \cap \mathcal{F}_g$, $\forall \mu \in -N_{\bar{\mathcal{F}}_f}(y)$, $\forall \omega \in N_{\mathcal{F}_g}(y)$, with $c^{max} < 1$,

$$\mu^\intercal\omega \leq c^{max}\|\mu\|_2\|\omega\|_2.$$

(d) The strong duality holds for the time-varying $\mathcal{L}_\gamma^k$ and the reference $\bar{\mathcal{L}}_\gamma$ Lagrangians associated with the ADMM.

Let $\mathbf{Fix}(T)$ denote the fixed point set of $\bar{T}$ that corresponds to the set of saddle points of (4.4). Before analysing convergence of ADMM-PU, we first show that the iteration (4.11) ensures $\|\bar{T}v^k - v^*\|_2$ is finite for all k, where v^* is any point in $\mathbf{Fix}(T)$ such that $\|v^0 - v^*\|_2$ is finite.

Lemma 4.2. *Under Assumption 4.1 the iteration (4.11) satisfies, for all k,*

$$\|\bar{T}v^k - v^*\|_2 \leq \|v^0 - v^*\|_2 + \bar{e}\Delta, \tag{4.22}$$

where $v^* \in \mathbf{Fix}(T)$.

Proof. From Lemma 4.1 we have

$$\begin{aligned}\|v^{k+1} - v^*\|_2 &= \|\bar{T}v^k + E^k\delta^k - v^*\|_2 \\ &\leq \|\bar{T}v^k - v^*\|_2 + \|E^k\delta^k\|_2 \\ &\leq \|\bar{T}v^k - v^*\|_2 + \bar{e}\|\delta^k\|_2.\end{aligned}$$

But (4.8) implies that $\bar{T}$ is nonexpansive, since $v = v^k$ and $w = v^*$ in (4.8) gives $\|\bar{T}v^k - v^*\|_2 \leq \|v^k - v^*\|_2$, and hence

$$\|v^{k+1} - v^*\|_2 \leq \|v^k - v^*\|_2 + \bar{e}\|\delta^k\|_2.$$

Summing both sides of this inequality over $0 \leq k \leq N$, we obtain

$$\|v^{N+1} - v^*\|_2 \leq \|v^0 - v^*\|_2 + \bar{e}\sum_{k=0}^{N}\|\delta^k\|_2.$$

From Assumption 4.1(b), it follows that, for all k,

$$\|v^k - v^*\|_2 \leq \|v^0 - v^*\|_2 + \bar{e}\Delta,$$

and (4.22) then follows from the nonexpansive of $\bar{T}$, which implies $\|\bar{T}v^k - v^*\|_2 \leq \|v^k - v^*\|_2$. □

We next show that the optimal dual variable for problem (4.3) is necessarily finite under Assumption 4.1(a) and (c). This result is also needed in Section 4.4, and it is therefore stated here for the general case of problem (4.2) with δ^k replaced by a given

realisation δ. In this setting $(x^*_\delta, z^*_\delta, \lambda^*_\delta)$ represents a saddle point of the associated Lagrangian function and $\mathcal{F}_f$ represents the constraint set $\{x : Ax \leq b+Dp\}$ associated with a given realisation $p \in \mathcal{P}$.

Lemma 4.3. *If $\mathcal{F}_f(\delta) \cap \mathcal{F}_g$ is compact, and if, $\forall y \in \mathcal{F}_f(\delta) \cap \mathcal{F}_g$, $\forall \mu \in -N_{\mathcal{F}_f}(y)$, $\forall \omega \in N_{\mathcal{F}g}(y)$, with $c^{max} < 1$ we have*

$$\mu^\intercal \omega \leq c^{max}(\delta)\|\mu\|_2\|\omega\|_2,$$

then the optimal dual variable λ^*_δ lies in a bounded set.

Proof. We prove Lemma 4.3 by contradiction and suppose that λ^*_δ may be infinite. From the Karush-Kuhn-Tucker (KKT) conditions, the Lagrange multiplier λ^*_δ satisfies $0 \in \partial f(x^*_\delta) + \lambda^*_\delta$ and $0 \in \partial g(x^*_\delta) - \lambda^*_\delta$. Therefore

$$\lambda^*_\delta \in \big(\partial \hat{g}(x^*_\delta) + N_{\mathcal{F}g}(x^*_\delta)\big) \cap -\big(\partial \hat{f}(x^*_\delta) + N_{\mathcal{F}_f}(x^*_\delta)\big), \tag{4.23}$$

so there must exist $\mu \in -N_{\mathcal{F}_f}(x^*_\delta)$ and $\omega \in N_{\mathcal{F}g}(x^*_\delta)$ such that $\mu = \lambda^*_\delta + d_{\hat{f}}$ and $\omega = \lambda^*_\delta - d_{\hat{g}}$ for some $d_{\hat{f}} \in \partial \hat{f}(x^*_\delta)$ and $d_{\hat{g}} \in \partial \hat{g}(x^*_\delta)$. But $\forall y \in \mathcal{F}_f \cap \mathcal{F}_g$, $\forall d_{\hat{f}} \in \partial \hat{f}(y)$, $\forall d_{\hat{g}} \in \partial \hat{g}(y)$,

$$\begin{aligned} &\frac{(\lambda^*_\delta + d_{\hat{f}})^\intercal(\lambda^*_\delta - d_{\hat{g}})}{\|\lambda^*_\delta + d_{\hat{f}}\|_2\|\lambda^*_\delta - d_{\hat{f}}\|_2} \\ &\geq \frac{\|\lambda^*_\delta\|_2^2 - \|\lambda^*_\delta\|_2(\|d_{\hat{f}}\|_2 + \|d_{\hat{g}}\|_2) - \|d_{\hat{f}}\|_2\|d_{\hat{g}}\|_2}{\|\lambda^*_\delta\|_2^2 + \|\lambda^*_\delta\|_2(\|d_{\hat{f}}\|_2 + \|d_{\hat{g}}\|_2) + \|d_{\hat{f}}\|_2\|d_{\hat{g}}\|_2}. \end{aligned} \tag{4.24}$$

Here $\|d_{\hat{f}}\|_2 \leq d_m$ and $\|d_{\hat{g}}\|_2 \leq d_m$ for some finite d_m since $\mathcal{F}_f(\delta) \cap \mathcal{F}_g$ is by assumption compact, and hence

$$\frac{(\lambda^*_\delta + d_{\hat{f}})^\intercal(\lambda^*_\delta - d_{\hat{g}})}{\|\lambda^*_\delta + d_{\hat{f}}\|_2\|\lambda^*_\delta - d_{\hat{f}}\|_2} \geq \frac{\|\lambda^*_\delta\|_2^2 - 2d_m\|\lambda^*_\delta\|_2 - d_m^2}{\|\lambda^*_\delta\|_2^2 + 2d_m\|\lambda^*_\delta\|_2 + d_m^2} \tag{4.25}$$

Therefore $\mu^\intercal\omega/(\|\mu\|_2\|\omega\|_2) \to 1$ as $\|\lambda^*_\delta\|_2 \to \infty$, contradicting the assertion that $c^{max} < 1$. □

Theorem 4.1. *Under Assumption 4.1, the ADMM-PU iteration (4.10) converges as $k \to \infty$, with:*

(i) $\|\text{obj}^k - \text{obj}^*\|_2 \to 0$ *a.s.*

(ii) $\|\lambda^k - \lambda^*\|_2 \to 0$ *a.s.*

(iii) $\|x^k - z^k\|_2 \to 0$ *a.s.*

where obj^* *and* λ^* *are respectively the optimal values of the objective and dual variable for problem* (4.3)*, and* $\text{obj}^k := \text{obj}(x^k, z^k) := \frac{1}{2}x^{k\intercal}Qx^k + c^\intercal x^k + g(z^k)$.

Remark 4.1. The ADMM iteration (4.5), in which $\bar{p}$ is known, provides deterministic guarantees of convergence of obj^k, λ^k, x^k and z^k analogous to (i)-(iii) [e.g. 1]).

Remark 4.2. Assumption 4.1(b) requires that $\|\delta^k\|_2$ converges sufficiently rapidly to zero as $k \to \infty$. Since $\|\delta^k\|_2 \leq \|\delta^k\|_1$, this convergence assumption is satisfied by any l_1-stable estimator (such as an exponentially stable Luenberger observer, for example) driven by a finite l_1-norm noise sequence [e.g. 172, Sec. 6.7].

Proof. The proof of Theorem 4.1 is given in two parts.

Part I (to prove that $\|(I - \bar{T})v^k\|_2^2 \to 0$ a.s.): Let v^* be any point in $\mathbf{Fix}(T)$ such that $\|v^0 - v^*\|_2$ is finite. Then the firmly non-expansive property of $\bar{T}$ implies (e.g. by setting $v = v^k$ and $w = v^*$ in (4.8)) that

$$\|\bar{T}v^k - v^*\|_2^2 + \|(I - \bar{T})v^k\|_2^2 \leq \|v^k - v^*\|_2^2. \tag{4.26}$$

But Lemma 4.1 implies

$$\begin{aligned}
&\|v^{k+1} - v^*\|_2^2 - \|\bar{T}v^k - v^*\|_2^2 \\
&\quad = \|E^k\delta^k\|_2^2 + 2(E^k\delta^k)^\intercal(\bar{T}v^k - v^*) \\
&\quad \leq \|E^k\delta^k\|_2\big(\|E^k\delta^k\|_2 + 2\|\bar{T}v^k - v^*\|_2\big).
\end{aligned}$$

Assumption 4.1(b) and (4.22) therefore imply

$$\begin{aligned}
&\|v^{k+1} - v^*\|_2^2 - \|\bar{T}v^k - v^*\|_2^2 \\
&\quad \leq \bar{e}\|\delta^k\|_2\big(\bar{e}\rho + 2\|v^0 - v^*\|_2 + 2\bar{e}\Delta\big) \\
&\quad = \kappa\,\|\delta^k\|_2,
\end{aligned} \tag{4.27}$$

where

$$\kappa = \bar{e}^2(\rho + 2\Delta) + 2\bar{e}\|v^0 - v^*\|_2.$$

Combining the inequality (4.27) with (4.26) yields

$$\|v^{k+1} - v^*\|_2^2 + \|(I - \bar{T})v^k\|_2^2 \leq \|v^k - v^*\|_2^2 + \kappa\|\delta^k\|_2.$$

Summing both sides of this inequality for $0 \leq k \leq N$ yields

$$\begin{aligned}\sum_{k=0}^{N}\|(I - \bar{T})v^k\|_2^2 &\leq \|v^0 - v^*\|_2^2 + \kappa\sum_{k=0}^{N}\|\delta^k\|_2 \\ &\leq \|v^0 - v^*\|_2^2 + \kappa\Delta\end{aligned}$$

for all N. Since the right hand side is finite for all N, this implies that $\|(I-\bar{T})v^k\|_2^2 \to 0$ a.s. as $k \to \infty$.

Part II (to prove properties (i)-(iii) in Theorem 4.1): Since the reference problem (4.3) with Assumption 4.1(a) and (c) satisfies the conditions of Lemma 4.3, the optimal dual variable λ^* (i.e. $\lambda^*_{\delta=0}$) in (4.4) is bounded. Additionally, since Assumption 4.1(a) implies that the feasible set of the convex optimisation problem (4.3) is compact, the set of all optimal solutions $\mathcal{X} \ni x^*$ is compact and convex. Hence $\mathbf{Fix}(T) = \{v^* \,|\, \forall x^* \in \mathcal{X}, v^* = x^* + \gamma\lambda^*\}$ is convex and compact. Therefore, for any given $\epsilon > 0$, choose $r \in (0, \epsilon]$ and consider the following compact set,

$$\Omega_r := \{v \in \mathbb{R}^n \,|\, \mathbf{dist}(v, \mathbf{Fix}(T)) \leq r\}. \tag{4.28}$$

Let $\alpha = \min_{\mathbf{dist}(v,\mathbf{Fix}(T))=r}\|(I - \bar{T})v\|_2^2$. Choose $\beta \in (0, \alpha)$ and let

$$\Omega_\beta := \{v \in \mathbb{R}^n \,|\, \|(I - \bar{T})v\|_2^2 \leq \beta\}. \tag{4.29}$$

Then, since $\|(I - \bar{T})v\|_2^2 \geq 0$ and $\|(I - \bar{T})v\|_2^2 = 0 \Leftrightarrow v \in \mathbf{Fix}(T)$, Ω_β is contained in the interior of Ω_r. Therefore we have $\mathbf{dist}(v, \mathbf{Fix}(T)) < \epsilon$ for all $v \in \Omega_\beta$. Furthermore, Part I showed that $\|(I - \bar{T})v^k\|_2^2 \to 0$ a.s., and it follows that

$$\mathbf{dist}(v^k, \mathbf{Fix}(T)) \to 0 \text{ a.s.} \tag{4.30}$$

Moreover, Assumption 4.1(b) implies $\|\delta^k\|_2 \to 0$ a.s. so (4.20) and Lemma 4.1 imply, for all $v \in \mathbb{R}^n$,

$$\|(\mathbf{prox}_{\gamma f^k} - \mathbf{prox}_{\gamma \bar{f}})v\|_2 \to 0 \text{ a.s.} \tag{4.31a}$$

$$\|(T^k - \bar{T})v\|_2 \to 0 \text{ a.s.} \tag{4.31b}$$

By substituting (4.31) into (4.12) we therefore have

$$\|x^k - \bar{x}(v^{k-1})\|_2 \to 0 \text{ a.s.}, \quad \bar{x}(v^{k-1}) := \mathbf{prox}_{\gamma \bar{f}} R_{\gamma g} v^{k-1}.$$

To complete the proof, we use the continuity of the objective function to infer that

$$\|\text{obj}(x^k, z^k) - \text{obj}(\bar{x}(v^{k-1}), z^k)\|_2 \to 0 \text{ a.s.}$$

and, by (4.30),

$$\|\text{obj}(\bar{x}(v^{k-1}), z^k) - \text{obj}^*\|_2 \to 0 \text{ a.s.}$$

and hence that $\|\text{obj}^k - \text{obj}^*\|_2 \to 0$ a.s. By an analogous argument we can also conclude that $\|\lambda^k - \lambda^*\|_2 \to 0$ and $\|x^k - z^k\|_2 \to 0$ a.s. □

4.4 Extension

We propose an extension of Theorem 4.1 that allows the assumption that the sequence $\{\|\delta^0\|_2, \|\delta^1\|_2, \ldots\}$ is almost surely summable to be relaxed. The following assumption replaces Assumption 4.1(c) with the requirement that the mean square estimate error $\mathbb{E}\left[\|\delta^k\|_2^2\right]$ converges with a rate of at least $O(1/k)$ in order to provide convergence in probability in Theorem 4.2.

Assumption 4.2. For all k we require that:

(a) $p(x^k) = \bar{p} + \delta^k$ results in a compact feasible set $x, z \in \mathcal{F}_f(\delta^k) \cap \mathcal{F}_g$ of problem (4.1).

(b) $\forall \delta \in \{\delta \,|\, \bar{p} + \delta \in \mathcal{P}\}$, $\forall y \in \mathcal{F}_f(\delta) \cap \mathcal{F}_g$, $\forall \mu \in -N_{\mathcal{F}_f(\delta)}(y)$, $\forall \omega \in N_{\mathcal{F}_g}(y)$, with $c^{max} < 1$,

$$\mu^\intercal \omega \leq c^{max} \|\mu\|_2 \|\omega\|_2.$$

(c) $\mathbb{E}\left[\|\delta^k\|_2^2\right] \to h(k)$ such that $0 \leq h(k) \leq C/k$ for some constant $C > 0$.

(d) The strong duality holds for the time-varying $\mathcal{L}_\gamma^k$ and the reference $\bar{\mathcal{L}}_\gamma$ Lagrangians associated with the ADMM.

Remark 4.3. The $O(1/k)$ rate of convergence of mean square error in Assumption 4.2(c) is achievable for many estimators in practice such as a Kalman filter with zero processing noise.

Lemma 4.4. *Under Assumption 4.2,* $\forall v^* \in \mathbf{Fix}(T)$,

$$\mathbb{E}\left[\mathbf{dist}\left(v^*, \mathbf{Fix}(T^k)\right)^2\right] \to 0. \tag{4.32}$$

Proof. According to the definition of $\mathbf{Fix}(T)$, we have $(I - \bar{T})v^* = 0$. Combining this result with Lemma 4.1 yields $T^k v^* - E^k \delta^k = v^*$. This implies

$$(I - T^k)v^* = E^k \delta^k. \tag{4.33}$$

The proof of this lemma follows a similar line of reasoning to the proof (Part II) of Theorem 4.1 as follows.

Since $\forall \delta \in \{\delta \,|\, \bar{p} + \delta \in \mathcal{P}\}$, the varying problem with Assumption 4.2(a),(b) satisfies the conditions of Lemma 4.3, the optimal dual variable λ_δ^* in (4.4) is bounded. Additionally, since Assumption 4.2(a) implies that the feasible set of the convex optimisation problem (4.3) is compact, the set of all optimal solutions $x_\delta^* \in \mathcal{X}_\delta$ is compact and convex. Hence $\mathbf{Fix}(T_\delta) = \{v_\delta^* \,|\, \forall x_\delta^* \in \mathcal{X}_\delta, v_\delta^* = v_\delta^* + \gamma \lambda_\delta^*\}$ is convex and compact, in which T_δ is the varying non-expansive operator defined similar to T^k, i.e, $T^k = T_\delta$ as $\delta = \delta^k$.

By combining (4.33) with Assumption 4.2(c) we have $\forall v^* \in \mathbf{Fix}(T)$,

$$\mathbb{E}\left[\|(I - T^k)v^*\|_2^2\right] \to 0,$$

which implies

$$(I - T^k)v^* \xrightarrow{\mathrm{P}} 0, \ \forall v^* \in \mathbf{Fix}(T). \tag{4.34}$$

We construct Ω_r, Ω_β as follows. $\forall \delta \in \{\delta \,|\, \bar{p} + \delta \in \mathcal{P}\} \cup \{0\}, \forall \epsilon > 0$, choose $r \in (0, \epsilon]$ and consider the following set

$$\Omega_r := \{v \in \mathbb{R}^n \,|\, \mathbf{dist}(v, \mathbf{Fix}(T_\delta)) \leq r\}. \tag{4.35}$$

Let $\alpha = \min_{\mathbf{dist}(v, \mathbf{Fix}(T_\delta)) = r} \|(I - T_\delta)v\|_2^2$. Choose $\beta \in (0, \alpha)$ and let

$$\Omega_\beta := \{v \in \mathbb{R}^n \,|\, \|(I - T_\delta)v\|_2^2 \leq \beta\}. \tag{4.36}$$

Then, since $\|(I - T_\delta)v\|_2^2 \geq 0$ and $\|(I - T_\delta)v\|_2^2 = 0 \Leftrightarrow v \in \mathbf{Fix}(T_\delta)$, Ω_β is contained in the interior of Ω_r. We therefore have

$$\mathbf{dist}(v, \mathbf{Fix}(T_\delta)) < \epsilon, \forall v \in \Omega_\beta. \tag{4.37}$$

Given ϵ and β greater than zero, as defined previously, (4.34) can be restated as

$$\mathbb{P}\left[v^* \notin \Omega_\beta(\delta) \,|\, \delta = \delta^k\right] \overset{k \to \infty}{\longrightarrow} 0.$$

Combining this with (4.37), we obtain

$$\mathbf{dist}(v^*, \mathbf{Fix}(T^k)) \overset{\mathrm{P}}{\to} 0. \tag{4.38}$$

We have already demonstrated that $\mathbf{Fix}(T_\delta)$ is contained within a compact set. Given that δ is also within a compact set, the sample space $\mathbf{dist}(v^*, \mathbf{Fix}(T_\delta))$ is inherently bounded. This combination—convergence in probability alongside a bounded sample space—yields mean square convergence, thereby establishing the proof of the lemma. □

Lemma 4.5. *Under Assumption 4.2, $\forall v^* \in \mathbf{Fix}(T)$,*

$$\mathbb{E}\left[\|T^k v^k - v^*\|_2^2\right] \leq \mathbb{E}\left[\|v^k - v^*\|_2^2\right] + \epsilon^k, \tag{4.39}$$

such that the non-negative iterate ϵ^k is bounded and $\epsilon^k \to 0$ as $k \to \infty$.

Proof. Let $v^* \in \mathbf{Fix}(T)$.

Given that $\mathbf{Fix}(T^k)$ is compact (as discussed in the proof of Lemma 4.4), and defining $\tilde{v}^k$ as a minimiser of the optimization problem associated with **dist** in Lemma 4.4, we have

$$\tilde{v}^k \in \underset{v \in \mathbf{Fix}(T^k)}{\arg\min} \|v^*, \mathbf{Fix}(T^k)\|_2.$$

It's worth noting that for all k, T^k is firmly non-expansive (4.8) and satisfies the following for all $v, w \in \mathbb{R}^n$:

$$\|T^k v - T^k w\|_2^2 + \|(I - T^k)v - (I - T^k)w\|_2^2 \leq \|v - w\|_2^2.$$

By substituting $v = v^k$ and $w = \tilde{v}^k$ into the inequality and utilizing the property $\tilde{v}^k \in \mathbf{Fix}(T^k)$, we deduce

$$\begin{aligned}
&\|T^k v^k - \tilde{v}^k\|_2^2 + \|(I - T^k)v^k\|_2^2 \leq \|v^k - \tilde{v}^k\|_2^2, \\
\Leftrightarrow\ &\|T^k v^k - v^* + v^* - \tilde{v}^k\|_2^2 + \|(I - T^k)v^k\|_2^2 \\
&\quad \leq \|v^k - v^* + v^* - \tilde{v}^k\|_2^2, \\
\Leftrightarrow\ &\|T^k v^k - v^*\|_2^2 + \|v^* - \tilde{v}^k\|_2^2 + 2(T^k v^k - v^*)^\intercal (v^* - \tilde{v}^k) \\
&\quad + \|T^k v^k - v^k\|_2^2 \leq \|v^k - v^*\|_2^2 + \|v^* - \tilde{v}^k\|_2^2 \\
&\quad + 2(v^k - v^*)^\intercal (v^* - \tilde{v}^k), \\
\Leftrightarrow\ &\|T^k v^k - v^*\|_2^2 + \|T^k v^k - v^k + v^* - \tilde{v}^k\|_2^2 \\
&\quad \leq \|v^k - v^*\|_2^2 + \|v^* - \tilde{v}^k\|_2^2, \\
\Rightarrow\ &\|T^k v^k - v^*\|_2^2 \leq \|v^k - v^*\|_2^2 + \|v^* - \tilde{v}^k\|_2^2, \\
\Rightarrow\ &\mathbb{E}\left[\|T^k v^k - v^*\|_2^2\right] \leq \mathbb{E}\left[\|v^k - v^*\|_2^2\right] + \mathbb{E}\left[\|v^* - \tilde{v}^k\|_2^2\right].
\end{aligned} \tag{4.40}$$

From Lemma 4.4 and the definition of $\tilde{v}^k$, we deduce that $\mathbb{E}\left[\|v^* - \tilde{v}^k\|_2^2\right] \to 0$. In the proofs of Theorem 4.1 and Lemma 4.4, we have shown that $\mathbf{Fix}(T)$ and $\mathbf{Fix}(T^k)$ are compact. This concludes the proof. □

Lemma 4.6. *Under Assumption 4.2, $\forall v^* \in \mathbf{Fix}(T)$, we have*

$$\|\mathbb{E}\left[(E^k \delta^k)^\intercal (T^k v^k - v^*)\right]\|_2 \to 0 \text{ as } k \to \infty. \tag{4.41}$$

Proof. The Cauchy-Schwarz inequality implies

$$\begin{aligned}
&\|\mathbb{E}\left[(E^k\delta^k)^\intercal(T^kv^k - v^*)\right]\|_2 \\
&\leq \sqrt{\mathbb{E}\left[\|E^k\delta^k\|_2^2\right]\mathbb{E}\left[\|T^kv^k - v^*\|_2^2\right]} \\
&\leq \sqrt{\frac{C\bar{e}^2}{k}\mathbb{E}\left[\|T^kv^k - v^*\|_2^2\right]} \text{ (Assumption 4.2(c) \& Lemma 4.1)} \\
&\leq \sqrt{\frac{C_0}{k}\left(\mathbb{E}\left[\|T^0v^0 - v^*\|_2^2\right] + \sum_{n=0}^{k}\epsilon^n\right)} \text{ (Lemma 4.5)} \\
&\leq \sqrt{\frac{C_1 + \sum_{n=1}^{k}\epsilon^n}{k}} \text{ (T^0 is non-expansive, v^* bounded).} \qquad (4.42)
\end{aligned}$$

Given that, as $k \to \infty$, $\epsilon^k \to 0$ (as per Lemma 4.5), we can conclude that the term $\frac{1}{k}\sum_{n=1}^{k}\epsilon^n$ converges to 0. Since, in addition, we have $\frac{C_1}{k} \to 0$, the proof of the lemma is established. □

Theorem 4.2. *Under Assumption 4.2, the ADMM-PU iteration (4.10) converges as $k \to \infty$, with:*

(i) $\text{obj}^k - \text{obj}^* \xrightarrow{P} 0$,

(ii) $\lambda^k - \lambda^* \xrightarrow{P} 0$,

(iii) $x^k - z^k \xrightarrow{P} 0$.

Proof. The firmly non-expansive property of $\bar{T}$ implies, for instance by setting $v = v^k$ and $w = v^*$ in (4.4), that

$$\|\bar{T}v^k - v^*\|_2^2 + \|(I - \bar{T})v^k\|_2^2 \leq \|v^k - v^*\|_2^2. \qquad (4.43)$$

Combing this with Lemma 4.1 we obtain

$$\begin{aligned}
\|(I - \bar{T})v^k\|_2^2 &\leq -\|T^kv^k - E^k\delta^k - v^*\|_2^2 + \|v^k - v^*\|_2^2 \\
&= -\|T^kv^k - v^*\|_2^2 + \|v^k - v^*\|_2^2 + 2(E^k\delta^k)^\intercal(T^kv^k - v^*) - \|E^k\delta^k\|_2^2 \\
&\leq -\|T^kv^k - v^*\|_2^2 + \|v^k - v^*\|_2^2 + 2(E^k\delta^k)^\intercal(T^kv^k - v^*), \\
\Rightarrow \mathbb{E}\left[\|(I - \bar{T})v^k\|_2^2\right] &\leq \mathbb{E}\left[\|v^k - v^*\|_2^2\right] - \mathbb{E}\left[\|T^kv^k - v^*\|_2^2\right] \\
&\quad + \|\mathbb{E}\left[(E^k\delta^k)^\intercal(T^kv^k - v^*)\right]\|_2,
\end{aligned}$$

$$\Rightarrow \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|(I-\bar{T})v^k\|_2^2\right] \leq \frac{1}{K}\underbrace{\mathbb{E}\left[\|v^1-v^*\|_2^2\right]}_{(a)} - \frac{1}{K}\mathbb{E}\left[\|T^kv^k-v^*\|_2^2\right] + \frac{1}{K}\sum_{k=1}^{K}\underbrace{\|\mathbb{E}\left[(E^k\delta^k)^\intercal(T^kv^k-v^*)\right]\|_2}_{(b)}. \tag{4.44}$$

From Lemma 4.5, we observe that the term labelled (a) in (4.44) is bounded from above by $\mathbb{E}\left[\|v^0-v^*\|_2^2\right]+\epsilon^0$, which is necessarily finite. According to Lemma 4.6, the term labelled (b) in (4.44)(b) converges to 0. Thus, we conclude

$$\mathbb{E}\left[\|(I-\bar{T})v^k\|_2^2\right] \to 0. \tag{4.45}$$

Following analogous steps to those in Part II of the proof of Theorem 4.1 and the proof of Lemma 4.4, we derive:

$$\mathbf{dist}(v^k, \mathbf{Fix}(T)) \xrightarrow{\mathrm{P}} 0. \tag{4.46}$$

By continuing with the steps in Part II of the proof of Theorem 4.1, we conclude the proof of Theorem 4.2. □

4.5 Numerical Study

This section investigates the convergence of the proposed algorithm using numerical simulations. The example we consider is the following resource allocation problem:

$$\begin{aligned} \min_{x,z} \quad & (z-s^m)^\intercal Q(z-s^m)+c^\intercal z && \text{(4.47a)}\\ \text{subject to} \quad & x=z && \text{(4.47b)}\\ & \mathbf{1}_n^\intercal x = p && \text{(4.47c)}\\ & s^l \leq z \leq s^u && \text{(4.47d)} \end{aligned}$$

where $x, z, s^l, s^u, s^m \in \mathbb{R}^n$, $Q := \mathbf{diag}(q_1, q_2, \ldots, q_n) \succeq 0$, and $p \in \mathbb{R}$. This problem can be viewed as n decentralised energy suppliers collaborating in order to match an

unknown demand $\bar{p}$. The parameters of the problem are generated as follows:

$$\forall i : \begin{cases} q_i \sim U(0,2),\ c_i \sim U(400,600), \\ s_i^m \sim U(80,120),\ s_i^r \sim U(20,30), \\ s_i^l = s_i^m - s_i^r,\ s_i^u = s_i^m + s_i^r, \end{cases} \tag{4.48a}$$

$$\bar{p} \sim U(\mathbf{1}_n^\intercal(s^m - 0.5s^r), \mathbf{1}_n^\intercal(s^m + 0.5s^r)), \tag{4.48b}$$

with the modification that $q_1 = 0$ so that problem (4.47) is not strongly convex. We choose $n = 10$, and set $\gamma = 0.1$ as the penalty term for ADMM iterations.

The following estimator models are considered for p^k:

$$\begin{aligned} &\text{Model 1:} \quad p(x^k) \sim N_{tr}(\bar{p}, (10e^{-\frac{k}{5}})^2, d^l, d^u), \\ &\text{Model 2:} \quad p(x^k) \sim N_{tr}(\bar{p}, (10e^{-\frac{k}{10}})^2, d^l, d^u), \\ &\text{Model 3:} \quad p(x^k) = \frac{1}{k}\sum_{i=1}^{k} p^i_{\text{meas}},\ p^i_{\text{meas}} \sim N_{tr}(\bar{p}, 1^2, d^l, d^u), \end{aligned}$$

where $d^l = \bar{p} - \frac{1}{2}\mathbf{1}_n^\intercal s^r, d^u = \bar{p} + \frac{1}{2}\mathbf{1}_n^\intercal s^r$. Models 1 and 2 can be interpreted as estimators that converge in mean with high (Model 1) and low (Model 2) convergence rates. Since the probability distribution of p^k (and hence that of δ^k) has constant support for all k, the almost sure convergence requirement of Assumption 1(b) does not hold. However, bounds on δ^k corresponding to any confidence level less than 1 for Models 1 and 2 are exponentially convergent and thus have finite l^1-norm. Model 3 estimates $\bar{p}$ by taking the running average of the i.i.d. measurements represented by p^k_{meas}. As in the case of Models 1 and 2, Model 3 does not satisfy Assumption 1(b) (almost sure convergence), and moreover the bounds on $\mathbb{E}\left[\|\delta^k\|_2^2\right]$ corresponding to a confidence level less than 1 converge at the rate of $1/k$ in this case, and therefore do not have finite l^1-norm. On the other hand all the three models satisfy Assumption 4.2(c).

We sample (4.48) for one instance and pass this instance to the proposed ADMM-PU algorithm (4.10) with each of the estimator models. The evolution of the fractional error in the objective value, $|\text{obj}^k - \text{obj}^*|/\text{obj}^*$, representing the residual error after k iterations, is shown in Fig. 4.1. For benchmarking purposes, we compute (using any capable solver) at each time step k the solution of the deterministic problem defined by (4.2) with the most recent estimate p^k, and this is denoted as $\text{opt}(p(x^k))$

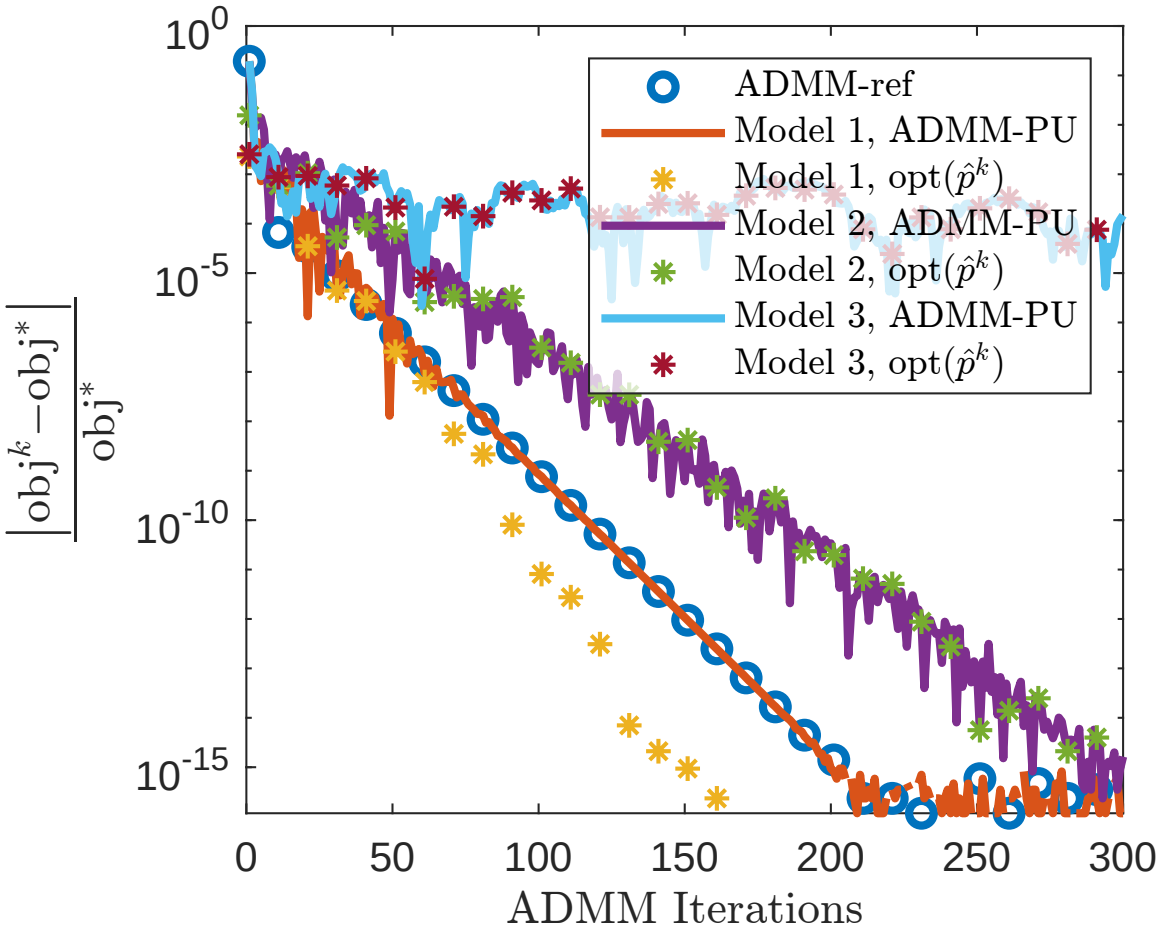


Figure 4.1: Simulation results

in Fig. 4.1. We also run the standard ADMM iteration (4.5) with the true value of $\bar{p}$ to solve problem (4.3) directly, and the evolution of these iterations is shown in Fig. 4.1 as ADMM-ref.

From the residual errors obtained with the Model 1 estimator (Fig. 4.1), it can be seen that ADMM-PU converges at the same rate as ADMM-ref, which is slower than opt($p(x^k)$). This is to be expected, because with the Model 1 estimator, $p(x^k)$ converges faster than ADMM-ref, while the convergence rate of ADMM-ref is necessarily an upper bound for the convergence rate of ADMM-PU. With the Model 2 estimator, $p(x^k)$ converges more slowly than ADMM-ref, and the proposed ADMM-PU algorithm is therefore able to track opt($p(x^k)$) closely. With the Model 3 estimator, the estimation error δ^k converges more slowly than for Models 1 and 2. Fig. 4.1 shows that ADMM-PU is sufficiently robust to track opt($p(x^k)$) in this case.

4.6 Conclusions

This chapter proposes a variant of ADMM which combines simultaneous iterations for optimisation and parameter estimation. The proposed approach guarantees convergence almost surely, provided that the sub-problem solved at each ADMM iteration takes the form of a mp-QP and the estimation error converges almost surely. Simulation results demonstrate that the proposed algorithm tracks the solution of the deterministic problem defined in terms of the most recent parameter estimate, provided the estimator converges more slowly than the standard ADMM iteration with no parameter uncertainty. For cases in which the estimator converges faster than this, the proposed algorithm converges at its maximum rate, which is the same rate as the standard ADMM iteration with no parameter uncertainty.

There are several directions to extend this work. Lemma 4.1 implies that the *proximal operator* of the time varying objective function (4.2) (i.e. $\mathbf{prox}_{\gamma f^k}(\cdot)$) can be expressed in terms of the proximal operator of the original objective of problem (4.1) and a remainder term that is bounded by a linear function of the estimator error. This property is used in Theorem 4.1 to prove convergence of the proposed algorithm based on ADMM. However, the proximal operator is the *resolvent* of the subdifferential operator and appears in various splitting methods other than DRS [see 10]. This analysis approach can therefore be applied more generally to first order methods other than ADMM.

Apart from parametric uncertainty, the parameter estimation error may also arise due to communication delays in a distributed optimisation setting. For example, [14] puts forward an algorithmic framework for asynchronous iteration updates, and we are therefore able to investigate the convergence behaviour when delays and parametric uncertainty coexist through an extension of the current work. We also plan to investigate how the algorithm works in a real-world context, e.g. applications in power systems with uncertainty or online collaboration of robots.

Chapter 5

Online optimisation with the Recursive Fixed-point method: A framework of Stochastic Nonexpansive Operators

5.1 Introduction

Many numerical algorithms are constructed with the aim of finding the fixed points of a given operator. In the context of convex optimisation, a wide class of algorithms perform a search for the zeros of a monotone operator [10] by finding the fixed points of a proximal operator [12]. Such operators are nonexpansive, and their repeated application defines an iteration that converges to a fixed point. This is the basis of the alternating direction method of multipliers (ADMM) and the proximal gradient

method (derived using Douglas-Rachford splitting and forward-backward splitting respectively [85]), which have been successfully applied to large scale and distributed optimisation problems [1].

In applications of optimisation in control and decision-making, optimisation problems often involve parameters that are either unknown or subject to measurement uncertainty and must therefore be estimated. This context necessitates optimisation algorithms capable of running concurrently with online parameter estimation in the presence of noise. We consider a general convex optimisation problem:

$$\min_x f(x, \hat{\theta}^k), \tag{5.1}$$

where f is a convex function, and $\hat{\theta}^k$ is an estimate of the problem parameter with uncertainty, where k is the time stamp of recursive estimation. To solve this problem, we apply one of the first-order algorithms that can be formulated as the fixed-point iteration of a nonexpansive operator $x^{k+1} \leftarrow T(x^k, \hat{\theta}^k)$ (See Chapter 1 and [10]). This chapter investigates the convergence of this iteration by providing a framework of stochastic nonexpansive operators.

Stochastic optimisation [173] addresses optimisation problems under uncertainty. Scenario-based representations, value-at-risk measures, and sample average approximations of the objective function are widely employed in energy systems, supply chain management, and finance. For linear dynamic systems, Robust and Stochastic Model Predictive Control (RMPC and SMPC) [150] have been proposed to handle uncertainty. RMPC assumes uncertainty is confined within a bounded polytope over the predicted horizon, while SMPC employs a stochastic framework. The problem of finding the fixed point of an operator with uncertainty can be interpreted as identifying the invariant set of a Markov stochastic process. In [174], the existence of such an invariant set for stochastically averaged Markov chains is proven. For the specific case of the projected gradient method, strong convexity is assumed.

5.1.1 Related work

For the iteration disturbance that comes from the asynchrony of distributed coordinate updates, a stochastic coordinate update framework of fixed-point iterations of a deterministic operator is designed in [14], where at each fixed-point iteration one of the coordinates of the overall iterate has a finite probability to be selected and updated by a specific agent with no delay. Almost sure convergence is guaranteed for averaged operators, and a linear convergence rate is achieved with contractive mappings. Since only one agent is selected per iteration, the convergence may be slow with large-scale problems.

The convergence analysis of consensus distributed algorithms perturbed by asynchrony modelled by probabilistic agent updates is investigated in [17], [175], [176]. These papers rely on the specific averaging operator (a special case of the projection) to guarantee almost sure convergence with the randomised Mann iteration proposed in [175], [177]. Unlike the approach of [14], at each iteration, every agent has a finite probability to perform its update, and the averaging aggregator replaces the missing updates using earlier updates. The work of [176] discusses cases in which the operator is perturbed by external i.i.d. noise and considers Sub-Weibull noise models instead of Markov's inequality in order to obtain narrower noise bounds. However, in order for the averaged operator to converge, a challenging assumption (12) is made, and in practice, the i.i.d. external noise assumption also lacks guarantee, since not all types of disturbance can be modelled as separate external noise as discussed in Chapter 4. In particular, for the uncertainty arising from Bayesian parameter estimators, the noises are persistently correlated.

The convergence properties of fixed point iterations perturbed by stochastic disturbances are studied in [178]. This work provides conditions ensuring convergence (in probability) to a fixed point. However, the approach uses tightened bounds on step sizes to ensure convergence, potentially affecting convergence rates when there is no uncertainty, and moreover, it does not allow for independently designed parameter estimators. Depending on the type of parameter uncertainty and the available data,

a variety of parameter estimators may be needed in practice (such as Kalman filters, recursive least squares, and averaging filters), and the convergence guarantees should therefore be obtained for a general class of stable estimators.

5.1.2 Contribution

The convergence properties of the combination of a proximal algorithm with parameter estimation were considered in Chapter 4, where the combined optimisation and estimation iterations were considered as a coupled system involving coupled nonexpansive operators. In this chapter, we provide a general framework for analysing convergence and robustness to stochastic uncertainty in systems of coupled averaged operators, by showing that the overall system can be interpreted as an averaged nonexpansive operator. The benchmark solution, instead of being set to the optimisation problem associated with the mean value of the uncertain parameter, is chosen as the fixed-point of the mean operator, a novel concept which is proposed to describe the overall stable point of the operators parametrised with uncertainty. This framework allows for non-i.i.d. and finite converging parameter noise, while providing convergence bounds for the first and second moments of the iterates. The numerical study based on the consequently proposed recursive fixed-point method shows the resilience to noise that appears in nearly all of the problem parameters. We also observe the "advantage over perfectionism" effect, that the proposed algorithm exhibits better convergence results compared to the time-varying optimal solutions, which take obviously more computational efforts, with respect to the most recent estimations of the parameters. We also apply the Monte-Carlo method to estimate and demonstrate the convergence of the proposed mean operator.

5.2 A Review of Nonexpansive Operators

As introduced in Chapter 2, operator theory provides a tool to analyse the convergence of first-order algorithms, which is analogous to the stability analysis of control

systems. Therefore, we begin by investigating the key properties of the fixed point iteration of a nonexpansive operator.

Definition 5.1 (Momentum operator)**.** We define the *momentum* operator $P : \mathbb{R}^n \to \mathbb{R}^n$ as $P := (T - I)x,\ \forall x \in \mathbb{R}^n$. When $P(x)$ is single-valued (i.e. a function) we use the notation:

$$p(x) := (T - I)x. \tag{5.2}$$

Definition 5.2. The *fixed point set* $\mathbf{Fix}(T)$ of an operator T is defined as $\forall x^* \in \mathbf{Fix}(T),\ x^* = Tx^*$. For an operator T with fixed points, the *fixed-point iteration* is defined as:

$$x^{k+1} := Tx^k = x^k + p(x^k), \tag{5.3}$$

where k is the temporal index for iterative steps.

Remark 5.1. The momentum can be viewed as the generalised gradient step associated with the fixed-point iteration $x^{k+1} \leftarrow Tx$. For an operator T with fixed points, $\forall x^* \in \mathbf{Fix}(T),\ p(x) = 0$.

Lemma 5.1 (Browder-Göhde-Kirk [179], [180])**.** *Let C be a compact and convex set in $\mathbb{R}^n$. If $T : C \to C$ is a nonexpansive operator, then T has at least one fixed point in C.*

Lemma 5.2. *For a nonexpansive operator T, $\forall q,\ X(q) := \{x \,|\, p(x) = q\}$ is closed and convex.*

Proof. [10, Sec. 3] discusses the case when $q = 0$ (i.e., fixed-point set of T), and here we expand this result to $\forall q$. Since $p(x)$ is continuous (Corollary 5.1), X is closed. $\forall x_A, x_B \in \mathbb{R}^n$ such that $p(x_A) = p(x_B) = q$, $\forall \theta \in [0, 1],\ x_C = \theta x_A + (1 - \theta)x_B$, from (2.25) we have:

$$\begin{aligned}
&\|x_C + p(x_C) - x_A - q\| \leq \|x_C - x_A\| = (1 - \theta)\|x_B - x_A\|, \\
&\|x_C + p(x_C) - x_B - q\| \leq \|x_C - x_B\| = \theta\|x_B - x_A\|, \\
\Rightarrow\ &\|x_C + p(x_C) - x_A - q\| + \|x_C + p(x_C) - x_B - q\| \leq \|x_B - x_A\|.
\end{aligned} \tag{5.4}$$

On the other hand, with triangular inequality applied:

$$\|x_B - x_A\| \leq \|x_C + p(x_C) - x_A - q\| + \|x_C + p(x_C) - x_B - q\|.$$

Therefore all the inequalities in (5.4) hold with equality, hence $x_C + p(x_C) = \theta(x_A + q) + (1-\theta)(x_B + q) = \theta x_A + (1-\theta)x_B + q = x_C + q$. □

Remark 5.2. When we study fixed-point iteration algorithms to solve convex optimisation problems in the engineering field, we usually assume that **Range**(T) is convex and compact, and the existence of fixed points. This is because in practice: (i) we assume the existence of an optimal solution that corresponds to a fixed point of T; (ii) the bounds of the iterates which are associated with the decision variables are convex and compact.

Lemma 5.3. *For an averaged operator T defined by (2.27), $\forall x_A, x_B \in \mathbb{R}^n$ we have:*

$$\|Tx_A - Tx_B\|^2 + \beta \|(T-I)x_A - (T-I)x_B\|^2 \leq \|x_A - x_B\|^2, \tag{5.5}$$

where $\beta := \alpha^{-1} - 1$ with α defined in (2.27). We combine (5.5) and (5.2) to have:

$$\beta \|\Delta p\|^2 + \|\Delta x + \Delta p\|^2 \leq \|\Delta x\|^2 \tag{5.6a}$$

$$\Rightarrow (1+\beta)\|\Delta p\|^2 + 2\langle \Delta x, \Delta p\rangle \leq 0, \tag{5.6b}$$

where $\Delta x := x_A - x_B$ and $\Delta p := p(x_A) - p(x_B)$.

Remark 5.3. As $\beta \to 0$ (and hence $\alpha \to 1$ in (2.27)), the averaged operator T becomes a general nonexpansive operator since (5.5) becomes (2.25). Moreover, (5.6b) shows that $p(x)$ is a negative-monotonic function.

Corollary 5.1. *For an averaged operator T defined by (2.27), $\forall x_A, x_B \in \mathbb{R}^n$ we have:*

$$\|\Delta x\|\|\Delta p\| \geq \frac{1+\beta}{2}\|\Delta p\|^2, \tag{5.7}$$

where $\beta := \alpha^{-1} - 1$ with α defined in (2.27), $\Delta x := x_A - x_B$ and $\Delta p := p(x_A) - p(x_B)$. We apply the Cauchy-Schwartz inequality to (5.6b) to obtain (5.7). Therefore $p(x)$ is Lipschitz continuous.

Lemma 5.4. *For an L-contractive operator T, $\forall x_A, x_B \in \mathbb{R}^n$ we have:*

$$\gamma \|\Delta x\|^2 + \|\Delta x + \Delta p\|^2 \leq \|\Delta x\|^2 \tag{5.8a}$$

$$\Rightarrow \gamma \|\Delta x\|^2 + \|\Delta p\|^2 + 2\langle \Delta x, \Delta p \rangle \leq 0, \tag{5.8b}$$

where $\gamma = 1 - L^2$, $\Delta x := x_A - x_B$ and $\Delta p := p(x_A) - p(x_B)$.

Remark 5.4. (5.8) can be viewed as replacing $\beta \|\Delta p\|^2$ in (5.6) with $\gamma \|\Delta x\|^2$. Similar to (5.6a)(5.6b), $L \to 1$ as $\gamma \to 0$, and the contractive operator T becomes a general nonexpansive operator.

For a nonexpansive operator T on $\mathbb{R}^n$ with fixed points, $\forall x_A, x_B \in \mathbb{R}^n$, we have the following four perspectives to view Δx and Δp in (5.6) and (5.8).

Perspective 1: $\|\Delta x\| \to 0$.

Lemma 5.5 (Monotonic $\|p(x)\|^2$ along the field line of $p(x)$)**.** *For a nonexpansive operator T on $\mathbb{R}^n$, $\forall t \in [0, +\infty)$ and $\forall x(0) \in \mathbb{R}^n$, we create a path C with $dx(t) = p(x(t))dt$. Then $\frac{d}{dt}\|p(x(t))\|^2 \leq 0$.*

Proof. To show this:

$$d(\|p(x(t))\|^2) = 2\langle p(t), dp \rangle = 2\langle \frac{dx(t)}{dt}, dp \rangle \leq -c \leq 0, \tag{5.9}$$

where $c \geq 0$, $c = (1 + \beta)\frac{\|dp\|^2}{dt}$ if T is averaged (5.6b), and $c = \frac{\gamma\|dx(t)\|^2 + \|dp\|^2}{dt}$ if T is contractive (5.8b). □

Perspective 2: $\Delta x = x - x^*$.

Definition 5.3 (Class κ functions [181], [182])**.** A function $f : [0, +\infty) \to [0, +\infty)$ is said to belong to *class* κ if:

- Strictly increasing: $f(x) < f(y)$, $\forall x < y$.
- Zero at zero: $f(0) = 0$.

A class κ function f belongs to *class* κ_∞ if $f(x) \to +\infty$ as $x \to \infty$.

Class κ functions, as an analytical tool, are widely used to analyse Lyapunov stability, compared to the geometric analysis used in Chapter 4. Now we bound $\|p(x)\|^2$ with class κ functions.

Lemma 5.6. *For a nonexpansive operator $T \neq I$ on $\mathbb{R}^n$ with fixed points, $\|p(x)\|^2$ is bounded by class κ functions from above and below:*

$$\kappa_l(\mathbf{dist}^2(x, \mathbf{Fix}(T))) \leq \|p(x)\|^2 \leq \kappa_u(\mathbf{dist}^2(x, \mathbf{Fix}(T))), \ \forall x \in \mathbb{R}^n. \tag{5.10}$$

Proof. We have the right-hand side inequality as a result of (5.7). From Lemma 2.5, $\forall x \in \mathbb{R}^n$, if $\mathbf{dist}^2(x, \mathbf{Fix}(T))$ is finite, we have finite $\|p(x)\|^2$, hence the left-hand side inequality follows. Here we assume $T \neq I$ to guarantee that there exists $x \in \mathbb{R}^n$ such that $\mathbf{dist}^2(x, \mathbf{Fix}(T)) \neq 0$. □

Remark 5.5. From Lemma 5.5 the sub-level sets $\|p(x)\|^2 \leq c$ are closed and connected; we can construct the sub-level sets $\|p(x)\|^2 \leq c$ by expanding the fixed-point set along the direction of field lines of $p(x)$.

Lemma 5.7. *For a nonexpansive operator T on $\mathbb{R}^n$ with fixed points, consider the case where x^k is a random variable. Then the fixed-point iteration $x^{k+1} \leftarrow Tx^k$ satisfies:*

$$\mathbb{E}\left[\left\|x^{k+1} - x^*\right\|^2\right] \leq \mathbb{E}\left[\left\|x^k - x^*\right\|^2\right] - c, \ \forall x^* \in \mathbf{Fix}(T), \tag{5.11}$$

where $c \geq 0$, $c = \beta\mathbb{E}\left[\left\|p(x^k)\right\|^2\right]$ if T is averaged (5.6a), and $c = \gamma\mathbb{E}\left[\left\|x^k - x^*\right\|^2\right]$ if T is contractive (5.8a).

Perspective 3: $\Delta x = x^{k+1} - x^k = p(x^k)$**.**

Lemma 5.8 (Monotonic $\left\|p(x^k)\right\|^2$ along the fixed-point iteration)**.** *For a nonexpansive operator T on $\mathbb{R}^n$, we perform the fixed-point iteration $x^{k+1} \leftarrow Tx^k$ and have:*

$$\left\|p(x^{k+1})\right\|^2 \leq \left\|p(x^k)\right\|^2 - c, \tag{5.12}$$

where $c \geq 0$, $c = \beta\left\|p(x^{k+1}) - p(x^k)\right\|^2$ if T is averaged (5.6a), and $c = \gamma\left\|p(x^k)\right\|^2$ if T is contractive (5.8a).

Perspective 4: $\Delta x = x^k - \mathbb{E}\left[x^k\right]$.

Lemma 5.9. *For a nonexpansive operator T on $\mathbb{R}^n$ with fixed points, consider the case where x^k is a random variable. Then the fixed-point iteration $x^{k+1} \leftarrow Tx^k$ satisfies:*

$$\mathbf{Var}\left[x^{k+1}\right] \leq \mathbf{Var}\left[x^k\right] - c, \tag{5.13}$$

where $c \geq 0$, $c = \beta\mathbb{E}\left[\left\|p(x^k) - p(\mathbb{E}\left[x^k\right])\right\|^2\right]$ if T is averaged (5.6a), and $c = \gamma\mathbf{Var}\left[x^k\right]$ if T is contractive (5.8a).

5.3 Stochastic Nonexpansive Operators and the Recursive First-order Algorithm

For the parametric fixed-point iteration $x^{k+1} \leftarrow T^k(x^k) = T(x^k, \theta^k)$, in this section we assume that at time k both x^k and θ^k are random variables. Hence the following notations are used: $\rho[z]$ represents the (either continuous or discrete) probability density function (PDF) of the random variable z; $[z|k]$ denotes the event conditioning on the historical data $\{x^0, \theta^0, \ldots, \theta^k\}$, hence:

$$\rho_k[z] := \rho\left[z|x^0, \theta^0, \ldots, \theta^k\right]. \tag{5.14}$$

Furthermore, $\mathbb{E}_z[g(z)] := \int_z g(z)\rho[z]\,d\Omega_z$ is the expected value of $g(z)$ over the random variable z, where Ω denotes the sample space of z, and $\mathbf{Var}_z[g(z)] := \mathbb{E}_z\left[\|g(z) - \mathbb{E}_z[g(z)]\|^2\right]$ is the respective variance of $g(z)$. Finally, $z \xrightarrow{\mathrm{P}} y$ indicates x converges to y in probability.

Lemma 5.10 (Bias-Variance decomposition)**.** *For a random variable $x \in \mathbb{R}^n$, $\forall y \in \mathbb{R}^n$, we have:*

$$\mathbb{E}\left[\|x - y\|^2\right] = \mathbb{E}\left[\|x - \mathbb{E}[x]\|^2\right] + \|y - \mathbb{E}[x]\|^2. \tag{5.15}$$

Lemma 5.11 (Preservation under convex combination [11])**.** *The convex combination $\bar{T}$ of nonexpansive operators $\{T_i\}_{i=1}^M$, defined by*

$$\bar{T} = \sum_{i=1}^{M} a_i T_i, \quad \textit{where } a_i \geq 0 \textit{ and } \sum_{i=1}^{M} a_i = 1, \tag{5.16}$$

is a nonexpansive operator. Moreover, if at least one of the operators $T_i \in \{T_i\}_{i=1}^M$ *is averaged or contractive, then* $\bar{T}$ *is averaged or contractive.*

Remark 5.6. We notice that the $\mathbb{E}[\cdot]$ is a convex combination, and this inspires the main idea behind this work that explores the convergence of the fixed-point iteration of a nonexpansive operator with uncertainty.

Definition 5.4 (Convex Combination-Invariant (CCI) operators)**.** An operator $T(x) : \mathbb{R}^n \to \mathbb{R}^n$ with $\text{Range}(T) = \mathcal{C}$ is said to be convex combination-invariant if, for all $x_A, x_B \in \mathcal{C}$ and $\alpha \in [0, 1]$, we have:

$$T(\alpha x_A + (1-\alpha)x_B) = \alpha T(x_A) + (1-\alpha)T(x_B). \tag{5.17}$$

Remark 5.7. Compared with Lemma 5.11, this proposed definition focuses on mapping invariance rather than property preservation. Although defined for the two-weight case, it can be extended to infinite weights in convex combinations (see (5.16)) by successive grouping. Note that the expectation operator $\mathbb{E}[\cdot]$ is a convex combination, playing a pivotal role in Theorems 5.1 and 5.2. Below in Lemma 5.12, we provide examples of CCI operators relevant to optimisation algorithm-oriented operator theory introduced in Chapter-Section 2.2.

Lemma 5.12 (Examples of CCI Operators)**.** *The following operators* $T : \mathbb{R}^n \to \mathbb{R}^n$ *are convex combination-invariant (CCI) (Definition 5.4).*

(a) If $T = Ax + c$ *is an affine function, where* $H : \mathbb{R}^n \to \mathbb{R}^n$ *is linear. For example,* T *is the gradient of a quadratic function* $f(x) = \langle x, Hx + c\rangle$*,* $T(x) = \partial f(x) = \nabla x = (H + H^\intercal)x + c$*.*

(b) If T *is a proximal operator (See Example (2.6)[10], [12]):*

$$T(v) = \mathbf{prox}_{\lambda f}(v) = \arg\min_x \left(h(x) + \mathcal{I}_{\mathcal{C}}(x) + \frac{1}{2\lambda}\|x - v\|^2 \right), \tag{5.18}$$

where $\lambda > 0$*,* $f = h(x) + \mathcal{I}_{\mathcal{C}}(x)$*,* $\partial h(x)$ *is CCI (e.g.* $h(x) = \langle x, Hx + c\rangle$ *is a quadratic function as in (a)), and* $\mathcal{I}_{\mathcal{C}}(x)$ *is the indicator function of a convex equality constraint set* $\mathcal{C} : Ax = b$*.*

(c) If T is the projection onto a convex set $\mathcal{C}$. That is, $T(v) = \mathbf{proj}_{\mathcal{C}}(v) = \arg\min_x \mathcal{I}_{\mathcal{C}}(x)$, where $\mathcal{I}_{\mathcal{C}}$ is the indicator function.

(d) If T is the projection onto a affine set $\mathcal{C} : Ax = b$. That is, $T(v) = \mathbf{proj}_{\mathcal{C}:Ax=b}(v)$.

Proof. **Part(a):** Affine mapping $T = Ax + c$ is CCI since, for all $x_A, x_B \in \mathbb{R}^n \supseteq \mathbf{Range}(Ax + c)$, and $\alpha \in [0, 1]$:

$$\begin{aligned} &A\big(\alpha x_A + (1-\alpha)x_B\big) + (\alpha + 1 - \alpha)c \\ &= \alpha A x_A + \alpha c + (1-\alpha)Ax_B + (1-\alpha)c \\ &= \alpha\big(Ax_A + c\big) + (1-\alpha)\big(Ax_B + c\big), \end{aligned} \tag{5.19}$$

satisfying (5.17).

Part(b): For $x_A, x_B \in \mathbf{Range}(\mathbf{prox}_{\lambda f}(v))$, $\alpha \in [0, 1]$ and $\lambda > 0$,

$$\begin{aligned} x_A = \mathbf{prox}_{\lambda f}(v_A) = \arg\min_{x_A}\left(f(x_A) + \frac{1}{2\lambda}\|x_A - v_A\|^2\right) &\Leftrightarrow 0 \in \lambda\partial f(x_A) + x_A - v_A, \\ x_B = \mathbf{prox}_{\lambda f}(v_B) = \arg\min_{x}\left(f(x_B) + \frac{1}{2\lambda}\|x - v_B\|^2\right) &\Leftrightarrow 0 \in \lambda\partial f(x_B) + x_B - v_B, \\ \Rightarrow -\left(\lambda\partial h(x_A) + x_A - v_A\right) \in \lambda N_{\mathcal{C}}(x_A),\ -\left(\lambda\partial h(x_B) + x_B - v_B\right) &\in \lambda N_{\mathcal{C}}(x_B), \end{aligned} \tag{5.20}$$

where $N_{\mathcal{C}}(x) = \partial\mathcal{I}_{\mathcal{C}}(x)$ is the normal cone operator (See (2.47) and Figure 2.7). As $\mathcal{C} : Ax - b = 0$,

$$N_{\mathcal{C}}(x) = \{v | \langle v, y - x\rangle \le 0,\ \forall y \in \mathcal{C}\} = \mathbf{null}(A)^{\perp} = \mathbf{Range}(A^{\intercal}) = \mathbf{col}(A^{\intercal}), \tag{5.21}$$

with $\mathbf{Dom}(N_{\mathcal{C}}) = \mathcal{C}$. As $x_A, x_B \in \mathbf{Range}(\mathbf{prox}_{\lambda f}(x))$, $x_A, x_B \in \mathcal{C}$. From Definition 2.1 we have:

$$\lambda x_A + (1-\alpha)x_B \in \mathcal{C} \Leftrightarrow N_{\mathcal{C}}(\alpha x_A + (1-\alpha)x_B) = \mathbf{col}(A^{\intercal}). \tag{5.22}$$

As $\partial h(x) = \nabla h(x) = Ax + c$, from (a) we have :

$$\partial h(\alpha x_A + (1-\alpha)x_B) = \alpha\partial h(x_A) + (1-\alpha)\partial h(x_B) \tag{5.23}$$

We combine (5.20), (5.21), (5.22), and (5.23) to have:

$$-\alpha\left(\lambda\partial h(x_A) + x_A - v_A\right) - (1-\alpha)\left(\lambda\partial h(x_B) + x_B - v_B\right) \in \lambda\mathbf{col}(A^{\intercal})$$

$$\Rightarrow 0 \in \lambda \mathbf{col}(A^\intercal) + \lambda h(\alpha x_A + (1-\alpha)x_B) + \alpha x_A + (1-\alpha)x_B - (\alpha v_A + (1-\alpha v_B))$$

$$\Rightarrow \mathbf{prox}_{\lambda f}(\alpha x_A + (1-\alpha)x_B) = \alpha \mathbf{prox}_{\lambda f}(x_A) + (1-\alpha)\mathbf{prox}_{\lambda f}(x_B). \tag{5.24}$$

One may also apply Lagrangian duality KKT conditions to solve this problem and have the closed form (affine) solution.

Part(c): For all $x_A, x_B \in \mathcal{C}$ and $\alpha \in [0,1]$:

$$\begin{aligned}\mathrm{proj}_{\mathcal{C}}(\alpha x_A + (1-\alpha)x_B) &\overset{\text{Def. 2.1}}{=} \alpha x_A + (1-\alpha)x_B \\ &= \alpha \,\mathrm{proj}_{\mathcal{C}}(x_A) + (1-\alpha)\,\mathrm{proj}_{\mathcal{C}}(x_B),\end{aligned} \tag{5.25}$$

satisfying (5.17).

Part(d): If $\mathcal{C} : Ax = b$, then $\forall v \in \mathbb{R}^n \supseteq \mathbf{Range}(\mathbf{proj}_{\mathcal{C}}(v))$,

$$\mathbf{proj}_{\mathcal{C}}(v) = v - A^+(Av - b) = (I - A^+A)v - A^+b, \tag{5.26}$$

where A^+ is the pseudo-inverse of A [12], [183], is an affine function. Hence it is CCI with (a). On the other hand, it is a special case of (b) when $h(x) = 0$, hence CCI. □

Definition 5.5. A *stochastic operator* $T(x,\theta) : \mathbb{R}^n \times \Omega_\theta \to \mathbb{R}^n$ is an operator with a random variable $\theta \in \Omega_\theta$ (Ω_θ is the sample space of θ) as the parameter. The *momentum* of a stochastic operator is:

$$p(x,\theta) := (T - I)(x,\theta). \tag{5.27}$$

A *stochastic nonexpansive operator* $T(x,\theta)$ is a stochastic operator, of which the *mean operator* :

$$T_m(x) := \mathbb{E}_{\theta|x}[T(x,\theta)] \tag{5.28}$$

is a nonexpansive operator. If T_m is averaged or contractive, we say T is a *stochastic averaged or contractive operator*. And the resulting *mean momentum* is:

$$p_m(x) := \mathbb{E}_{\theta|x}[(T(x,\theta) - I)(x)]. \tag{5.29}$$

We define the *disturbance* $d(x,\theta)$ of a stochastic operator $T(x,\theta)$ as:

$$d(x,\theta) := T(x,\theta) - T_m(x) = T(x,\theta) - \mathbb{E}_{\theta|x}[T(x,\theta)]. \tag{5.30}$$

To show that $d(x,\theta)$ is *zero-mean*:

$$\mathbb{E}_{\theta|x}\left[d(x,\theta)\right] = \mathbb{E}_{\theta|x}\left[T(x,\theta) - T_m(x)\right] \overset{(5.28)}{=} 0. \tag{5.31}$$

We also define the *disturbance variance* $v_d(x)$ of a stochastic operator $T(x,\theta)$ as:

$$v_d(x) := \mathbb{E}_{\theta|x}\left[\|T(x,\theta) - T_m(x)\|^2\right] = \mathbb{E}_{\theta|x}\left[\|d(x,\theta)\|^2\right]. \tag{5.32}$$

Remark 5.8. We may view the stochastic nonexpansive operator as a form of time-inhomogeneous Markov chain operator, while we focus more on the nonexpansiveness of the mean operator instead of the transition probability density function. It is worth noting that the mean operator $T_m(x)$ and the disturbance variance $v_d(x)$ are defined by the expectation operator and are therefore deterministic.

Lemma 5.13. *For a stochastic nonexpansive operator $T(x,\theta) : \mathbb{R}^n \times \Omega \to \mathbb{R}^n$ and a function $g(x) : \mathbb{R}^n \to \mathbb{R}^n$, we have:*

$$\mathbb{E}_{x,\theta}\left[\langle d(x,\theta), g(x)\rangle\right] = 0. \tag{5.33}$$

Proof. This follows from

$$\mathbb{E}_{x,\theta}\left[\langle d(x,\theta), g(x)\rangle\right] = \mathbb{E}_x\left[\mathbb{E}_{\theta|x}\left[\langle d(x,\theta), g(x)\rangle\right]\right] = \mathbb{E}_x\left[\langle \mathbb{E}_{\theta|x}\left[d(x,\theta)\right], g(x)\rangle\right] = 0.$$

□

Assumption 5.1. For the stochastic averaged operator $T(x^k,\theta^k) : \mathbb{R}^n \times \Omega_\theta \to \mathbb{R}^n$, we conduct the fixed-point iteration $x^{k+1} \leftarrow T(x^k,\theta^k)$, where $\{\theta^k\}$ has time-varying distributions. We assume:

(a) (Mean operator convergence) As $k \to \infty$, $\left\|(T_m^k - T_m^\infty)(x)\right\|^2 \to 0, \forall x \in \mathbb{R}^n$, and we hereby define β_m^∞ as in (5.5) for T_m^∞.

(b) (Bounded innovation disturbance with unbounded $p_m^\infty(x)$)

$$\begin{aligned}\limsup_{k\to\infty} v_d^k(x^k) &\overset{(5.32)}{=} \limsup_{k\to\infty} \mathbb{E}_{\theta^k|x^k}\left[\left\|T(x^k,\theta^k) - \mathbb{E}_{\theta^k|x^k}\left[T(x^k,\theta^k)\right]\right\|^2\right] \\ &= \sigma^2 + \eta\left(\mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right)\right),\end{aligned} \tag{5.34}$$

in which we bound $\eta(\cdot) : [0, \infty) \to [0, \infty) \geq 0$ as $\forall x \in \mathbb{R}^n$:

$$\begin{aligned} \|p_m^\infty(x)\|^2 = \|(T_m^\infty - I)(x)\|^2 &\geq (\kappa_\sigma + \frac{\eta}{\beta_m^\infty}) \left(\mathbf{dist}^2\left(x, \mathbf{Fix}(T_m^\infty)\right)\right) \\ &\geq \left(1 + \frac{1}{c_\eta}\right) \frac{\eta}{\beta_m^\infty} \left(\mathbf{dist}^2\left(x, \mathbf{Fix}(T_m^\infty)\right)\right), \end{aligned} \tag{5.35}$$

where $\kappa_\sigma(\cdot)$ is a class κ_∞ function and $c_\eta \geq 0$.

(c) (Conditional independence) For the fixed-point iteration $x^{k+1} \leftarrow T(x^k, \theta^k)$: θ^k and x^k are independent conditioning on $\{x^0, \theta^0, \ldots, \theta^{k-1}\}$.

Remark 5.9. Regarding Assumption 5.1:

- In Assumption 5.1(b), in addition to the class κ lower bound deduced in Lemma 5.6, we assume the lower bound to be class κ_∞. This will be satisfied, for example, if $\mathbf{Range}(T_m^\infty)$ is a compact set (i.e. If $x^0 \in \mathbb{R}^n$, $\|p_m^\infty(x^0)\|^2 = \|x^1 - x^0\|^2 \geq \mathbf{dist}^2\left(x^0, \mathbf{Range}(T_m^\infty)\right)$, which is unbounded).

- In real-world applications, x^k is dependent on $\{x^0, \theta^0, \ldots, \theta^{k-1}\}$ by the definition of the fixed-point iteration; θ^k as the output from a filter is usually dependent on $\{\theta^0, \ldots, \theta^{k-1}\}$ (e.g. estimates from a low-pass filter); θ^k may also depend on $\{x^0, \ldots, x^{k-1}\}$ as for an estimator that collects the input signal from the controller (e.g. estimates from a Kalman filter). Hence, rather than assuming $\{\theta^k\}$ to be i.i.d., Assumption 5.1(c) allows for dependence between x^k and θ^k without the condition on $\{x^0, \theta^0, \ldots, \theta^{k-1}\}$.

Theorem 5.1. *For a stochastic averaged operator $T(x, \theta) : \mathbb{R}^n \times \Omega_\theta \to \mathbb{R}^n$, for any specific fixed-point iteration process $x^{k+1} \leftarrow T(x^k, \theta^k)$, $k \in \mathcal{K} = \{0, 1, 2, \ldots, K-1\}$, as $k \to \infty$:*

(a) If all Assumptions 5.1(a)(b)(c) are satisfied: The Cesàro mean, $\frac{1}{K}\sum_{k\in\mathcal{K}} \mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right)$, for this fixed-point iteration is bounded by Markov's inequality:

$$\mathbb{P}\left[\frac{1}{K}\sum_{k\in\mathcal{K}} \mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right) \geq \delta\right] \leq \frac{\sigma^2}{\beta_m^\infty \kappa_\sigma(\delta)} \overset{\delta\to\infty}{\to} 0. \tag{5.36}$$

- *If* $\sigma^2 \overset{k\to\infty}{\to} 0$*, we have:*

$$\frac{1}{K}\sum_{k\in\mathcal{K}} \mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right) \overset{P}{\to} 0. \tag{5.37}$$

(b) If all Assumptions 5.1(a)(b)(c) are satisfied: The Cesàro mean of the transitional expectations, $\frac{1}{K}\sum_{k\in\mathcal{K}} \mathbf{Var}\left[p(x^k,\theta^k)\right] = \frac{1}{K}\sum_{k\in\mathcal{K}} \mathbb{E}\left[\left\|x^{k+1}-x^k-\mathbb{E}\left[x^{k+1}-x^k\right]\right\|^2\right]$*, is bounded:*

$$\frac{1}{K}\sum_{k\in\mathcal{K}} \mathbf{Var}\left[p(x^k,\theta^k)\right] \leq \left(1+\frac{1}{\beta_m^\infty}\right)(1+c_\eta)\sigma^2. \tag{5.38}$$

(c) If all Assumptions 5.1(a)(b)(c) are satisfied, and the iteration process becomes stationary as $k\to\infty$*, then* $\mathbf{dist}^2\left(\mathbb{E}\left[x^k\right], \mathbf{Fix}(T_m^\infty)\right)$ *and* $\mathbb{E}\left[\left\|p(x^k,\theta^k)\right\|^2\right]$ *are bounded by*

$$\mathbf{dist}^2\left(\mathbb{E}\left[x^k\right], \mathbf{Fix}(T_m^\infty)\right) \leq \kappa_\sigma^{-1}\left(\frac{1+c_\eta}{1+\beta_m^\infty}\sigma^2\right), \tag{5.39}$$

$$\mathbb{E}\left[\left\|p(x^k,\theta^k)\right\|^2\right] = \mathbb{E}\left[\left\|x^{k+1}-x^k\right\|^2\right] \leq \left(1+\frac{1}{\beta_m^\infty}\right)(1+c_\eta)\sigma^2. \tag{5.40}$$

(d) If we only know Assumptions 5.1(a)(c) are satisfied, and T_m^∞ *is a convex combination-invariant (CCI) operator (See Definition 5.4), then*

$$\mathbf{dist}^2\left(\mathbb{E}\left[x^k\right], \mathbf{Fix}(T_m^\infty)\right) \overset{k\to\infty}{\to} 0. \tag{5.41}$$

Proof of Theorem 5.1. Without loss of generality, we locate the origin of the coordinate system in $\mathbf{Fix}(T_m^\infty)$ (i.e. assume that $x^* = 0 \in \mathbf{Fix}(T_m^\infty)$), where T_m^∞ is the (deterministic) asymptotic mean operator.

At time k, since x^k can be expressed as a function of $\{x^0, \theta^0, \ldots, \theta^{k-1}\}$, with Assumption 5.1(c), $\mathbb{P}\left[\theta^k|x^k\right]$ is equivalent to $\mathbb{P}\left[\theta^k|x^0,\theta^0,\ldots,\theta^{k-1}\right]$. Hence:

$$T_m^k(x^k) \overset{(5.28)}{:=} \mathbb{E}_{\theta^k|x^k}\left[T(x^k,\theta^k)\right] = \mathbb{E}_{\theta^k|k-1}\left[T(x^k,\theta^k)\right]. \tag{5.42}$$

Part 1: To Calculate $\left\|\mathbb{E}\left[x^{k+1}\right]-x^*\right\|^2$ $(= \left\|\mathbb{E}\left[x^{k+1}\right]\right\|^2$ as $x^* = 0$ assumed above without loss of generality):

At time, k for any possible $x^k \in \mathbb{R}^n$, we choose $x_A = x^k$, $x_B = x^*$, and apply (5.6b) with $T_m^k(x)$:

$$2\langle x^k, p_m^k(x^k)\rangle + (1+\beta_m^k)s_f(x^k) + m_f(x^k) = 0, \tag{5.43}$$

where: β_m^k and $p_m^k(x^k) := (T_m^k - I)(x^k)$ are defined according to (5.5)(5.29) for T_m^k; $s_f(x^k) := \left\|p_m^k(x^k)\right\|^2 = \left\|p_m^k(x^k) - p_m^k(x^*)\right\|^2$ is the square error of $p_m^k(x^k)$ with respect to $p_m^k(x^*)$ (the momentum of a fixed point); and $m_f(x^k)$ is the non-negative margin that fills the gap of the inequality in (5.6b). We notice that (5.43) is also equivalent to:

$$\begin{aligned} &2\langle \mathbb{E}\left[x^k\right], p_m^k(x^k)\rangle + \underbrace{2\langle x^k - \mathbb{E}\left[x^k\right], p_m^k(x^k) - p_m^k(\mathbb{E}\left[x^k\right])\rangle}_{\text{(i)}} \\ &+ 2\langle x^k - \mathbb{E}\left[x^k\right], p_m^k(\mathbb{E}\left[x^k\right])\rangle + (1+\beta_m^k)s_f(x^k) + m_f(x^k) = 0. \end{aligned} \tag{5.44}$$

At time k, for any possible $x^k \in \mathbb{R}^n$, we choose $x_A = x^k$, $x_B = \mathbb{E}\left[x^k\right]$ and re-apply (5.6b) to have:

$$\underbrace{2\langle x^k - \mathbb{E}\left[x^k\right], p_m^k(x^k) - p_m^k(\mathbb{E}\left[x^k\right])\rangle}_{=(5.44)\text{(i)}} + (1+\beta_m^k)s_e(x^k) + m_e(x^k) = 0, \tag{5.45}$$

where $s_e(x^k) := \left\|p_m^k(x^k) - p_m^k(\mathbb{E}\left[x^k\right])\right\|^2$ is the square error of $p_m^k(x^k)$ with respect to $p_m^k(\mathbb{E}\left[x^k\right])$, and $m_e(x^k)$ is the non-negative margin that fills the inequality. Combining (5.45) with (5.44) we obtain:

$$\begin{aligned} &2\langle \mathbb{E}\left[x^k\right], p_m^k(x^k)\rangle + 2\langle x^k - \mathbb{E}\left[x^k\right], p_m^k(\mathbb{E}\left[x^k\right])\rangle \\ &+ (1+\beta_m^k)s_f(x^k) + m_f(x^k) - (1+\beta_m^k)s_e(x^k) - m_e(x^k) = 0. \end{aligned} \tag{5.46}$$

Taking the expectation $\mathbb{E}_{x^k|k-1}\left[(5.46)\right]$ on both sides:

$$\begin{aligned} &2\langle \mathbb{E}\left[x^k\right], \mathbb{E}\left[p_m^k(x^k)\right]\rangle + 2\langle \underbrace{\mathbb{E}\left[x^k\right] - \mathbb{E}\left[x^k\right]}_{=0}, p_m^k(\mathbb{E}\left[x^k\right])\rangle \\ &+ \underbrace{\mathbb{E}\left[(1+\beta_m^k)s_f(x^k) + m_f(x^k) - (1+\beta_m^k)s_e(x^k) - m_e(x^k)\right]}_{\text{(i)}} \\ &= 2\langle \mathbb{E}\left[x^k\right], \mathbb{E}\left[p_m^k(x^k)\right]\rangle + (1+\beta_m^k)S_f^k + M_f^k - (1+\beta_m^k)S_e^k - M_e^k \end{aligned}$$

$$= 0. \tag{5.47}$$

where $S_f^k, M_f^k, S_e^k, M_e^k$ are defined by taking $\mathbb{E}[\cdot]$ of the respective squared errors and margins in (5.47)(i). We apply the bias-variance decomposition (5.15) to the squared errors S_f^k, S_e^k and obtain:

$$\begin{aligned} S_f^k &= \mathbb{E}\left[\left\|p_m^k(x^k) - 0\right\|^2\right] \\ &= \left\|\mathbb{E}\left[p_m^k(x^k)\right]\right\|^2 + \mathbb{E}\left[\left\|p_m^k(x^k) - \mathbb{E}\left[p_m^k(x^k)\right]\right\|^2\right] \\ &= \left\|\mathbb{E}\left[p_m^k(x^k)\right]\right\|^2 + \mathbf{Var}\left[p_m^k(x^k)\right], \end{aligned} \tag{5.48}$$

$$\begin{aligned} S_e^k &= \mathbb{E}\left[\left\|p_m^k(x^k) - p_m^k(\mathbb{E}\left[x^k\right])\right\|^2\right] \\ &= \left\|\mathbb{E}\left[p_m^k(x^k)\right] - p_m^k(\mathbb{E}\left[x^k\right])\right\|^2 + \mathbb{E}\left[\left\|p_m^k(x^k) - \mathbb{E}\left[p_m^k(x^k)\right]\right\|^2\right] \\ &= \Delta_e^k + \mathbf{Var}\left[p_m^k(x^k)\right], \end{aligned} \tag{5.49}$$

where $\Delta_e^k := \left\|\mathbb{E}\left[p_m^k(x^k)\right] - p_m^k(\mathbb{E}\left[x^k\right])\right\|^2$, which is the difference due to the nonlinearity of $p_m^k(x^k)$. Combining (5.48)(5.49) with (5.47) we have:

$$\begin{aligned} &2\langle \mathbb{E}\left[x^k\right], \mathbb{E}\left[p_m^k(x^k)\right] \rangle + \left\|\mathbb{E}\left[p_m^k(x^k)\right]\right\|^2 \\ &= \left\|\mathbb{E}\left[x^k\right]\right\|^2 + M_e^k + \Delta_e^k + \beta_m^k S_e^k - M_f^k - \beta_m^k S_f^k. \end{aligned} \tag{5.50}$$

With Assumption 5.1(c) and (5.2), we calculate $\mathbb{E}_{x^k,\theta^k}\left[p(x^k,\theta^k)\right]$:

$$\mathbb{E}_{x^k,\theta^k}\left[p(x^k,\theta^k)\right] \overset{\text{Assump.5.1(c)}}{=} \mathbb{E}_{x^k}\left[\mathbb{E}_{\theta^k|x^k}\left[p(x^k,\theta^k)\right]\right] \overset{(5.29)}{=} \mathbb{E}\left[p_m^k(x^k)\right]. \tag{5.51}$$

To calculate $\left\|\mathbb{E}\left[x^{k+1}\right]\right\|^2$:

$$\begin{aligned} \left\|\mathbb{E}\left[x^{k+1}\right]\right\|^2 &= \left\|\mathbb{E}\left[x^k\right] + \mathbb{E}_{x^k,\theta^k}\left[\underbrace{x^{k+1} - x^k}_{=p(x^k,\theta^k)}\right]\right\|^2 \\ &\overset{(5.51)}{=} \left\|\mathbb{E}\left[x^k\right]\right\|^2 + 2\langle \mathbb{E}\left[x^k\right], \mathbb{E}\left[p_m^k(x^k)\right] \rangle + \left\|\mathbb{E}\left[p_m^k(x^k)\right]\right\|^2 \\ &\overset{(5.50)}{=} \left\|\mathbb{E}\left[x^k\right]\right\|^2 + M_e^k + \Delta_e^k + \beta_m^k S_e^k - M_f^k - \beta_m^k S_f^k. \end{aligned} \tag{5.52}$$

Part 2: To calculate $\mathbf{Var}\left[x^{k+1}\right] = \mathbb{E}\left[\left\|x^{k+1} - \mathbb{E}\left[x^{k+1}\right]\right\|^2\right]$:

$$\mathbb{E}\left[\left\|x^{k+1} - \mathbb{E}\left[x^{k+1}\right]\right\|^2\right]$$

$$
\begin{aligned}
&\overset{(5.30)(5.29)}{=}\mathbb{E}_{x^k,\theta^k}\left[\left\|x^k+p_m^k(x^k)+d(x^k,\theta^k)-\left(\mathbb{E}\left[x^k\right]+\mathbb{E}_{x^k,\theta^k}\left[p(x^k,\theta^k)\right]\right)\right\|^2\right]\\
&\overset{(5.51)}{=}\mathbb{E}_{x^k,\theta^k}\left[\left\|x^k+p_m^k(x^k)+d(x^k,\theta^k)-\left(\mathbb{E}\left[x^k\right]+\mathbb{E}\left[p_m^k(x^k)\right]\right)\right\|^2\right]\\
&\overset{\text{A.5.1(c)}(5.33)}{=}\mathbb{E}_{x^k}\left[\mathbb{E}_{\theta^k|x^k}\left[\left\|d(x^k,\theta^k)\right\|^2+\left\|x^k+p_m^k(x^k)-\left(\mathbb{E}\left[x^k\right]+\mathbb{E}\left[p_m^k(x^k)\right]\right)\right\|^2\right]\right]\\
&=\mathbb{E}_{x^k}\left[\mathbb{E}_{\theta^k|x^k}\left[\left\|d(x^k,\theta^k)\right\|^2\right]+\left\|x^k+p_m^k(x^k)-\left(\mathbb{E}\left[x^k\right]+\mathbb{E}\left[p_m^k(x^k)\right]\right)\right\|^2\right]\\
&=\mathbb{E}\left[v_d^k(x^k)\right]+\mathbb{E}\left[\left\|x^k+p_m^k(x^k)-\left(\mathbb{E}\left[x^k+p_m^k(x^k)\right]\right)\right\|^2\right], \qquad (5.53)
\end{aligned}
$$

where $\mathbb{E}\left[v_d^k(x^k)\right] \overset{(5.32)}{=} \mathbb{E}_{x^k,\theta^k}\left[\left\|d(x^k,\theta^k)\right\|^2\right] = \mathbb{E}_{x^k,\theta^k}\left[\left\|T(x^k,\theta^k)-\mathbb{E}_{\theta^k|x^k}\left[T(x^k,\theta^k)\right]\right\|^2\right] \overset{(5.42)}{=} \mathbb{E}_{x^k,\theta^k}\left[\left\|T(x^k,\theta^k)-\mathbb{E}_{\theta^k|k-1}\left[T(x^k,\theta^k)\right]\right\|^2\right]$, as assumed in Assump. 5.1(b).

At time k, for any possible $x^k \in \mathbb{R}^n$, we choose $x_A = x^k$, $x_B = \mathbb{E}\left[x^k\right]$ and apply (5.6a) to obtain

$$
\begin{aligned}
&\left\|x^k+p_m^k(x^k)-\left(\mathbb{E}\left[x^k\right]+p_m^k(\mathbb{E}\left[x^k\right])\right)\right\|^2+\beta_m^k s_e(x^k)+m_e(x^k)\\
&=\left\|x^k-\mathbb{E}\left[x^k\right]\right\|^2, \qquad (5.54)
\end{aligned}
$$

where the square error $s_e(x^k)$ and the margin $m_e(x^k)$ are defined in (5.45). We take the expectation $\mathbb{E}_{x^k}\left[(5.54)\right]$ on both sides:

$$
\begin{aligned}
&\mathbb{E}\left[\left\|x^k+p_m^k(x^k)-\left(\mathbb{E}\left[x^k\right]+p_m^k(\mathbb{E}\left[x^k\right])\right)\right\|^2\right]+\beta_m^k S_e^k+M_e^k\\
&=\mathbb{E}\left[\left\|x^k-\mathbb{E}\left[x^k\right]\right\|^2\right]=\mathbf{Var}\left[x^k\right], \qquad (5.55)
\end{aligned}
$$

where the expected squared error S_e^k and margin M_e^k are defined in (5.47). We combine (5.55) with (5.53):

$$
\begin{aligned}
\underbrace{\mathbf{Var}\left[x^{k+1}\right]}_{(5.53)} &\overset{(5.15)}{=}\mathbb{E}\left[v_d^k(x^k)\right]+\mathbb{E}\left[\left\|x^k+p_m^k(x^k)-\mathbb{E}\left[x^k\right]-p_m^k(\mathbb{E}\left[x^k\right])\right\|^2\right]\\
&\quad-\underbrace{\left\|\mathbb{E}\left[x^k\right]+p_m^k(\mathbb{E}\left[x^k\right])-\mathbb{E}\left[x^k+p_m^k(x^k)\right]\right\|^2}_{=\Delta_e^k (5.49)}\\
&\overset{(5.55)}{=}\mathbf{Var}\left[x^k\right]+\mathbb{E}\left[v_d^k(x^k)\right]-\beta_m^k S_e^k-\Delta_e^k-M_e^k. \qquad (5.56)
\end{aligned}
$$

Part 3: To calculate $\mathbb{E}\left[\left\|x^{k+1}\right\|^2\right]$ and conclude the proof:

$$
\begin{aligned}
&\mathbb{E}\left[\left\|x^{k+1}\right\|^2\right]=\mathbf{Var}\left[x^{k+1}\right]+\left\|\mathbb{E}\left[x^{k+1}\right]\right\|^2=(5.52)+(5.56)\\
=&\mathbf{Var}\left[x^k\right]+\left\|\mathbb{E}\left[x^k\right]\right\|^2+\mathbb{E}\left[v_d^k(x^k)\right]-\beta_m^k S_f^k-M_f^k
\end{aligned}
$$

$$= \mathbb{E}\left[\left\|x^k\right\|^2\right] + \mathbb{E}\left[v_d^k(x^k)\right] - \beta_m^k S_f^k - M_f^k. \tag{5.57}$$

With Assumption 5.1(a), $(T_m^k - T_m^\infty)(x) \overset{k\to\infty}{\to} 0, \forall x \in \mathbb{R}^n$, therefore:

$$\left\|\mathbb{E}\left[x^{k+1}\right]\right\|^2 \text{ (5.52)} \overset{k\to\infty}{\to} \left\|\mathbb{E}\left[x^k\right]\right\|^2 + \bar{M}_e^k + \bar{\Delta}_e^k + \beta_m^\infty \bar{S}_e^k - \bar{M}_f^k - \beta_m^\infty \bar{S}_f^k, \tag{5.58a}$$

$$\mathbf{Var}\left[x^{k+1}\right] \text{ (5.56)} \overset{k\to\infty}{\to} \mathbf{Var}\left[x^k\right] + \mathbb{E}\left[v_d^k(x^k)\right] - \beta_m^\infty \bar{S}_e^k - \bar{\Delta}_e^k - \bar{M}_e^k, \tag{5.58b}$$

$$\mathbb{E}\left[\left\|x^{k+1}\right\|^2\right] \text{ (5.57)} \overset{k\to\infty}{\to} \mathbb{E}\left[\left\|x^k\right\|^2\right] + \mathbb{E}\left[v_d^k(x^k)\right] - \beta_m^\infty \bar{S}_f^k - \bar{M}_f^k, \tag{5.58c}$$

where $\{\bar{\Delta}_e^k, \bar{M}_e^k, \bar{S}_e^k, \bar{M}_f^k, \bar{S}_f^k\}$ are defined by replacing $p_m^k(\cdot)$ in $\{\Delta_e^k, M_e^k, S_e^k, M_f^k, S_f^k\}$ with $p_m^\infty(\cdot)$. As $k \to \infty$, we take the Cesàro mean of (5.58c), combine this with Assumption 5.1(b) and obtain:

$$\begin{aligned}
&\frac{1}{K}\sum_{k\in\mathcal{K}}\left(\beta_m^\infty \bar{S}_f^k + \bar{M}_f^k - \mathbb{E}\left[v_d^k(x^k)\right]\right) \overset{k\to\infty}{\to} \frac{1}{K}\sum_{k\in\mathcal{K}}\left(\mathbb{E}\left[\left\|x^k\right\|^2\right] - \mathbb{E}\left[\left\|x^{k+1}\right\|^2\right]\right) \\
\Rightarrow &\frac{1}{K}\sum_{k\in\mathcal{K}}\left(\beta_m^\infty \bar{S}_f^k + \bar{M}_f^k - \mathbb{E}\left[v_d^k(x^k)\right]\right) \overset{k\to\infty}{\to} 0 \\
\overset{\text{A.5.1(b)}}{\Rightarrow} &\limsup_{k\to\infty}\frac{1}{K}\sum_{k\in\mathcal{K}}\bar{S}_f^k = \limsup_{k\to\infty}\frac{1}{K}\sum_{k\in\mathcal{K}}\mathbb{E}_{x^k}\left[\left\|p_m^\infty(x^k)\right\|^2\right] \\
&\leq \frac{\sigma^2}{\beta_m^\infty} + \frac{1}{K}\sum_{k\in\mathcal{K}}\mathbb{E}_{x^k}\left[\frac{\eta\left(\mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right)\right)}{\beta_m^\infty}\right] \\
\overset{\text{A.5.1(b)}}{\Rightarrow} &\limsup_{k\to\infty}\frac{1}{K}\sum_{k\in\mathcal{K}}\mathbb{E}_{x^k}\left[\kappa_\sigma\left(\mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right)\right)\right] \leq \frac{\sigma^2}{\beta_m^\infty}.
\end{aligned} \tag{5.59}$$

We apply Markov's inequality to (5.59):

$$\begin{aligned}
&\mathbb{P}\left[\frac{1}{K}\sum_{k\in\mathcal{K}}\mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right) \geq \delta\right] \leq \mathbb{P}\left[\frac{1}{K}\sum_{k\in\mathcal{K}}\kappa_\sigma\left(\mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right)\right) \geq \kappa_\sigma(\delta)\right] \\
\overset{\text{(i)}}{\leq} &\frac{\frac{1}{K}\sum_{k\in\mathcal{K}}\mathbb{E}\left[\kappa_\sigma\left(\mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right)\right)\right]}{\kappa_\sigma(\delta)} \overset{(5.59)}{\leq} \frac{\sigma^2}{\beta_m^\infty \kappa_\sigma(\delta)},
\end{aligned} \tag{5.60}$$

in which (i) applied the linearity of $\mathbb{E}[\cdot]$, and this results in (5.36) of Theorem 5.1(a).

Similarly we have:

$$\begin{aligned}
&\frac{1}{K}\sum_{k\in\mathcal{K}}\mathbb{E}_{x^k}\left[\eta\left(\mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right)\right)\right] \\
\overset{\text{A.5.1(b)}}{\leq} &\frac{1}{K}\sum_{k\in\mathcal{K}}\beta_m^\infty c_\eta \mathbb{E}_{x^k}\left[\left\|p_m^\infty(x^k)\right\|^2 - \frac{\eta\left(\mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right)\right)}{\beta_m^\infty}\right]
\end{aligned}$$

$$\overset{(5.59)}{\leq} \frac{\beta_m^\infty c_\eta \sigma^2}{\beta_m^\infty} = c_\eta \sigma^2. \tag{5.61}$$

Then we calculate the Cesàro mean of the transitional expectation:

$$\begin{aligned}
&\frac{1}{K}\sum_{k\in\mathcal{K}} \mathbf{Var}_{x^k,\theta^k}\left[p(x^k,\theta^k)\right] \\
=&\frac{1}{K}\sum_{k\in\mathcal{K}} \mathbb{E}_{x^k,\theta^k}\left[\left\|p(x^k,\theta^k) - \mathbb{E}_{x^k,\theta^k}\left[p(x^k,\theta^k)\right]\right\|^2\right] \\
\overset{(5.30)}{=}&\frac{1}{K}\sum_{k\in\mathcal{K}} \mathbb{E}_{x^k,\theta^k}\left[\left\|p_m^k(x^k) + d(x^k,\theta^k) - \mathbb{E}\left[p_m^k(x^k)\right]\right\|^2\right] \\
\overset{\text{A.5.1(c)}}{=}&\frac{1}{K}\sum_{k\in\mathcal{K}} \mathbb{E}_{x^k,\theta^k}\left[\left\|d(x^k,\theta^k)\right\|^2 + \left\|p_m^k(x^k) - \mathbb{E}\left[p_m^k(x^k)\right]\right\|^2\right] \\
\leq&\frac{1}{K}\sum_{k\in\mathcal{K}} \mathbb{E}_{x^k,\theta^k}\left[\left\|d(x^k,\theta^k)\right\|^2\right] + \frac{1}{K}\sum_{k\in\mathcal{K}} \mathbb{E}\left[\left\|p_m^k(x^k)\right\|^2\right] \\
\overset{\text{A.5.1(a)(b)}}{\leq}&\sigma^2 + \frac{1}{K}\sum_{k\in\mathcal{K}} \mathbb{E}\left[\eta\left(\mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right)\right) + \left\|p_m^\infty(x^k)\right\|^2\right] \\
\overset{(5.59)}{\leq}&\left(1+\frac{1}{\beta_m^\infty}\right)\left(\sigma^2 + \frac{1}{K}\sum_{k\in\mathcal{K}} \mathbb{E}\left[\eta\left(\mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right)\right)\right]\right) \\
\overset{(5.61)}{\leq}&\left(1+\frac{1}{\beta_m^\infty}\right)(1+c_\eta)\sigma^2,
\end{aligned} \tag{5.62}$$

which proves Theorem 5.1(b).

If the iteration process reaches stationary condition (S.C.) as $k \to \infty$, we have:

$$0 \overset{\text{S.C.}}{=} \mathbb{E}\left[x^{k+1} - x^k\right] \overset{(5.30)(5.29)}{=} \mathbb{E}_{x^k,\theta^k}\left[p_m^\infty(x^k) + d(x^k,\theta^k)\right] = \mathbb{E}\left[p_m^\infty(x^k)\right]. \tag{5.63}$$

On the other hand, under S.C., all terms in (5.58) become constant. We focus on (5.58b):

$$\begin{aligned}
&\beta_m^\infty \bar{S}_e^k + \bar{\Delta}_e^k + \bar{M}_e^k && \overset{\text{S.C.}}{=} \mathbb{E}\left[v_d^k(x^k)\right] \\
\overset{(5.49)}{\Rightarrow}\ &\beta_m^\infty(\bar{\Delta}_e^k + \mathbf{Var}_{x\in\mathcal{K}}\left[p_m^\infty(x^k)\right]) + \bar{\Delta}_e^k + \bar{M}_e^k && \overset{\text{A.5.1(b)}}{\leq} \sigma^2 + \mathbb{E}_{x^k}\left[\eta\left(\mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right)\right)\right] \\
\Rightarrow\ &(1+\beta_m^\infty)\left\|\mathbb{E}\left[p_m^\infty(x^k)\right] - p_m^\infty(\mathbb{E}\left[x^k\right])\right\|^2 && \overset{(5.61)}{\leq} (1+c_\eta)\sigma^2 \\
&\overset{(5.63)}{\Rightarrow} \left\|p_m^\infty(\mathbb{E}\left[x^k\right])\right\|^2 && \leq \frac{1+c_\eta}{1+\beta_m^\infty}\sigma^2, \\
&\overset{\text{A.5.1(b)}}{\Rightarrow} \mathbf{dist}^2\left(\mathbb{E}\left[x^k\right], \mathbf{Fix}(T_m^\infty)\right) && \leq \kappa_\sigma^{-1}\left(\frac{1+c_\eta}{1+\beta_m^\infty}\sigma^2\right),
\end{aligned} \tag{5.64}$$

which proves (5.39) in Theorem 5.1(c); (5.40) follows directly from (5.38) under S.C.

With (5.37), which follows directly from (5.36) as $\sigma^2 \overset{k\to\infty}{\to} 0$, we have proved Theorem 5.1(a)(b)(c).

If we only know Assumptions 5.1(a)(c), and T_m^∞ is convex combination-invariant, we notice that $\mathbb{E}[\cdot]$ is a convex combination and have:

$$\mathbb{E}\left[x^{k+1}\right] = \mathbb{E}\left[T_m^k(x^k)\right] \overset{\text{Def. } 5.5}{=} \mathbb{E}\left[T_m^k(x^k)\right] \overset{k\to\infty}{\to} \mathbb{E}\left[T_m^\infty(x^k)\right] \overset{\text{Def. } 5.4}{=} T_m^\infty(\mathbb{E}\left[x^k\right]), \tag{5.65}$$

which means as $k \to \infty$, the $\mathbb{E}\left[x^{k+1}\right] \leftarrow \mathbb{E}\left[x^k\right]$ performs the fixed-point iteration under T_m^∞. Since T_m^∞ is averaged, $\mathbb{E}\left[x^k\right]$ converges to $\mathbf{Fix}(T_m^\infty)$ (See [10], [62], [63]). This concludes our proof. □

Similar to Theorem 5.1, for a stochastic contractive operator we have the following results.

Assumption 5.2. For the stochastic contractive operator $T(x^k, \theta^k) : \mathbb{R}^n \times \Omega_\theta \to \mathbb{R}^n$, we conduct the fixed-point iteration $x^{k+1} \leftarrow T(x^k, \theta^k)$, where $\{\theta^k\}$ has time-varying distributions. We assume:

(a) (Mean operator convergence) As $k \to \infty$, $\left\|(T_m^k - T_m^\infty)(x)\right\|^2 \to 0, \forall x \in \mathbb{R}^n$, where T_m^∞ is L-contractive With $\gamma_m^\infty = 1 - L^2$ defined as in (5.5).

(b) (Bounded innovation disturbance with $\|x\|^2$)

$$\begin{aligned}\limsup_{k\to\infty} \mathbb{E}\left[v_d^k(x^k)\right] &\overset{(5.32)}{=} \limsup_{k\to\infty} \mathbb{E}_{x^k,\theta^k}\left[\left\|T(x^k,\theta^k) - \mathbb{E}_{\theta^k|x^k}\left[T(x^k,\theta^k)\right]\right\|^2\right] \\ &= \sigma^2 + \eta\left(\mathbf{dist}^2\left(x^k, \mathbf{Fix}(T_m^\infty)\right)\right).\end{aligned} \tag{5.66}$$

in which we bound $\eta(\cdot) : [0,\infty) \to [0,\infty) \geq 0$ as $\forall x \in \mathbb{R}^n$:

$$\|x\|^2 \geq \left(1 + \frac{1}{c_\eta}\right)\frac{\eta}{\gamma_m^\infty}\left(\mathbf{dist}^2\left(x, \mathbf{Fix}(T_m^\infty)\right)\right), \tag{5.67}$$

where $c_\eta \geq 0$.

(c) (Conditional independence) For the fixed-point iteration $x^{k+1} \leftarrow T(x^k, \theta^k)$: θ^k and x^k are independent conditioning on $\{x^0, \theta^0, \ldots, \theta^{k-1}\}$.

Theorem 5.2. *For a stochastic contractive operator $T(x,\theta):\mathbb{R}^n\times\Omega_\theta\to\mathbb{R}^n$, for any specific fixed-point iteration process $x^{k+1}\leftarrow T(x^k,\theta^k),\ k\in\mathcal{K}=\{0,1,2,\dots,K-1\}$, as $k\to\infty$, $\forall x^*\in\mathbf{Fix}(T_m^\infty)$, the Cesàro mean $\frac{1}{K}\sum_{k\in\mathcal{K}}\left\|x^k-x^*\right\|^2$ is bounded:*

$$\mathbb{E}\left[\frac{1}{K}\sum_{k\in\mathcal{K}}\left\|x^k-x^*\right\|^2\right]\leq\frac{1+c_\eta}{\gamma_m^\infty}\sigma^2. \tag{5.68}$$

- *If Assumptions 5.2(a)(b)(c) are satisfied, and the process reaches stationary condition (S.C.),*

$$\mathbb{E}\left[\left\|x^k-x^*\right\|^2\right]\leq\frac{1+c_\eta}{\gamma_m^\infty}\sigma^2. \tag{5.69}$$

- *If Assumptions 5.2(a)(b)(c) are satisfied, and $\sigma^2\overset{k\to\infty}{\to}0$, we have:*

$$\mathbb{E}\left[\frac{1}{K}\sum_{k\in\mathcal{K}}\left\|x^k-x^*\right\|^2\right]\overset{k\to\infty}{\to}0. \tag{5.70}$$

- *If we only know Assumptions 5.2(a)(c) are satisfied, and T_m^∞ is a convex combination-invariant CCI operator (See Definition 5.4), then*

$$\mathbf{dist}^2\left(\mathbb{E}\left[x^k\right],\mathbf{Fix}(T_m^\infty)\right)\overset{k\to\infty}{\to}0. \tag{5.71}$$

Proof of Theorem 5.2. With the discussion in Remark 5.4, we refer to the proof of Theorem 5.1 to support the following arguments.

Without loss of generality, we locate the origin of the coordinates in $\mathbf{Fix}(T_m^\infty)$ (i.e. assume $x^*=0\in\mathbf{Fix}(T_m^\infty)$), where T_m^∞ is the (deterministic) asymptotic mean operator.

Part 1: To calculate $\mathbb{E}\left[\left\|x^{k+1}\right\|^2\right]$: At k, for any possible x^k:

$$\begin{aligned}
&\left\|T_m^k(x^k)-x^*\right\|^2\overset{(5.8a)}{=}\left\|x^k-x^*\right\|^2-\gamma_m^k\left\|x^k\right\|^2-m_f(x^k)\\
\Rightarrow&\left\|T_m^k(x^k)\right\|^2=\left\|x^k\right\|^2-\gamma_m^k\left\|x^k\right\|^2-m_f(x^k)\\
\Rightarrow&\,\mathbb{E}\left[\left\|T_m^k(x^k)\right\|^2\right]=\mathbb{E}\left[\left\|x^k\right\|^2\right]-\gamma_m^k\mathbb{E}\left[\left\|x^k\right\|^2\right]-M_f^k,
\end{aligned} \tag{5.72}$$

where $M_f^k=\mathbb{E}\left[m_f(x^k)\right]$ is the margin with respect to the fixed-point set of T_m^k.

$$\mathbb{E}\left[\left\|x^{k+1}\right\|^2\right]=\mathbb{E}\left[\left\|T(x^k,\theta^k)\right\|^2\right]\overset{(5.30)}{=}\mathbb{E}\left[\left\|T_m^k(x^k)+d(x^k,\theta^k)\right\|^2\right]$$

$$
\begin{aligned}
&\overset{(5.42)}{=} \mathbb{E}_{x^k}\left[\mathbb{E}_{\theta^k|x^k}\left[\left\|T_m^k(x^k)+d(x^k,\theta^k)\right\|^2\right]\right]\\
&\overset{(5.33)\text{A.5.2(c)}}{=} \mathbb{E}\left[\left\|T_m^k(x^k)\right\|^2\right]+\mathbb{E}\left[\left\|d(x^k,\theta^k)\right\|^2\right]\\
&\overset{(5.72)\text{A.5.2(5.32)}}{=} \mathbb{E}\left[\left\|x^k\right\|^2\right]-\gamma_m^k\mathbb{E}\left[\left\|x^k\right\|^2\right]-M_f^k+\mathbb{E}\left[v_d^k(x^k)\right]. \qquad (5.73)
\end{aligned}
$$

Part 2: Concluding the proof. With Assumption 5.2(a) we have $(T_m^k - T_m^\infty)(x) \overset{k\to\infty}{\to} 0$ $\forall x \in \mathbb{R}^n$, therefore:

$$
\mathbb{E}\left[\left\|x^{k+1}\right\|^2\right] \overset{(5.73)}{\overset{k\to\infty}{\to}} \mathbb{E}\left[\left\|x^k\right\|^2\right]+\mathbb{E}\left[v_d^k(x^k)\right]-\gamma_m^\infty\mathbb{E}\left[\left\|x^k\right\|^2\right]-\bar{M}_f^k, \qquad (5.74a)
$$

where $\{\bar{\Delta}_e^k, \bar{M}_e^k, \bar{S}_e^k, \bar{M}_f^k, \bar{S}_f^k\}$ are defined by replacing $p_m^k(\cdot)$ in $\{\Delta_e^k, M_e^k, S_e^k, M_f^k, S_f^k\}$ with $p_m^\infty(\cdot)$. As $k \to \infty$, we take the Cesàro mean of (5.74a), combine this with Assumption 5.2(b) and obtain:

$$
\begin{aligned}
&\frac{1}{K}\sum_{k\in\mathcal{K}}\left(\gamma_m^\infty\mathbb{E}\left[\left\|x^k\right\|^2\right]+\bar{M}_f^k-\mathbb{E}\left[v_d^k(x^k)\right]\right) \overset{k\to\infty}{\to} \frac{1}{K}\sum_{k\in\mathcal{K}}\left(\mathbb{E}\left[\left\|x^k\right\|^2\right]-\mathbb{E}\left[\left\|x^{k+1}\right\|^2\right]\right)\\
&\Rightarrow \frac{1}{K}\sum_{k\in\mathcal{K}}\left(\gamma_m^\infty\mathbb{E}\left[\left\|x^k\right\|^2\right]+\bar{M}_f^k-\mathbb{E}\left[v_d^k(x^k)\right]\right) \overset{k\to\infty}{\to} 0.\\
&\overset{\text{A.5.2(b)}}{\Rightarrow} \limsup_{k\to\infty}\frac{1}{K}\sum_{k\in\mathcal{K}}\mathbb{E}\left[\left\|x^k\right\|^2\right] \le \frac{\sigma^2}{\gamma_m^\infty}+\frac{1}{K}\sum_{k\in\mathcal{K}}\mathbb{E}_{x^k}\left[\frac{\eta\left(\mathbf{dist}^2\left(x^k,\mathbf{Fix}(T_m^\infty)\right)\right)}{\gamma_m^\infty}\right]. \qquad (5.75)
\end{aligned}
$$

We take the Cesàro mean of $\eta\left(\mathbf{dist}^2\left(x^k,\mathbf{Fix}(T_m^\infty)\right)\right)$:

$$
\begin{aligned}
&\frac{1}{K}\sum_{k\in\mathcal{K}}\mathbb{E}_{x^k}\left[\eta\left(\mathbf{dist}^2\left(x^k,\mathbf{Fix}(T_m^\infty)\right)\right)\right]\\
&\overset{\text{A.5.2(b)}}{\le} \frac{1}{K}\sum_{k\in\mathcal{K}}\gamma_m^\infty c_\eta\mathbb{E}_{x^k}\left[\left\|x^k\right\|^2-\frac{\eta\left(\mathbf{dist}^2\left(x^k,\mathbf{Fix}(T_m^\infty)\right)\right)}{\gamma_m^\infty}\right]\\
&\overset{(5.75)}{\le} \frac{\gamma_m^\infty c_\eta\sigma^2}{\gamma_m^\infty} = c_\eta\sigma^2. \qquad (5.76)
\end{aligned}
$$

Therefore from (5.75) we have:

$$
\begin{aligned}
&\limsup_{k\to\infty}\frac{1}{K}\sum_{k\in\mathcal{K}}\mathbb{E}\left[\left\|x^k\right\|^2\right] \le \frac{\sigma^2}{\gamma_m^\infty}+\frac{1}{K}\sum_{k\in\mathcal{K}}\mathbb{E}_{x^k}\left[\frac{\eta\left(\mathbf{dist}^2\left(x^k,\mathbf{Fix}(T_m^\infty)\right)\right)}{\gamma_m^\infty}\right]\\
&\overset{(5.76)}{\le} \frac{1+c_\eta}{\gamma_m^\infty}\sigma^2, \qquad (5.77)
\end{aligned}
$$

which proves (5.68). Finally (5.69) and (5.70) follow directly from (5.68) as $\sigma^2 \overset{k\to\infty}{\to} 0$ under the assumption of stationary conditions as $k \to \infty$.

If we only know Assumptions 5.2(a)(c), and T_m^∞ is convex combination-invariant, we notice that $\mathbb{E}[\cdot]$ is a convex combination and have:

$$\mathbb{E}\left[x^{k+1}\right] = \mathbb{E}\left[T_m^k(x^k)\right] \overset{\text{Def. } 5.5}{=} \mathbb{E}\left[T_m^k(x^k)\right] \overset{k\to\infty}{\to} \mathbb{E}\left[T_m^\infty(x^k)\right] \overset{\text{Def. } 5.4}{=} T_m^\infty(\mathbb{E}\left[x^k\right]), \tag{5.78}$$

which means as $k \to \infty$, the $\mathbb{E}\left[x^{k+1}\right] \leftarrow \mathbb{E}\left[x^k\right]$ performs the fixed-point iteration under T_m^∞. Since T_m^∞ is contractive, $\mathbb{E}\left[x^k\right]$ converges to $\mathbf{Fix}(T_m^\infty)$ (See [10], [62], [63]). This concludes our proof. □

Lemma 5.14. *For a composite stochastic operator*

$$T_C(x, \theta_A, \theta_B) := T_B\big(T_A(x, \theta_A), \theta_B\big) : \mathbb{R}^n \times \Omega_{\theta_A} \times \Omega_{\theta_B} \to \mathbb{R}^n,$$

where $\theta_A \in \Omega_{\theta_A}$ and $\theta_B \in \Omega_{\theta_B}$ are random variables, and $T_A(x, \theta_A)$ and $T_B(x, \theta_B)$ are stochastic operators, we have:

(a) If both conditions hold:

(i) for any $\theta_A \in \Omega_{\theta_A}$, the operator $T_A(x, \theta_A)$ is nonexpansive;

(ii) $T_{m,B}(x) = \mathbb{E}_{\theta_B|x}[T_B(x, \theta_B)]$ is nonexpansive,

then $T_C(x, \theta_A, \theta_B)$ is a stochastic nonexpansive operator; that is, $T_{m,C}(x) = \mathbb{E}_{\theta_A,\theta_B|x}[T_C(x, \theta_A, \theta_B)]$ is nonexpansive.

(b) In addition to (a), if either

(i) for any $\theta_A \in \Omega_{\theta_A}$, the operator $T_A(x, \theta_A)$ is averaged (or contractive);

(ii) $T_{m,B}(x)$ is averaged (or contractive),

then $T_C(x, \theta_A, \theta_B)$ is a stochastic averaged (respectively, contractive) operator; that is, $T_{m,C}(x) = \mathbb{E}_{\theta_A,\theta_B|x}[T_C(x, \theta_A, \theta_B)]$ is averaged (respectively, contractive).

Proof. To show this, note that

$$T_C(x,\theta_A,\theta_B) = T_B\big(T_A(x,\theta_A),\theta_B\big) \overset{(5.28)}{=} T_{m,B}\big(T_A(x,\theta_A)\big) + d_B(x,\theta_B) \overset{(\star)}{=} T_{m,C}(x) + d_{BA}(x,\theta_A) + d_B(x,\theta_B), \tag{5.79}$$

where, in $(\star)$,

$$T_{m,C}(x) = \mathbb{E}_{\theta_A|x}\left[T_{m,B}\big(T_A(x,\theta_A)\big)\right].$$

Part (a). By the assumptions and the properties of composite nonexpansive operators [11], together with Lemma 5.11, the operator $T_{m,C}(x)$ is nonexpansive. Moreover,

$$\mathbb{E}_{\theta_A,\theta_B|x}\left[d_{BA}(x,\theta_A) + d_B(x,\theta_B)\right] = \mathbb{E}_{\theta_A|x}\left[d_{BA}(x,\theta_A)\right] + \mathbb{E}_{\theta_B|x}\left[d_B(x,\theta_B)\right] = 0, \tag{5.80}$$

implying $\mathbb{E}_{\theta_A,\theta_B|x}\left[T_C(x,\theta_A,\theta_B)\right] = T_{m,C}(x)$. Hence, $T_C(x,\theta_A,\theta_B)$ is a stochastic nonexpansive operator, which proves (a).

Part (b). If either $T_A(x,\theta_A)$ or $T_{m,B}(x)$ is averaged (or contractive), then by the properties of composite nonexpansive operators [10], [11], the operator $T_{m,C}(x) + d_{BA}(x,\theta_A)$ is averaged (or contractive). Applying Lemma 5.11 to the convex combination $\mathbb{E}_{\theta_A|x}\left[T_{m,B}\big(T_A(x,\theta_A)\big)\right]$, we conclude that $T_{m,C}(x)$ is also averaged (or contractive). Therefore, $T_C(x,\theta_A,\theta_B)$ is a stochastic averaged (respectively, contractive) operator, proving (b). □

Theorem 5.3. *For a composite stochastic operator*

$$T_C(x,\theta_A,\theta_B) := T_B\big(T_A(x,\theta_A),\theta_B\big) : \mathbb{R}^n \times \Omega_{\theta_A} \times \Omega_{\theta_B} \to \mathbb{R}^n,$$

we have:

(a) Suppose both:

(i) for any $\theta_A \in \Omega_{\theta_A}$, the operator $T_A(x,\theta_A)$ is nonexpansive;

(ii) the operator $T_{m,B}(x) = \mathbb{E}_{\theta_B|x}\left[T_B(x,\theta_B)\right]$ is nonexpansive,

then the following inequality holds:

$$(1+\beta_{m,B})\,\left\|\big(T_{m,C}-T_{m,B}\,T_{m,A}\big)(x)\right\|^2\;+\;\beta_{m,B}\,\phi(x)\;+\;v_{d,BA}(x)\;\leq\;v_{d,A}(x), \tag{5.81}$$

where $\phi(x)\;:=\;\mathbb{E}_{\theta_A|x}\left[\left\|\big(d_{BA}-d_A\big)(x,\theta_A)\right\|^2\right]$, $v_{d,BA}(x)\;=\;\mathbb{E}_{\theta_A|x}\left[\|d_{BA}(x,\theta_A)\|^2\right]$, *and* $v_{d,A}(x)\;=\;\mathbb{E}_{\theta_A|x}\left[\|d_A(x,\theta_A)\|^2\right]$ *are defined according to* (5.32), *and* $\beta_{m,B}\geq 0$ *is the coefficient of averagedness from* (5.5).

(b) In addition to (a), if

$$v_{d,A}(x)\;=\;\mathbb{E}_{\theta_A|x}\left[\|d_A(x,\theta_A)\|^2\right]\;\overset{k\to\infty}{\to}\;0,$$

then we have

$$\left\|\big(T_{m,C}(x)-T_{m,B}\,T_{m,A}\big)(x)\right\|^2\;\overset{k\to\infty}{\to}\;0, \tag{5.82}$$

$$v_{d,BA}(x)\;=\;\mathbb{E}_{\theta_A|x}\left[\|d_{BA}(x,\theta_A)\|^2\right]\;\overset{k\to\infty}{\to}\;0. \tag{5.83}$$

Proof of Theorem 5.3. Consider the composite stochastic operator

$$T_C(x,\theta_A,\theta_B)\;:=\;T_B\big(T_A(x,\theta_A),\theta_B\big)\colon\mathbb{R}^n\times\Omega_{\theta_A}\times\Omega_{\theta_B}\;\to\;\mathbb{R}^n.$$

If (i) for any $\theta_A\in\Omega_{\theta_A}$, the operator $T_A(x,\theta_A)$ is nonexpansive, and (ii) the operator $T_{m,B}(x)=\mathbb{E}_{\theta_B|x}\left[T_B(x,\theta_B)\right]$ is nonexpansive, we let $x_A=T_A(x,\theta_A)$ and $x_B=T_{m,A}(x)$. By applying (5.5) to $T_{m,B}$, we have

$$\begin{aligned}
&\beta_{m,B}\,\|T_{m,B}(T_A(x,\theta_A))-T_{m,B}(T_{m,A}(x))-(T_A(x,\theta_A)-T_{m,A}(x))\|^2 \qquad (5.84)\\
&\quad+\|T_{m,B}(T_A(x,\theta_A))-T_{m,B}(T_{m,A}(x))\|^2\leq\|T_A(x,\theta_A)-T_{m,A}(x)\|^2\\
&\overset{(5.79)}{\Rightarrow}\beta_{m,B}\,\|(T_{m,C}-T_{m,B}T_{m,A})(x)+(d_{BA}-d_A)(x,\theta_A)\|^2\\
&\quad+\|(T_{m,C}-T_{m,B}T_{m,A})(x)+d_{BA}(x,\theta_A)\|^2\leq\|d_A(x,\theta_A)\|^2\\
&\overset{(5.33)}{\Rightarrow}(1+\beta_{m,B})\,\|(T_{m,C}-T_{m,B}T_{m,A})(x)\|^2+\beta_{m,B}\mathbb{E}_{\theta_A|x}\left[\|(d_{BA}-d_A)(x,\theta_A)\|^2\right]\\
&\quad+v_{d,BA}(x)\leq v_{d,A}(x), \qquad (5.85)
\end{aligned}$$

which proves (a). Finally, by applying this inequality under the condition $v_{d,A}(x)\overset{k\to\infty}{\to}0$, we obtain (b). □

5.3.1 Recursive fixed-point method

In practice, when designing a decision making system (e.g. a feedback optimal controller), one must solve an online optimisation problem with uncertain parameters for the control inputs.

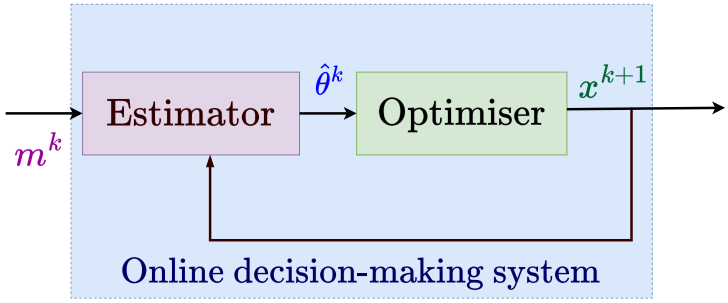


Figure 5.1: Feedback controller as an online decision-making system

As illustrated in Figure 5.1, the estimator recursively computes the uncertain parameter $\hat{\theta}^k$ required by the optimiser, based on plant measurements m^k and other inbound signals. With an MMSE (Minimum Mean Square Error) estimator introduced in Chapter 2, we have

$$\hat{\theta}^k \;=\; \mathbb{E}_{\theta^k|m^k}\left[\theta^k\right]$$

(i.e., the Bayesian posterior mean), and it typically takes time for $\mathbf{Var}\left[\theta^k\right]$ to decrease to sufficiently small values. Meanwhile, the optimiser solves an online optimisation problem for the next control output x^{k+1}, which is required by the estimator, the plant, and external subsystems. However, due to limited computational capabilities, the optimiser can only produce a suboptimal solution.

Algorithm 5.1 Recursive fixed-point method

Input: x^0

Repeat:

1: Receive the most recent parameter estimate $\hat{\theta}^k$
2: $x^{k+1} \leftarrow T(x^k, \hat{\theta}^k)$
3: $k \leftarrow k+1$
4: Output x^{k+1}

For first-order optimisation algorithms formulated as fixed-point iterations of nonexpansive operators, and applying the stochastic nonexpansive operator theory proposed in this chapter, we propose the following recursive fixed-point method (Algorithm 5.1). In Step 2, instead of

$$x^{k+1} \leftarrow \mathbf{Fix}(T(x^k, \hat{\theta}^k)),$$

we perform only a single fixed-point iteration

$$x^{k+1} \leftarrow T(x^k, \hat{\theta}^k)$$

and then output the result. From Lemma 5.11, any convex combination of averaged (or contractive) operators remains averaged (or contractive). Consequently, the stochastic mean operator

$$\mathbb{E}_{\theta^k|x^k}\left[T(x^k, \theta^k)\right]$$

is also averaged, since, for each realisation of θ^k, the first-order step $T(x^k, \theta^k)$ is typically averaged (as it corresponds to solving a convex optimisation problem specified by that particular realisation of θ^k).

Theorems 5.1 and 5.2 establish convergence in probability when $\mathbf{Var}\left[\theta^k\right]$ converges, under the respective averaged or contractive settings. Theorem 5.3 then treats the case of composite operators, encompassing first-order algorithms with operator splitting methods. In the next section, we present a numerical study to verify whether this algorithm produces reliable results.

5.4 Numerical Study

This section presents a numerical study on the convergence of Algorithm 5.1. The following quadratic programming problem is considered:

$$\begin{aligned} \text{minimise} \quad & \frac{1}{2}\langle x, Hx\rangle + \langle c, x\rangle, \\ \text{subject to} \quad & Ax \leq b, \\ & Ex = d, \end{aligned} \tag{5.86}$$

where $x \in \mathbb{R}^n$ is the decision variable, $H \in \mathbb{R}^{n\times n} \succeq 0$, and $c \in \mathbb{R}^n$ define the quadratic objective function. The parameters $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $E \in \mathbb{R}^{q\times n}$, and $d \in \mathbb{R}^q$ represent the inequality and equality constraints. We set $n = 20$, $m = 40$, and $q = 1$. The matrix A and vector b are deterministic, defining box constraints such that $x_i \in [-10, 10]$ for all i.

The parameters c, E, and d are random variables with nominal mean values sampled as $\bar{c}, \bar{E}, \bar{d} \overset{e.w.i.}{\sim} N(0,1)$, where $X \overset{e.w.i.}{\sim} D$ indicates that each element of X is independently drawn from the distribution D (i.e., $\bar{c}_i, \bar{E}_{jk}, \bar{d}_t \sim N(0,1), \forall i,j,k,t$). The matrix H is also random, constructed to satisfy $\bar{H} = V\bar{\Lambda}V^\intercal \succeq 0$. Here, V is a deterministic orthogonal matrix, chosen as the orthonormal basis for $\mathbf{col}(B)$, where $B \overset{e.w.i.}{\sim} N(0,1)$. The diagonal matrix $\Lambda = \mathrm{diag}(\lambda)$ is defined such that $\bar{\lambda}_i = \|s_i\|$, with $s_i \sim N(0,1)$ for $i > 0.5n$, and $\lambda_i = 0$ for $i \leq 0.5n$. This eigenvalue configuration ensures the quadratic objective is not strongly convex.

To solve this problem, we estimate the random parameters $\theta = \{H, c, E, d\}$ at each iteration, obtaining $\{H^k, c^k, E^k, d^k\}$ (i.e., $\hat{\theta}^k$ in Algorithm 5.1). The projected gradient method (2.68) is adapted for Algorithm 5.1 as follows for all k:

$$x^{k+1} \leftarrow T(x^k, \theta^k) = \mathbf{proj}_{\mathcal{C}^k}\left(x^k - \alpha(H^k x^k + c^k)\right), \tag{5.87}$$

where $\mathcal{C}^k$ is the feasible set defined by $\{A^k, b^k, E^k, d^k\}$, $\alpha = 1/\max_i \bar{\lambda}_i$ is the fixed step size, and $T(x,\theta)$ is the projected gradient operator parameterised by θ.

We consider three noise modes and two options for sampling the realisations of the random parameters H, c, E, d at each iteration, described as follows for all k:

$$\{H^k, c^k, E^k, d^k\} \overset{\text{-a (add-only)}}{\leftarrow} \{\bar{H}, \bar{c}, \bar{E}, \bar{d}\} + \{V\delta^k_\Lambda V^\intercal, \delta^k_c, \delta^k_E, \delta^k_d\}, \tag{5.88a}$$

$$H^k \overset{\text{-b (add+abs)}}{\leftarrow} V|\bar{\Lambda} + \delta^k_\Lambda|V^\intercal \quad (|\cdot| \text{ takes elementwise absolute values}),$$

$$\{c^k, E^k, d^k\} \overset{\text{-b (add+abs)}}{\leftarrow} \{\bar{c}, \bar{E}, \bar{d}\} + \{\delta^k_c, \delta^k_E, \delta^k_d\}, \tag{5.88b}$$

$$\text{Mode 1}: \quad \{\delta^k_\Lambda, \delta^k_c, \delta^k_E, \delta^k_d\} = C_0 k^{-C_p/2}\{C_H\,\mathrm{diag}(s^k_\lambda), s^k_c, s^k_E, s^k_d\}, \tag{5.88c}$$

$$\text{Mode 2}: \quad \{\delta^k_\Lambda, \delta^k_c, \delta^k_E, \delta^k_d\} = C_0 e^{-k/2C_e}\{C_H\,\mathrm{diag}(s^k_\lambda), s^k_c, s^k_E, s^k_d\}, \tag{5.88d}$$

$$\text{Mode 3}: \quad \{\delta^k_\Lambda, \delta^k_c, \delta^k_E, \delta^k_d\} = C_0\{C_H\,\mathrm{diag}(s^k_\lambda), s^k_c, s^k_E, s^k_d\}, \tag{5.88e}$$

where $\{s_\lambda^k, s_c^k, s_E^k, s_d^k\} \overset{\text{e.w.i.}}{\sim} N(0,1)$, and C_H, C_0, C_p, C_e are scalar constants. The constant C_H scales δ_Λ^k, as the problem is sensitive to the quadratic term of the objective. The option "-b (add+abs)" ensures $H^k \succeq 0$ in most cases, whereas "-a (add-only)" allows H^k to be indefinite but maintains $\mathbb{E}\left[H^k\right] \succeq 0$. However, the "-b (add+abs)" option slightly shifts $\mathbb{E}\left[H\right]$ away from $\bar{H}$ due to the adjustment $|\bar{\Lambda} + \delta_\Lambda^k|$. Mode 3 produces random noise with constant variance, while Modes 2 and 1 generate diminishing noise with polynomial or exponential decay rates, respectively. We do not verify the feasibility of $\mathcal{C}^k$, as it is generally guaranteed.

Simulation results are benchmarked against the optimal solution $x^* \in \mathbf{Fix}(T(x, \bar{\theta}))$, obtained by solving the deterministic optimisation problem with $\bar{\theta} = \{\bar{H}, \bar{c}, \bar{E}, \bar{d}\}$. The optimal objective value is $f^* = f(x^*) = \frac{1}{2}\langle x^*, \bar{H}x^*\rangle + \langle \bar{c}, x^*\rangle$. Two types of normalised error functions are defined:

$$e_f(x) := \frac{\|f(x) - f(x^*)\|^2}{\|f(x^*)\|^2}, \quad e_x(x) := \frac{\|x - x^*\|^2}{\|x^*\|^2}. \tag{5.89}$$

In addition to the iterates $\{x^k\}$ generated by (5.87), we compute $\{\tilde{x}^k \mid \tilde{x}^k \in \mathbf{Fix}(T^k)\}$, where $T^k = T(x, \theta^k)$. The resulting error metrics are denoted as $e_{f,\text{iter}}^k$, $e_{x,\text{iter}}^k$, $e_{f,\mathbf{Fix}(T^k)}^k$, and $e_{x,\mathbf{Fix}(T^k)}^k$, respectively.

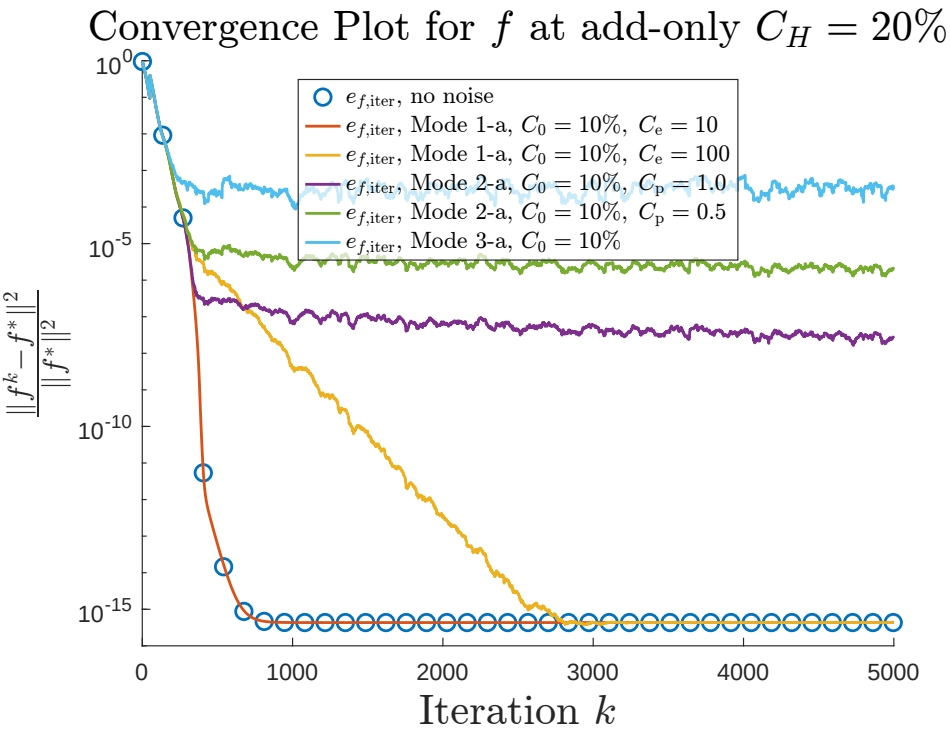
5.4.1 Discussion of results

Figures 5.2 and 5.3 present the simulation results under different noise modes, with fixed parameters $\{C_0, C_H\} = \{0.1, 0.2\}$. In all cases, some degree of convergence is observed. However, in Figure 5.2, the iterating problem is nonconvex[1] due to the indefinite nature of H^k when additive-only noise is applied to the zero eigenvalues of $\mathbb{E}\left[H^k\right]$.

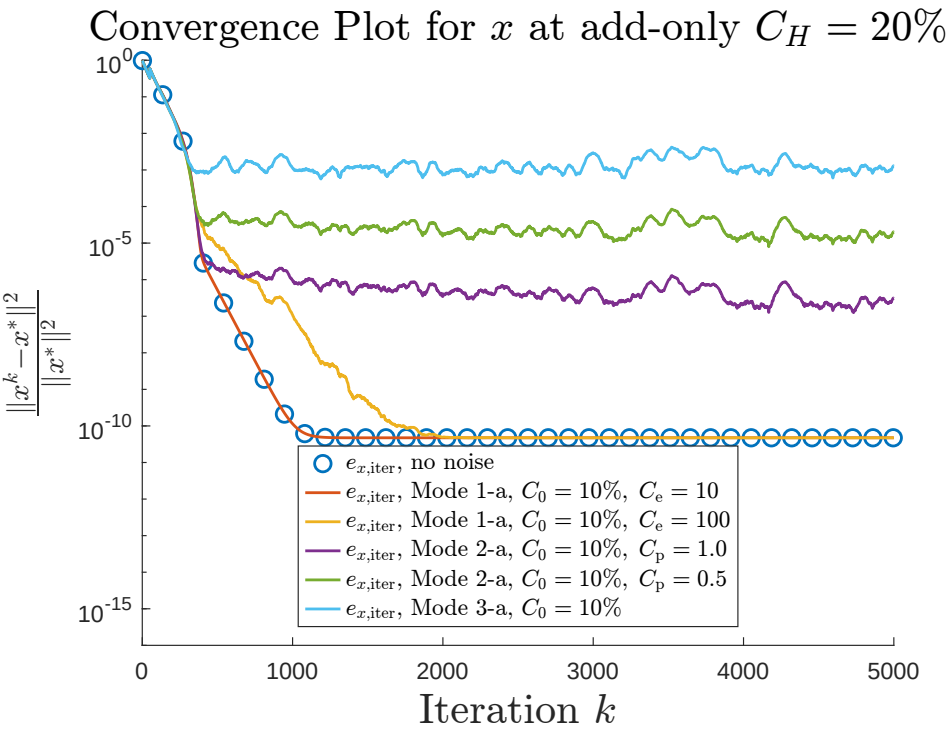
This behaviour can be explained by the proposed framework, as the iterating gradient-projection operator

$$T_{GP}(z, \theta) = \mathbf{proj}_{\mathcal{C}^{k-1}}(z) - \alpha(H^k(\mathbf{proj}_{\mathcal{C}^{k-1}}(z)) + \mathcal{C}^k) \tag{5.90}$$

[1]For this reason, $e_{f,\mathbf{Fix}(T^k)}^k$ and $e_{x,\mathbf{Fix}(T^k)}^k$ are not calculated in Figure 5.2.

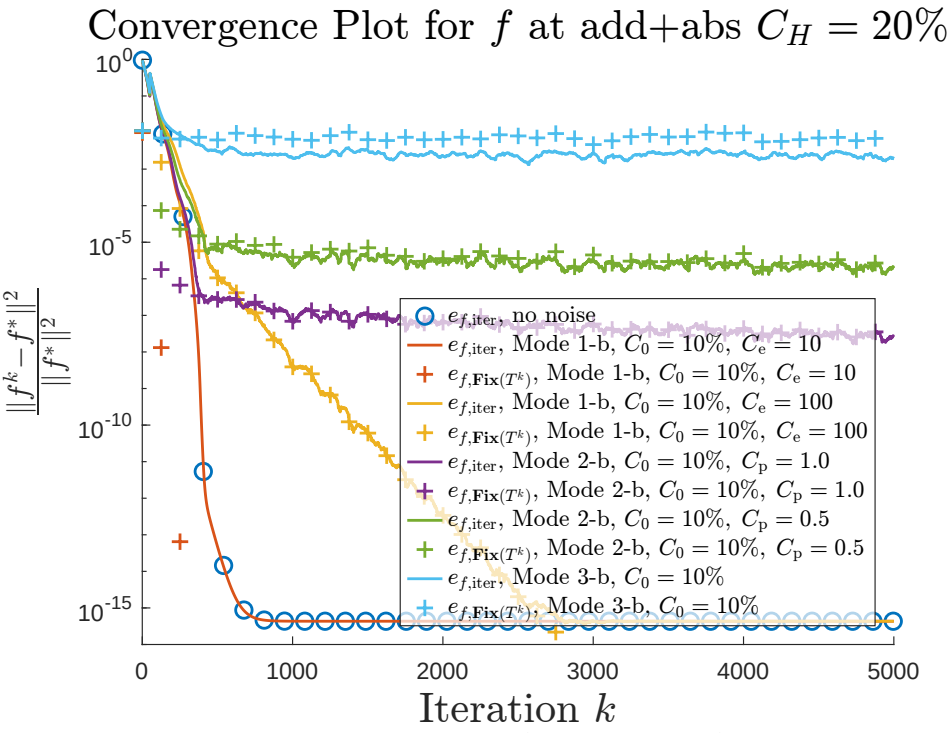


(a) Convergence plot for $e^k_{f,\text{iter}}$.

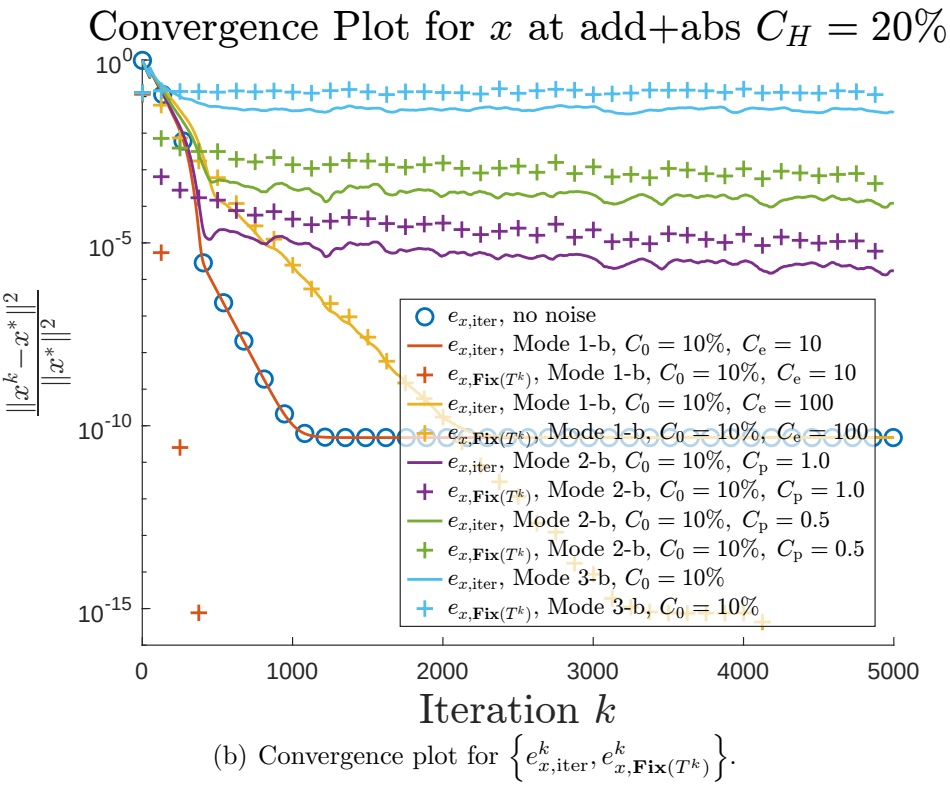


(b) Convergence plot for $e^k_{x,\text{iter}}$.

Figure 5.2: Convergence plot for add-only $C_H = 20\%$ with different noise convergence rates.



(a) Convergence plot for $\left\{e^k_{f,\text{iter}}, e^k_{f,\mathbf{Fix}(T^k)}\right\}$.



(b) Convergence plot for $\left\{e^k_{x,\text{iter}}, e^k_{x,\mathbf{Fix}(T^k)}\right\}$.

Figure 5.3: Convergence plot for add+abs $C_H = 20\%$ with different noise convergence rates.

is a stochastic nonexpansive operator, as defined in Definition 5.5 (i.e., $\mathbb{E}_{\theta|z}\left[T_{GP}(z,\theta)\right]$ is nonexpansive).

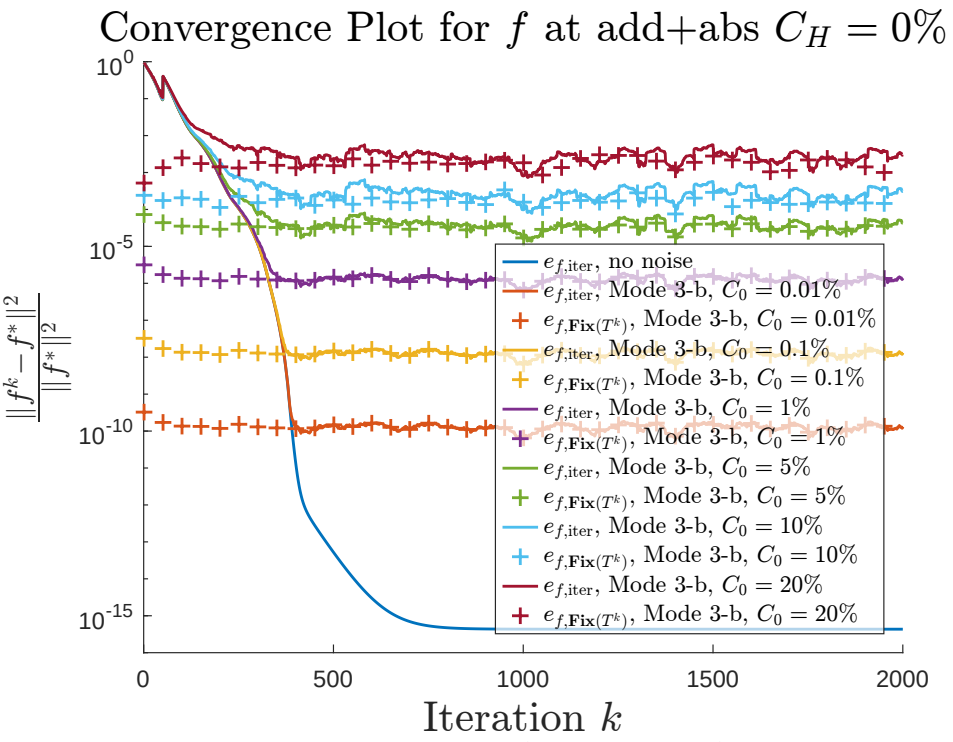
In both figures, as the noise mode transitions from Mode 3 to Mode 1 (i.e., from constant to diminishing noise with faster convergence rates), convergence is observed: to finite errors in Mode 3, and to small values in Modes 2 and 1. For fast exponentially decreasing noise (Mode 1-a, $C_e = 10$), the convergence rate of $e^k_{x,\text{iter}}$ reaches its maximum, overlapping with that of the no-noise case.

In Figure 5.3, this is particularly evident in Mode 1-b ($C_e = 10$), where it can be seen that $\left\{e^k_{f,\mathbf{Fix}(T^k)}, e^k_{x,\mathbf{Fix}(T^k)}\right\}$ rapidly converges to zero, but $\left\{e^k_{f,\text{iter}}, e^k_{x,\text{iter}}\right\}$ converges at a rate comparable to the case without noise. An intriguing observation in Figure 5.3 is that, over time, the plots of $\left\{e^k_{f,\text{iter}}, e^k_{x,\text{iter}}\right\}$ fall below those of $\left\{e^k_{f,\mathbf{Fix}(T^k)}, e^k_{x,\mathbf{Fix}(T^k)}\right\}$ for cases with slow or non-diminishing noise (Modes 2-b and 3-b). This implies that the iterates x^k generated by (5.87) outperform $\tilde{x}^k \in \mathbf{Fix}(T(x,\theta^k))$, which represents the optimal solution to the optimisation problem using the most recent realisations of the uncertain parameters.

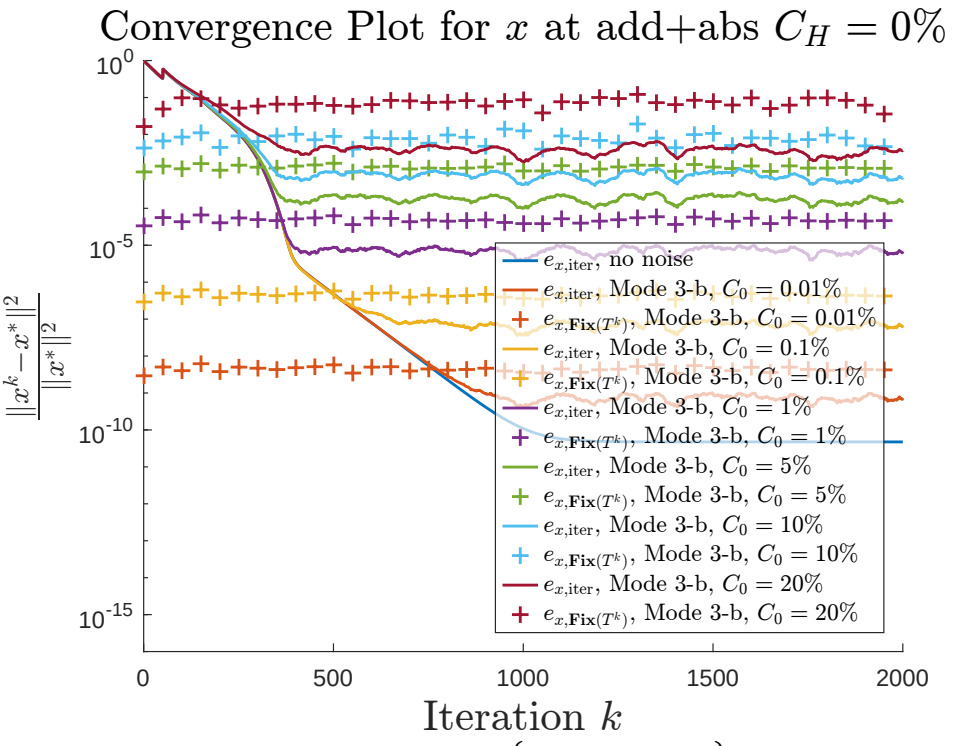
This phenomenon of "advantage over perfectionism" can be explained by the observation that $\tilde{x}^k$ consistently overshoots the time-varying $\mathbf{Fix}(T^k)$, whereas the fixed-point iterate x^k benefits from: (i) the nonexpansiveness of T^k, yielding more stable iterates, and (ii) avoiding overshoots by iterating only one step to approach $\mathbf{Fix}(T^k)$. To investigate this, we perform additional simulations to compare $\left\{e^k_{f,\text{iter}}, e^k_{x,\text{iter}}\right\}$ and $\left\{e^k_{f,\mathbf{Fix}(T^k)}, e^k_{x,\mathbf{Fix}(T^k)}\right\}$ for different values of C_0 and C_H under noise Mode 3-b. The results are presented in Figures 5.4, 5.5, and 5.6.

The aforementioned phenomenon is more pronounced in Figures 5.4(b), 5.5(b), and 5.6(b) compared to Figures 5.4(a), 5.5(a), and 5.6(a). This can be attributed to x^k, in its full dimensionality compared to $f(x^k)$, leveraging the advantages of the nonexpansiveness of T^k.

When comparing the plots with increasing C_H (i.e., from Figure 5.4 to 5.5 and 5.6), both $\left\{e^k_{f,\text{iter}}, e^k_{x,\text{iter}}\right\}$ and $\left\{e^k_{f,\mathbf{Fix}(T^k)}, e^k_{x,\mathbf{Fix}(T^k)}\right\}$ demonstrate sensitivity to noise in the quadratic objective, as discussed earlier when defining C_H. However, Figure 5.6

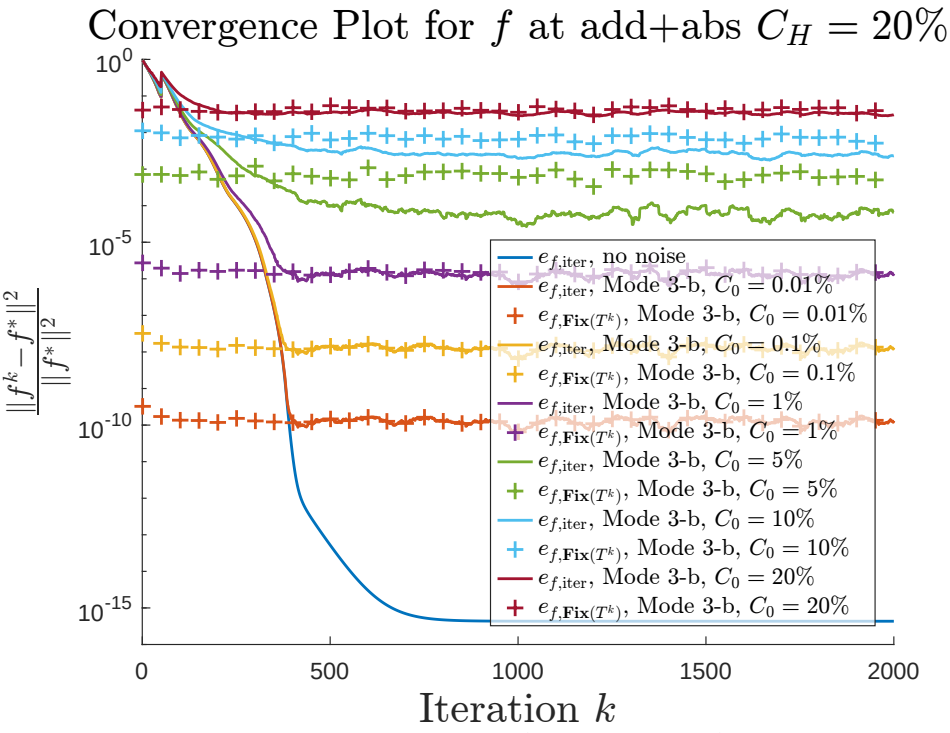


(a) Convergence plot for $\left\{e^k_{f,\text{iter}}, e^k_{f,\mathbf{Fix}(T^k)}\right\}$.

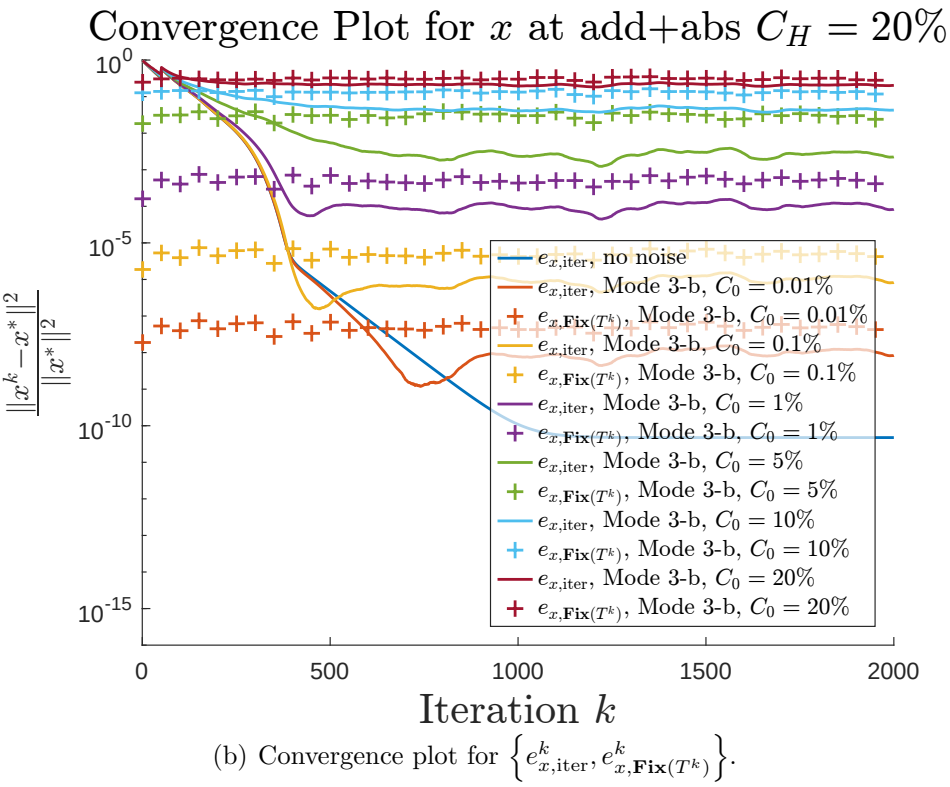


(b) Convergence plot for $\left\{e^k_{x,\text{iter}}, e^k_{x,\mathbf{Fix}(T^k)}\right\}$.

Figure 5.4: Convergence plot for add+abs $C_H = 0\%$ with different constant noise levels.

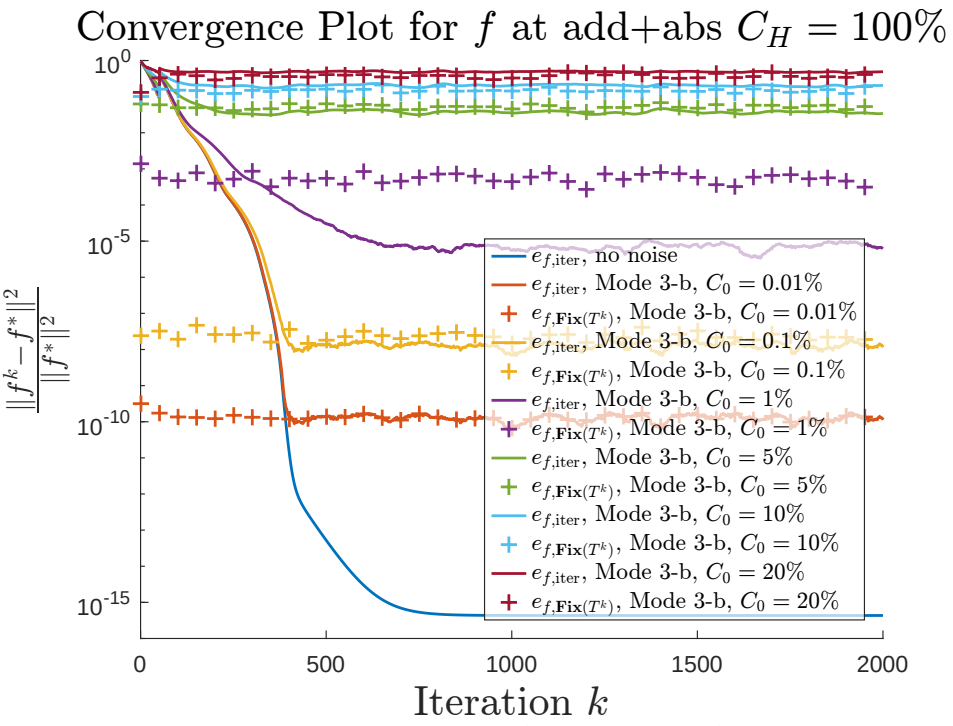


(a) Convergence plot for $\left\{e^k_{f,\text{iter}}, e^k_{f,\mathbf{Fix}(T^k)}\right\}$.

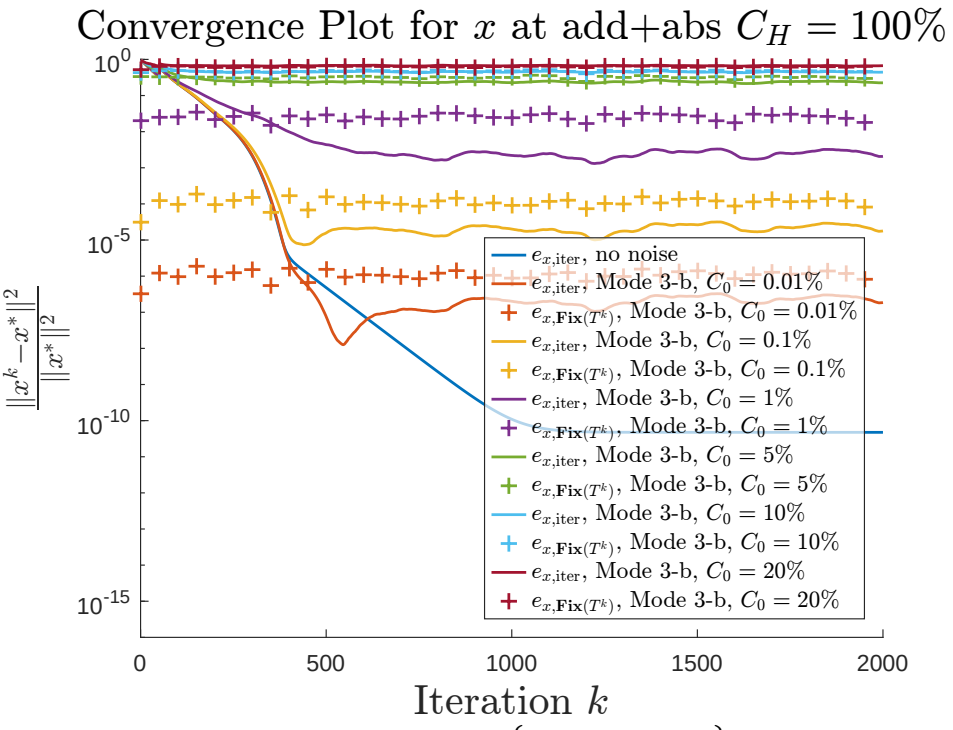


(b) Convergence plot for $\left\{e^k_{x,\text{iter}}, e^k_{x,\mathbf{Fix}(T^k)}\right\}$.

Figure 5.5: Convergence plot for add+abs $C_H = 20\%$ with different constant noise levels.



(a) Convergence plot for $\left\{e^k_{f,\text{iter}}, e^k_{f,\mathbf{Fix}(T^k)}\right\}$.



(b) Convergence plot for $\left\{e^k_{x,\text{iter}}, e^k_{x,\mathbf{Fix}(T^k)}\right\}$.

Figure 5.6: Convergence plot for add+abs $C_H = 100\%$ with different constant noise levels.

reveals that the proposed algorithm further exploits the aforementioned advantage for larger values of C_H.

To investigate the sensitivity of the proposed algorithm to different noise levels, we define the following fixed-point iteration:

$$x_m^{k+1} \leftarrow \hat{T_m^{\infty}}(x_m^k), \tag{5.91}$$

where $\hat{T_m^{\infty}}$ is the estimated $\mathbb{E}_{\theta^k|x}\left[T(x,\theta^k)\right]$ (the mean operator defined in (5.28)) as $k \to \infty$. For constant noise (Mode 3-a), the random variables $\{\theta^k\}$ are i.i.d., implying $T_m^k = T_m^{\infty} = T_m$ for all k. Thus, we estimate $\hat{T_m^{\infty}}$ using the ensemble sample mean:

$$\hat{T_m^{\infty}}(x) = \frac{1}{N_s}\sum_{i=1}^{N_s} T(x,\theta[i]), \tag{5.92}$$

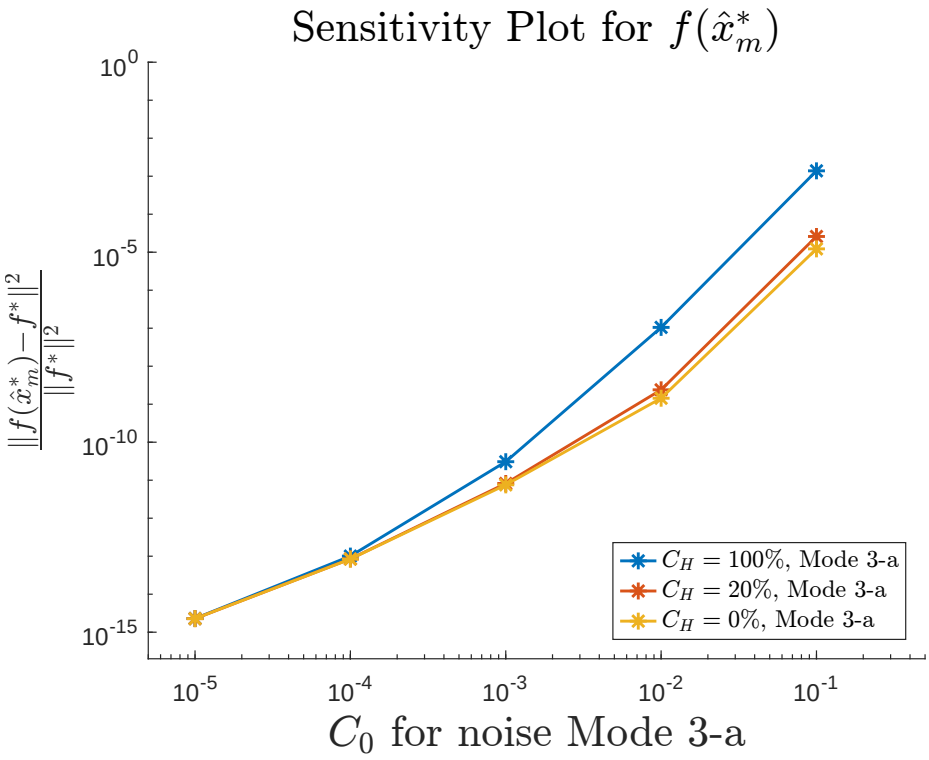
where $\{\theta[i]\}_{i=1}^{N_s}$ are N_s independent random samples of θ^k. From Theorems 5.1 and 5.2, it follows that x^k is distributed around $\mathbf{Fix}(T_m^{\infty})$, and at stationarity, $\mathbf{dist}^2\left(\mathbb{E}\left[x^k\right], \mathbf{Fix}(T_m^{\infty})\right)$ is bounded.

We perform the iteration (5.91) until convergence to ensure x_m^k converges to $\hat{x}_m^*$, providing an estimate of $x_m^* \in \mathbf{Fix}(T_m^{\infty})$. Empirically, convergence is achieved when the residual $\left\|x_m^{k+1} - x_m^k\right\|^2 / \left\|x_m^k\right\|^2$ falls below $0.1\% \left\|x_m^k - x^*\right\|^2 / \left\|x^*\right\|^2$, requiring fewer than 1000 steps of (5.91) even with $C_0 = 10\%$. The iteration starts from $x_m^0 = x^* \in \mathbf{Fix}(T(x,\bar{\theta}))$, with $N_s = 1000$ independently drawn samples at each k.

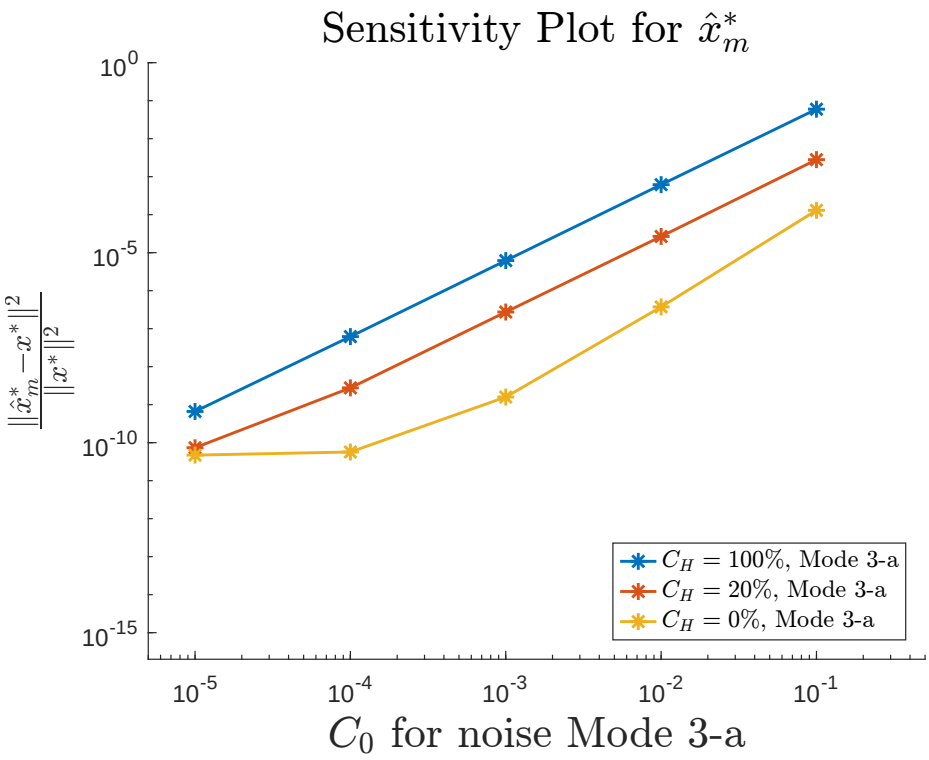
In Figure 5.7, we plot the errors of the estimates $\{\hat{x}_m^*\}$ under different constant noise levels of Mode 3-a on a log-log scale and observe a strongly monotonic relationship. By comparing Figure 5.7(b) with Figure 5.7(a), we observe that $\hat{x}_m^*$ is more sensitive to C_H than $f(\hat{x}_m^*)$.

This observation can be attributed to the previously mentioned advantage of the proposed algorithm: by updating only one step per iteration of T^k, it leverages the nonexpansiveness of T^k while avoiding overshooting $\mathbf{Fix}(T^k)$.

We recognise the open question of whether $x_m^* \in \mathbf{Fix}(\mathbb{E}_{\theta|x}\left[T(x,\theta)\right])$ is a viable solution, potentially surpassing the benchmark $x^* \in \mathbf{Fix}(T(x,\mathbb{E}\left[\theta\right]))$. In practice, our algorithm conserves computational effort by performing only a single iteration



(a) Sensitivity plot for $e^k_{f(\hat{x}^*_m)}$.



(b) Sensitivity plot for $e^k_{\hat{x}^*_m}$.

Figure 5.7: Sensitivity plot for $\hat{x}^*_m$ with different add-only C_H and constant noise levels.

per T^k, thus avoiding any waiting period for parameter noise convergence. At the same time, it capitalises on the most recent realisations of the uncertain parameters and leverages the iterative stability provided by the nonexpansiveness of T^k. These observations suggest that $x_m^* \in \mathbf{Fix}(\mathbb{E}_{\theta|x}[T(x,\theta)])$ is indeed a viable solution.

5.4.2 Summary of numerical study

This numerical study examines the performance of the proposed online algorithm in solving a random quadratic programming (QP) problem under various noise modes. We consider random parameters $\theta = \{H, c, E, d\}$ and assess the algorithm's iterates $\{x^k\}$ alongside the fixed-point solutions $\{\tilde{x}^k\}$, where each $\tilde{x}^k$ solves the optimisation problem with the most recent realisations of the uncertain parameters.

Our results indicate that, even with indefinite matrices H^k (Mode "-a"), the algorithm achieves notable convergence. In certain scenarios, x^k outperforms $\tilde{x}^k$, highlighting an "advantage over perfectionism." This occurs because updating x^k by one step leverages the nonexpansiveness of $T(x,\theta)$ while avoiding overshoots toward $\mathbf{Fix}(T(x,\theta))$. Additional simulations demonstrate the algorithm's robustness under diminishing noise (Modes 1 and 2) and constant noise (Mode 3). We further estimate the mean operator T_m^∞ by averaging multiple samples of θ (5.92) and observe bounded convergence around $\mathbf{Fix}(T_m^\infty)$. Moreover, increasing the scaling factor C_H amplifies the algorithm's sensitivity to noise in the quadratic objective. However, a larger C_H may also enhance the previously mentioned advantage, allowing x^k to remain stable relative to $\tilde{x}^k$. Overall, the numerical findings confirm the effectiveness and robustness of the proposed approach in the presence of uncertain parameters and varying noise levels.

5.5 Conclusion

This chapter introduces a framework for analysing convergence and robustness in averaged operators under stochastic uncertainty. We describe the novel concept of the stochastic mean operator to characterise stable fixed-points of uncertain operators

parametrised by estimated parameters. The proposed recursive fixed-point method reduces computational demands by replacing full optimisation steps with a single step per estimated parameter while maintaining convergence guarantees. Numerical studies demonstrate the algorithm's robustness across various noise conditions and highlighted the "advantage over perfectionism", where the proposed method under limited computational resources outperforms time-varying optimal solutions. These results validate the framework's effectiveness and practical relevance for optimisation problems with uncertain parameters.

Future research will focus on applying this framework to specific problems and solvers, for which tighter bounds on uncertainty are available. A promising direction to investigate is whether, in problems that do not satisfy the combination-invariance condition proposed in this chapter, the error relative to this condition can be used characterise the mean values of iterates. This would allow more practical sample mean-based methods to be applied as proper estimations of the stochastic result.

Chapter 6

Conclusion and Outlook

This thesis provides a deep investigation into the first order optimisation algorithms under the uncertainties induced by asynchrony (Chapter 3) and parameter estimations (Chapters 4 and 5). In Chapter 5 we propose a general framework of stochastic operators and consequentially the recursive fixed-point method which solves online optimisation problems with uncertainty.

Asynchronous ADMM via a Data Exchange Server

In this work, a decentralised asynchronous communication and update protocol is introduced, leveraging the ADMM to address a convex optimisation problem, which involves two sets of local cost functions and constraints with local coupling consensus. The algorithm proposed in this study has some features in common with [15, Algorithm 4], which employs a centralized aggregator to oversee data exchange and to perform parts of the primal and dual variable updates. However, notable distinctions are present: in our formulation we introduce a data server operating on independent clock cycles to manage asynchronous data exchanges among agents—this is purely a data handling role with no computational involvement. Additionally, our approach employs local consensus blocks in lieu of a singular common consensus and utilises a vectorised augmentation parameter as opposed to a scalar one.

The numerical study of this work shows the convergence of the proposed algorithm with varying asynchrony parameters, and suggests that the theoretical bounds are somewhat conservative. Hence future work will focus on tightening the sufficient

conditions of our approach. A comparison with [15] is also provided which emphasises the key bipartite structure of the proposed algorithm.

Optimisation with Parametric Uncertainty

In this work we exploit the inherent robustness of solver iterations to devise an ADMM algorithm with parametric uncertainty (ADMM-PU) for the following distributed optimisation problem with uncertainty:

$$\min_{x,z} \quad f^k(x) + g(z), \quad \text{subject to} \quad x - z = 0,$$

where

$$f^k(x) = \bar{f}(x) + \mathcal{I}_{\mathcal{F}_f^k}(x) = \frac{1}{2}x^\intercal Q x + c^\intercal x + \mathcal{I}_{\mathcal{F}_f^k}(x), \quad \mathcal{F}_f^k : \; Ax \leq b + Dp^k,$$

in which $f^k, g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are convex, closed, proper functions. Let $g(z) = \hat{g}(z) + \mathcal{I}_{\mathcal{F}g}(z)$. $\hat{g} : \mathbb{R}^n \to \mathbb{R}$ is convex and continuous, $\mathcal{F}_f^k, \mathcal{F}_g \subset \mathbb{R}^n$ are convex constraint sets, and $\mathcal{I}_{\mathcal{C}}(x)$ is the indicator function of a set $\mathcal{C}$.

At each iteration, our approach uses the most recent estimate p^k to replace the unknown parameter $\bar{p}$ by replacing the uncertain function f with its corresponding estimate f^k. Our approach is applicable more generally to first order methods that can be expressed as DRS methods. If the solver iteration can be expressed in the form of a DRS operator, then, under two groups of assumptions, we show the solution estimate necessarily converges to the solution of the reference deterministic problem with $p^k = \bar{p}$. The first set of assumptions assumes bounded feasible sets for the time-varying part (f^k) of the problem, a l_1-norm stable estimator driven by a finite l_1-norm noise sequence, and no overlap between the normal cones of the constraint sets as $p^k = \bar{p}$. The second set of assumptions assumes a compact overall feasible set, an estimator that guarantees $O(1/k)$ variance convergence, and no overlap between the normal cones of the constraint sets for all p^k. Apart from the two main convergence theorems, this chapter provides several additional subsidiary results: firstly, the above-mentioned problem can be formulated as a DRS applied to

the reference time-invariant problem with a bounded piecewise affine residual; secondly, the work provides conditions that guarantee the boundedness of the set of the optimal dual variables; thirdly the work points out and investigates the convergence of the time-varying nonexpansive operator and thus motivates the following chapter, which generalises the result with a framework of stochastic nonexpansive operators.

Online optimisation with the Recursive Fixed-point method: A framework of stochastic nonexpansive operators

This chapter develops a general framework of stochastic nonexpansive operators to analyse convergence and robustness in optimisation algorithms involving parametric uncertainty. By interpreting the iteration as an averaged nonexpansive operator, we introduce the concept of the stochastic mean nonexpansive operator and the resulting fixed-point set, which represents the overall stable point. This framework accommodates non-i.i.d. and finite variance parameter noise, providing convergence bounds for the first and second moments of the iterates.

A consequent recursive fixed-point method is proposed, where a single iteration of the parameter estimator replaces the multiple steps that would be required for estimator to converge to an accurate estimate. As a result we consider solving the optimisation problem parametrised by time-varying parameter estimates, thus significantly reducing computational effort. The stochastic mean operator is shown to remain averaged, ensuring convergence in probability if the parameter variance $\mathbf{Var}\left[\theta^k\right]$ vanishes asymptotically, and allowing the derivation of Markov bounds when $\mathbf{Var}\left[\theta^k\right]$ converges to finite bounds. Composite operators, such as those arising in first-order optimisation methods with splitting methods, are also addressed, further extending the framework's applicability.

The numerical study evaluates the algorithm's performance on random quadratic programming problems under varying noise conditions. Results demonstrate that the proposed method achieves convergence and robustness across different noise modes. Notably, an "advantage over perfectionism" effect is observed, where the algorithm

outperforms time-varying optimal solutions by leveraging nonexpansiveness and avoiding overshoots. Simulations confirm stability and effectiveness even with temporary non-convexity and non-diminishing noise, validating the framework's practical relevance in noisy optimisation settings. A Monte-Carlo sampling method is used in order to verify the existence of the fixed-point set of the stochastic mean operator.

6.1 Future research directions

Model the asynchrony with the proposed stochastic framework

In Chapter 3 we use a deterministic approach to bound the uncertainty of the asynchrony by the strong convexity of part of the cost functions. In [17], [175]–[177], asynchrony-induced uncertainty is investigated in the context of the distributed optimisation problem $\min \sum_i f_i(x)$ with randomised Mann iteration [17], [177] applied to ADMM [17], a primal-dual method [175] or to projection gradient method [176] which is as follows:

$$z_i^k \leftarrow x^k - \alpha \nabla f_i(x^k) \tag{6.1a}$$

$$x^{k+1} \leftarrow \frac{1}{m} z_i^k, \tag{6.1b}$$

which is a special form of the projected gradient method with projection onto the affine constraint set $x_i = x_j, \forall i, j$. In [17], [175]–[177], convergence is proved via the scaled norm $|||\cdot|||$ introduced in [17], [177], when z_i^k is missing and replaced by z_i^{k-1} due to probabilistic asynchrony. To apply the concept of stochastic mean operator proposed in Chapter 5, we can explore the case where constraints other than $x_i = x_j, \forall i, j$ are included for problems with constrained objectives. On the other hand, instead of using z_i^{k-1} as the replacement for a missing update z_i^k, we can use an estimate of $\nabla f_i(x^k)$ to replace the gradient step (i.e. $z_i^k \leftarrow x^k - \alpha \hat{\nabla} f_i(x^k)$) and provide convergence bounds.

Apart from the projected gradient method, which is a forward-backward splitting (Example 2.8), we may also investigate the convergence under both the asynchrony and parametric uncertainty of other splitting methods such as the Douglas-Rachford

splitting and the ADMM algorithm introduced in Chapters 3 and 4 to solve problems with non-smooth objectives.

Tighten the bound for the convergence of the mean value

We proposed the concept of convex combination-invariance (CCI, see Definition 5.4) in Chapter 5, focusing on the specific stochastic expectation invariance for fixed-point iterations by emphasising the example of projection onto a convex set operator which is not affine but convex combination-invariant (CCI) given that the input is already inside the convex set.

From the numerical study of Chapter 5 we observe that, under stationary conditions, the offset $\left\|\frac{1}{K}\sum_k x^k - \hat{x}_m^*\right\|^2 / \|\hat{x}_m^*\|^2$ can be as small as 10^{-4}, where $\frac{1}{K}\sum_k x^k$ is the Cesàro mean of x^k. Hence a possible future research direction is to investigate the convex combination-invariance offset of the mean operator with the aim of tightening the bounds on $\mathbf{dist}^2\left(\mathbb{E}\left[x^k\right], \mathbf{Fix}(T_m^\infty)\right)$ under stationary conditions when $\mathbb{E}\left[x^k\right] = \mathbb{E}\left[T_m^\infty(x^k)\right]$ is guaranteed. The QP example described in this numerical study would be a good starting point for this work, since the gradient is affine and for each active constraint set the projection is also affine.

Apart from the CCI property with respect to the iterates x^k, another possible research direction is to study the CCI property with respect to the parameter θ for the stochastic operator $T(x,\theta)$. This could lead to useful bounds on $\mathbf{dist}^2\left(\mathbf{Fix}(T(x,\mathbb{E}\left[\theta\right])), \mathbf{Fix}(\mathbb{E}_{\theta|x}\left[T(x,\theta)\right])\right)$.

Provide a numerical analysis of the "advantage over perfectionism" effect

In Chapter 5, the numerical analysis reveals the "advantage over perfectionism" effect, characterised by:

$$e^k_{x,\text{iter}} < e^k_{x,\mathbf{Fix}(T^k)}, \tag{6.2}$$

where $e^k_{x,\text{iter}}$ and $e^k_{x,\mathbf{Fix}(T^k)}$ denote the error of the recursive fixed-point iterates and the error of the iterating fixed points, respectively. We therefore propose the following conjectures:

- $\mathbf{Var}\left[x^k\right] < \mathbf{Var}\left[\mathbf{Fix}(T(x,\theta^k))\right]$, suggesting that our proposed recursive fixed-point method outperforms the exact solutions to the iterating problems.
- Compared to $\mathbf{Fix}(T(x,\mathbb{E}\left[\theta^k\right]))$, the fixed point $\mathbf{Fix}(\mathbb{E}_{x|\theta}\left[T(x,\theta^k)\right])$ serves as a more appropriate benchmark for the overall coupling problem.

To investigate these conjectures, we intend to develop a theoretical analysis of this newly observed effect, potentially reshaping our understanding of coupled decision-making systems.

Discuss the convergence rate and noise gains of the closed-loop system

In Chapter 5, we analyse the case of "one estimation step followed by one optimisation step" for Algorithm 5.1. As observed in Figure 4.1 of Chapter 4, the optimiser eventually "catches up" to the convergence rate of the estimator. This motivates an investigation into the convergence rate of the coupled system and the development of a framework to optimise the step ratio between the estimator and the optimiser. Such a study also characterises the convergence behaviour of the recursive fixed-point method from the initial condition to the asymptotically stable state.

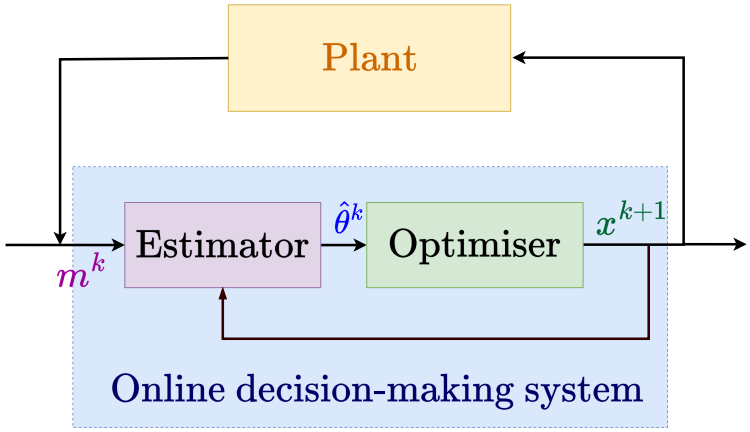


Figure 6.1: Closed-loop system.

Additionally, Chapters 4 and 5 propose robustness analyses of the noise gain from $\hat{\theta}^k$ to x^{k+1}. We are particularly interested in quantifying the noise gains of the closed-loop system. For instance, by applying the small-gain theorem, we can analyse the

loop gain of the system, provided that an appropriate model of the controlled plant is available, as illustrated in Figure 6.1.

Investigate the performance of the recursive fixed-point method applied to real-world systems

In recent years, AI-based estimation algorithms have emerged, often leveraging statistical inference through neural network models. These approaches typically produce uncertain estimates and are widely adopted in industries involving pattern recognition from visual or linguistic inputs. We plan to explore such applications and assess whether these algorithms can be enhanced via the recursive fixed-point method proposed in Chapter 5.

Recent developments in data-driven control theory [184] offer frameworks for modelling system dynamics, observers, and controllers directly from data. We aim to contribute to this growing field by integrating the recursive fixed-point method within the data-driven control paradigm.

Epilogue

During my years in Oxford, I experienced numerous life-changing moments, including the global pandemic and the unexpected virality of my name—a rare Chinese name—becoming a tag in a meme that amassed billions of views online. I am deeply grateful for the unwavering support of my family and my supervisor, Prof. Mark Cannon, especially during many dark and challenging times. I also thank my college (St Peter's), the University, the NHS, my dear friends, and countless strangers who offered their kindness and help. I am indebted to the professors whose critical feedback has honed my academic judgement to a top-tier standard.

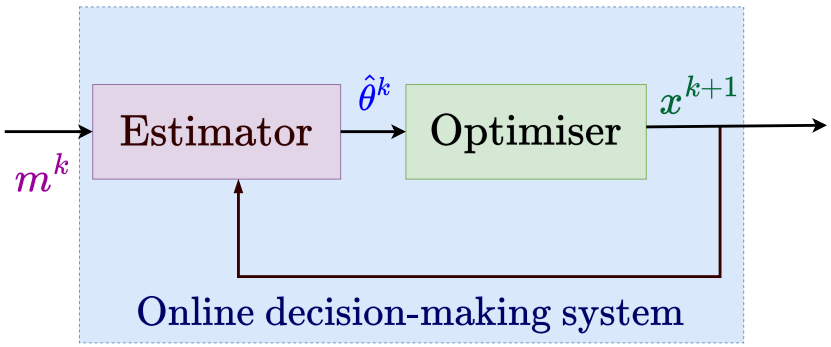


Figure 6.2: Online decision-making system.

Surviving this DPhil journey has required considerable philosophical reflection. These contemplations contributed to the development of the recursive fixed-point method illustrated in Figure 6.2. I share below the most essential principles that emerged from this process:

1. Every intelligent agent[1] can only experience its own inner stream of subjective

[1] An organic or inorganic being capable of logical reasoning.

thought[2].

2. Every intelligent agent interacts with others through input/output devices—via objective measurement and actuation[3].

3. Every intelligent agent perceives only the "present moment"[4].

This thesis bears the subtitle: *Iterate to minimise the uncertain gap between rationality and reality*—a philosophical reflection of the proposed algorithm. I believe this principle resonates beyond engineering, extending into the meaning of our lives.

I wish you all the very best, wherever and whenever you may be, as you read this final line.

[2]Analogous to the iterating variables m^k, $\hat{\theta}^k$, and x^{k+1} in Figure 6.2.

[3]Analogous to m^k and x^{k+1} in Figure 6.2.

[4]The sensations of "past" and "future" are reconstructions and predictions occurring in the present. Variables m^k, $\hat{\theta}^k$, and x^{k+1} in Figure 6.2 are updated at each iteration.

Bibliography

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010, ISSN: 19358237. DOI: 10.1561/2200000016.

[2] A. Nedić, "Convergence rate of distributed averaging dynamics and optimization in networks," *Foundations and Trends in Systems and Control*, vol. 2, no. 1, pp. 1–100, 2015, ISSN: 23256826. DOI: 10.1561/2600000004.

[3] T. Yang, X. Yi, J. Wu, *et al.*, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019, ISSN: 13675788. DOI: 10.1016/j.arcontrol.2019.05.006.

[4] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009, ISSN: 00189286. DOI: 10.1109/TAC.2008.2009515.

[5] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010, ISSN: 00189286. DOI: 10.1109/TAC.2010.2041686.

[6] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015, arXiv: 1303.2289, ISSN: 00189286. DOI: 10.1109/TAC.2014.2364096.

[7] P. Lin, W. Ren, and Y. Song, "Distributed multi-agent optimization subject to nonidentical constraints and communication delays," *Automatica*, vol. 65, pp. 120–131, Mar. 2016, Publisher: Elsevier Ltd, ISSN: 00051098. DOI: 10.1016/j.automatica.2015.11.014. [Online]. Available: http://dx.doi.org/10.1016/j.automatica.2015.11.014.

[8] K. J. Arrow, L Hurwicz, and H Uzawa, *Studies in Linear and Non-linear Programming*. Stanford University Press, 1972, Series Title: Stanford mathematical studies in the social sciences.

[9] R Glowinski and A Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires," fr, *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, vol. 9, no. R2, pp. 41–76, 1975, Publisher: Dunod Place: Paris. [Online]. Available: http://www.numdam.org/item/M2AN_1975__9_2_41_0/.

[10] E. K. Ryu and S. Boyd, "A Primer on Monotone Operator Methods Survey," *Appl. Comput. Math*, pp. 3–43, 2016.

[11] P. L. Combettes and I. Yamada, "Compositions and convex combinations of averaged nonexpansive operators," en, *Journal of Mathematical Analysis and Applications*, vol. 425, no. 1, pp. 55–70, May 2015, ISSN: 0022247X. DOI: 10.1016/j.jmaa.2014.11.044. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0022247X14010865.

[12] N. Parikh, "Proximal Algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014, ISSN: 2167-3888. DOI: 10.1561/2400000003.

[13] Q. Ho, J. Cipar, H. Cui, *et al.*, "More effective distributed ML via a stale synchronous parallel parameter server," *Advances in Neural Information Processing Systems*, no. 1, pp. 1–9, 2013, ISSN: 10495258.

[14] Z. Peng, Y. Xu, M. Yan, and W. Yin, "ARock: An Algorithmic Framework for Asynchronous Parallel Coordinate Updates," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, A2851–A2879, Jan. 2016, ISSN: 1064-8275. DOI: 10.1137/15M1024950. [Online]. Available: http://epubs.siam.org/doi/10.1137/15M1024950.

[15] T. H. Chang, M. Hong, W. C. Liao, and X. Wang, "Asynchronous Distributed ADMM for Large-Scale Optimization - Part I: Algorithm and Convergence Analysis," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3118–3130, Jun. 2016, Publisher: IEEE, ISSN: 1053587X. DOI: 10.1109/TSP.2016.2537271. [Online]. Available: http://ieeexplore.ieee.org/document/7423789/.

[16] E. Wei and A. Ozdaglar, "On the O(1/k) convergence of asynchronous distributed alternating Direction Method of Multipliers," *2013 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013 - Proceedings*, pp. 551–554, 2013, Publisher: IEEE ISBN: 9781479902484. DOI: 10.1109/GlobalSIP.2013.6736937.

[17] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," *Proceedings of the IEEE Conference on Decision and Control*, pp. 3671–3676, 2013, arXiv: 1303.2837 Publisher: IEEE ISBN: 9781467357173, ISSN: 01912216. DOI: 10.1109/CDC.2013.6760448.

[18] J. Zhou and Y. Lei, "Asynchronous Group-Based ADMM Algorithm under Efficient Communication Structure," in *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, IEEE, Dec. 2018, pp. 135–140, ISBN: 978-1-72811-141-4. DOI: 10.1109/BDCloud.2018.00032. [Online]. Available: https://ieeexplore.ieee.org/document/8672228/.

[19] M. Ma, J. Ren, G. B. Giannakis, and J. Haupt, "FAST ASYNCHRONOUS DECENTRALIZED OPTIMIZATION: ALLOWING MULTIPLE MASTERS," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, Nov. 2018, pp. 633–637, ISBN: 978-1-72811-295-4. DOI: 10.1109/GlobalSIP.2018.8646514. [Online]. Available: https://ieeexplore.ieee.org/document/8646514/.

[20] T. H. Chang, W. C. Liao, M. Hong, and X. Wang, "Asynchronous Distributed ADMM for Large-Scale Optimization - Part II: Linear Convergence Analysis and Numerical Performance," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3131–3144, Jun. 2016, arXiv: 1509.02604 Publisher: IEEE, ISSN: 1053587X. DOI: 10.1109/TSP.2016.2537261.

[21] A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini, "Tracking-ADMM for distributed constraint-coupled optimization," *Automatica*, vol. 117, p. 108 962, 2020, arXiv: 1907.10860 Publisher: Elsevier Ltd, ISSN: 00051098. DOI: 10.1016/j.automatica.2020.108962. [Online]. Available: https://doi.org/10.1016/j.automatica.2020.108962.

[22] V. Khatana and M. V. Salapaka, "D-DistADMM: A O(1/k) Distributed ADMM for Distributed Optimization in Directed Graph Topologies," en, in *2020 59th IEEE Conference on Decision and Control (CDC)*, Jeju, Korea (South): IEEE, Dec. 2020, pp. 2992–2997, ISBN: 978-1-72817-447-1. DOI: 10.1109/CDC42340.2020.9304086. [Online]. Available: https://ieeexplore.ieee.org/document/9304086/.

[23] W. Jiang and T. Charalambous, "Distributed Alternating Direction Method of Multipliers using Finite-Time Exact Ratio Consensus in Digraphs," en, in *2021 European Control Conference (ECC)*, Delft, Netherlands: IEEE, Jun. 2021, pp. 2205–2212, ISBN: 978-94-6384-236-5. DOI: 10.23919/ECC54610.2021.9654976. [Online]. Available: https://ieeexplore.ieee.org/document/9654976/.

[24] N. Bastianello, R. Carli, L. Schenato, and M. Todescato, "Asynchronous Distributed Optimization Over Lossy Networks via Relaxed ADMM: Stability and Linear Convergence," en, *IEEE Transactions on Automatic Control*, vol. 66, no. 6, pp. 2620–2635, Jun. 2021, ISSN: 0018-9286, 1558-2523, 2334-3303. DOI: 10.1109/TAC.2020.3011358. [Online]. Available: https://ieeexplore.ieee.org/document/9146334/.

[25] A. Mohammadi and A. Kargarian, "Learning-Aided Asynchronous ADMM for Optimal Power Flow," en, *IEEE Transactions on Power Systems*, vol. 37, no. 3, pp. 1671–1681, May 2022, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2021.3120260. [Online]. Available: https://ieeexplore.ieee.org/document/9573286/.

[26] Z. Pan and M. Cannon, "Asynchronous ADMM via a Data Exchange Server," *IEEE Transactions on Control of Network Systems*, pp. 1–12, 2024, ISSN: 2325-5870, 2372-2533. DOI: 10.1109/TCNS.2024.3354840. [Online]. Available: https://ieeexplore.ieee.org/document/10400939/.

[27] V. Balakrishnan, "System identification: Theory for the user (second edition)," en, *Automatica*, vol. 38, no. 2, pp. 375–378, Feb. 2002, ISSN: 00051098. DOI: 10.1016/S0005-1098(01)00214-X. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S000510980100214X.

[28] N. Rontsis, P. Goulart, and Y. Nakatsukasa, "Efficient Semidefinite Programming with Approximate ADMM," *Journal of Optimization Theory and Applications*, vol. 192, no. 1, pp. 292–320, Jan. 2022, arXiv: 1912.02767 Publisher: Springer, ISSN: 15732878. DOI: 10.1007/s10957-021-01971-3.

[29] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, Jan. 2002, ISSN: 00051098. DOI: 10.1016/S0005-1098(01)00174-1.

[30] Z. Pan and M. Cannon, "Optimisation with Parametric Uncertainty: An ADMM Approach," *IFAC World Congress 2023*, 2023.

[31] P. L. Combettes and J. C. Pesquet, "Proximal splitting methods in signal processing," *Springer Optimization and Its Applications*, vol. 49, pp. 185–212, 2011, arXiv: 0912.3522, ISSN: 19316836. DOI: 10.1007/978-1-4419-9569-8_10.

[32] G. Liu, Y. Xu, and K. Tomsovic, "Bidding Strategy for Microgrid in Day-Ahead Market Based on Hybrid Stochastic/Robust Optimization," en, *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 227–237, Jan. 2016, ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2015.2476669. [Online]. Available: http://ieeexplore.ieee.org/document/7273948/.

[33] J. Choi, Y. Shin, M. Choi, W.-K. Park, and I.-W. Lee, "Robust Control of a Microgrid Energy Storage System Using Various Approaches," en, *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2702–2712, May 2019, ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2018.2808914. [Online]. Available: https://ieeexplore.ieee.org/document/8301583/.

[34] S. Nikkhah, A. Allahham, M. Royapoor, J. W. Bialek, and D. Giaouris, "Optimising Building-to-Building and Building-for-Grid Services Under Uncertainty: A Robust Rolling Horizon Approach," en, *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 1453–1467, Mar. 2022, ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2021.3135570. [Online]. Available: https://ieeexplore.ieee.org/document/9651516/.

[35] S. Bengea, V. Adetola, K. Kang, *et al.*, "Parameter estimation of a building system model and impact of estimation error on closed-loop performance," en, in *IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, USA: IEEE, Dec. 2011, pp. 5137–5143, ISBN: 978-1-61284-801-3 978-1-61284-800-6 978-1-4673-0457-3 978-1-61284-799-3. DOI: 10.1109/CDC.2011.6161302. [Online]. Available: http://ieeexplore.ieee.org/document/6161302/.

[36] C. C. Okaeme, S. Mishra, and J. T.-Y. Wen, "Passivity-Based Thermohygrometric Control in Buildings," en, *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1661–1672, Sep. 2018, ISSN: 1063-6536, 1558-0865, 2374-0159. DOI: 10.1109/TCST.2017.2730164. [Online]. Available: https://ieeexplore.ieee.org/document/7999263/.

[37] J. Zhang, S. Luo, C. Xia, Y. Zhu, and R. Xia, "Optimal Power Flow Calculation For Wind Power Grid Connection Based On Adjustable Robust Optimization Theory," en, in *2022 7th Asia Conference on Power and Electrical Engineering (ACPEE)*, Hangzhou, China: IEEE, Apr. 2022, pp. 1927–1931, ISBN: 978-1-66541-819-5. DOI: 10.1109/ACPEE53904.2022.9783825. [Online]. Available: https://ieeexplore.ieee.org/document/9783825/.

[38] X. Yuan, J. Su, C. Yu, and S. Ye, "Power Grid Software Cost Estimation Based on Improved COCOMO Model," en, in *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*, Changchun, China: IEEE, May 2023, pp. 1265–1269, ISBN: 9798350398410. DOI: 10.1109/ICETCI57876.2023.10176686. [Online]. Available: https://ieeexplore.ieee.org/document/10176686/.

[39] M. Zhou, S. Lu, Q. Wu, L. Ma, X. Liao, and X. Zhang, "Research on investment balance rate calculation and computer prediction model of power grid infrastructure projects," en, in *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*, Shenyang, China: IEEE, Jan. 2022, pp. 1117–1121, ISBN: 978-1-66544-276-3. DOI: 10.1109/ICPECA53709.2022.9719289. [Online]. Available: https://ieeexplore.ieee.org/document/9719289/.

[40] X. Bai, L. Qu, and W. Qiao, "Robust AC Optimal Power Flow for Power Networks With Wind Power Generation," en, *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 4163–4164, Sep. 2016, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2015.2493778. [Online]. Available: http://ieeexplore.ieee.org/document/7314992/.

[41] J. Guo, G. Hug, and O. Tonguz, "Impact of communication delay on asynchronous distributed optimal power flow using ADMM," in *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, arXiv: 1711.01702, vol. 2018-Janua, IEEE, Oct. 2017, pp. 177–182, ISBN: 978-1-5386-0943-9. DOI: 10.1109/SmartGridComm.2017.8340718. [Online]. Available: http://ieeexplore.ieee.org/document/8340718/.

[42] M. T. Lawder, B. Suthar, P. W. C. Northrop, *et al.*, "Battery Energy Storage System (BESS) and Battery Management System (BMS) for Grid-Scale Applications," en, *Proceedings of the IEEE*, vol. 102, no. 6, pp. 1014–1030, Jun. 2014, ISSN: 0018-9219, 1558-2256. DOI: 10.1109/JPROC.2014.2317451. [Online]. Available: http://ieeexplore.ieee.org/document/6811152/.

[43] S. Sahu, R. Dutt, and A. Acharyya, "Battery States Co-estimation Methodology Using Dual Square Root Unscented Kalman Filter," en, in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, Monterey, CA, USA: IEEE, May 2023, pp. 1–5, ISBN: 978-1-66545-109-3. DOI: 10.1109/ISCAS46773.2023.10181678. [Online]. Available: https://ieeexplore.ieee.org/document/10181678/.

[44] S. Sundaresan, S. Sunil, B. Balasingam, and K. R. Pattipati, "Joint Estimation of Open Circuit Voltage and Equivalent Circuit Model Parameters Using State-Space Model Optimization," en, in *2023 IEEE Transportation Electrification Conference & Expo (ITEC)*, Detroit, MI, USA: IEEE, Jun. 2023, pp. 1–6, ISBN: 9798350397420. DOI: 10.1109/ITEC55900.2023.10186902. [Online]. Available: https://ieeexplore.ieee.org/document/10186902/.

[45] R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory," en, *Journal of Basic Engineering*, vol. 83, no. 1, pp. 95–108, Mar. 1961, ISSN: 0021-9223. DOI: 10.1115/1.3658902. [Online]. Available: https://asmedigitalcollection.asme.org/fluidsengineering/article/83/1/95/426820/New-Results-in-Linear-Filtering-and-Prediction.

[46] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*, eng, 6. pr. Englewood Cliffs, N.J: Prentice-Hall, 1982, ISBN: 978-0-13-152462-0.

[47] V. V. Vazirani, *Approximation Algorithms*, en. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, ISBN: 978-3-642-08469-0 978-3-662-04565-7. DOI: 10.1007/978-3-662-04565-7. [Online]. Available: http://link.springer.com/10.1007/978-3-662-04565-7.

[48] S. Bubeck, "Convex Optimization: Algorithms and Complexity," en, *Foundations and Trends® in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015, ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000050. [Online]. Available: http://www.nowpublishers.com/article/Details/MAL-050.

[49] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Mar. 2004, vol. 100, Publication Title: Journal of the American Statistical Association Issue: 471 ISSN: 0162-1459, ISBN: 978-0-521-83378-3. DOI: 10.1017/CBO9780511804441. [Online]. Available: https://www.cambridge.org/core/product/identifier/9780511804441/type/book.

[50] M. V. Solodov, "Constraint Qualifications," en, in *Wiley Encyclopedia of Operations Research and Management Science*, 1st ed., Wiley, Jan. 2011, ISBN: 978-0-470-40063-0 978-0-470-40053-1. DOI: 10.1002/9780470400531.eorms0978.

[Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/9780470400531.eorms0978.

[51] G. Wachsmuth, "On LICQ and the uniqueness of Lagrange multipliers," *Operations Research Letters*, vol. 41, no. 1, pp. 78–80, Jan. 2013, ISSN: 01676377. DOI: 10.1016/j.orl.2012.11.009.

[52] R. T. Rockafellar, *Convex Analysis:* Princeton University Press, Dec. 1970, ISBN: 978-1-4008-7317-3. DOI: 10.1515/9781400873173. [Online]. Available: https://www.degruyter.com/document/doi/10.1515/9781400873173/html.

[53] R. Rockafellar, "On the maximal monotonicity of subdifferential mappings," en, *Pacific Journal of Mathematics*, vol. 33, no. 1, pp. 209–216, Apr. 1970, ISSN: 0030-8730, 0030-8730. DOI: 10.2140/pjm.1970.33.209. [Online]. Available: http://msp.org/pjm/1970/33-1/p19.xhtml.

[54] Y. Nesterov, *Introductory Lectures on Convex Optimization* (Applied Optimization), P. M. Pardalos and D. W. Hearn, Eds. Boston, MA: Springer US, 2004, vol. 87, ISBN: 978-1-4613-4691-3 978-1-4419-8853-9. DOI: 10.1007/978-1-4419-8853-9. [Online]. Available: http://link.springer.com/10.1007/978-1-4419-8853-9.

[55] D. P. Bertsekas, *Convex optimization algorithms*, eng. Nashua: Athena scientific, 2015, ISBN: 978-1-886529-28-1.

[56] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces* (CMS Books in Mathematics), en. Cham: Springer International Publishing, 2017, ISBN: 978-3-319-48310-8 978-3-319-48311-5. DOI: 10.1007/978-3-319-48311-5. [Online]. Available: https://link.springer.com/10.1007/978-3-319-48311-5.

[57] R. T. Rockafellar and R. J. B. Wets, *Variational Analysis* (Grundlehren der mathematischen Wissenschaften), M. Berger, P. De La Harpe, F. Hirzebruch, *et al.*, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, vol. 317, ISBN: 978-3-540-62772-2 978-3-642-02431-3. DOI: 10.1007/978-3-642-02431-3. [Online]. Available: http://link.springer.com/10.1007/978-3-642-02431-3.

[58] P. L. Combettes, "Quasi-Fejérian Analysis of Some Optimization Algorithms," en, in *Studies in Computational Mathematics*, vol. 8, Elsevier, 2001, pp. 115–152, ISBN: 978-0-444-50595-8. DOI: 10.1016/S1570-579X(01)80010-0. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1570579X01800100.

[59] P. L. Combettes, "Fejér Monotonicity in Convex Optimization," en, in *Encyclopedia of Optimization*, C. A. Floudas and P. M. Pardalos, Eds., Boston, MA: Springer US, 2008, pp. 1016–1024, ISBN: 978-0-387-74758-3 978-0-387-74759-0. DOI: 10.1007/978-0-387-74759-0_179. [Online]. Available: https://link.springer.com/10.1007/978-0-387-74759-0_179.

[60] H. H. Bauschke, M. N. Dao, and W. M. Moursi, "On Fejér monotone sequences and nonexpansive mappings," *ArXiv*, 2015, Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1507.05585. [Online]. Available: https://arxiv.org/abs/1507.05585.

[61] F. E. Browder, "Semicontractive and semiaccretive nonlinear mappings in Banach spaces," en, *Bulletin of the American Mathematical Society*, vol. 74, no. 4, pp. 660–665, 1968, ISSN: 0273-0979, 1088-9485. DOI: 10.1090/S0002-9904-1968-11983-4. [Online]. Available: https://www.ams.org/bull/1968-74-04/S0002-9904-1968-11983-4/.

[62] S. Banach, "Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales," en, *Fundamenta Mathematicae*, vol. 3, pp. 133–181, 1922, ISSN: 0016-2736, 1730-6329. DOI: 10.4064/fm-3-1-133-181. [Online]. Available: http://www.impan.pl/get/doi/10.4064/fm-3-1-133-181.

[63] W. R. Mann, "Mean value methods in iteration," en, *Proceedings of the American Mathematical Society*, vol. 4, no. 3, pp. 506–510, Jun. 1953, ISSN: 0002-9939, 1088-6826. DOI: 10.1090/S0002-9939-1953-0054846-3. [Online]. Available: https://www.ams.org/proc/1953-004-03/S0002-9939-1953-0054846-3/.

[64] M. A. Cauchy, "M´ethode ge´ne´rale pour la r´esolution des syste‘mes d'e´quations simultane´es," en, *Comp. Rend. Sci. Paris*, vol. 25(1847), pp. 536–538, 1847.

[65] S. Boyd, "Subgradient methods," 2003.

[66] P. Tseng, "Dual ascent methods for problems with strictly convex costs and linear constraints. A unified approach," *SIAM Journal on Control and Optimization*, vol. 28, no. 1, pp. 214–242, 1990, ISSN: 03630129. DOI: 10.1137/0328011.

[67] P. Tseng and D. P. Bertsekas, "Relaxation methods for problems with strictly convex separable costs and linear constraints," *Mathematical Programming*, vol. 38, no. 3, pp. 303–321, Oct. 1987, ISSN: 0025-5610. DOI: 10.1007/BF02592017. [Online]. Available: http://link.springer.com/10.1007/BF02592017.

[68] F. J. Aragón Artacho, J. M. Borwein, V. Martín-Márquez, and L. Yao, "Applications of convex analysis within mathematics," en, *Mathematical Programming*, vol. 148, no. 1-2, pp. 49–88, Dec. 2014, ISSN: 0025-5610, 1436-4646. DOI: 10.1007/s10107-013-0707-3. [Online]. Available: http://link.springer.com/10.1007/s10107-013-0707-3.

[69] G. J. Minty, "Monotone (nonlinear) operators in Hilbert space," *Duke Mathematical Journal*, vol. 29, no. 3, Sep. 1962, ISSN: 0012-7094. DOI: 10.1215/S0012-7094-62-02933-2. [Online]. Available: https://projecteuclid.org/journals/duke-mathematical-journal/volume-29/issue-3/Monotone-nonlinear-operators-in-Hilbert-space/10.1215/S0012-7094-62-02933-2.full.

[70] A. Auslender and M. Teboulle, *Asymptotic Cones and Functions in Optimization and Variational Inequalities* (Springer Monographs in Mathematics), eng. New York, NY: Springer-Verlag New York, Inc, 2003, ISBN: 978-0-387-22590-6. DOI: 10.1007/b97594.

[71] B. Martinet, "Brève communication. Régularisation d'inéquations variationnelles par approximations successives," fr, *Revue française d'informatique et de recherche opérationnelle. Série rouge*, vol. 4, no. R3, pp. 154–158, 1970, ISSN: 0373-8000. DOI: 10.1051/m2an/197004R301541. [Online]. Available: http://www.esaim-m2an.org/10.1051/m2an/197004R301541.

[72] R. T. Rockafellar, "Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming.," *Mathematics of Operations Research*, vol. 1, no. 2, pp. 97–116, 1976, ISSN: 0364765X. DOI: 10.1287/moor.1.2.97.

[73] H. Brezis and P. L. Lions, "Produits infinis de resolvantes," fr, *Israel Journal of Mathematics*, vol. 29, no. 4, pp. 329–345, Dec. 1978, ISSN: 1565-8511. DOI: 10.1007/BF02761171. [Online]. Available: https://doi.org/10.1007/BF02761171.

[74] B. Lemaire, "About the Convergence of the Proximal Method," en, in *Advances in Optimization*, W. Oettli and D. Pallaschke, Eds., Berlin, Heidelberg: Springer, 1992, pp. 39–51, ISBN: 978-3-642-51682-5. DOI: 10.1007/978-3-642-51682-5_4.

[75] D. P. Palomar and Y. C. Eldar, *Convex Optimization in Signal Processing and Communications*, en. Cambridge University Press, 2010, Google-Books-ID: UOpnvPJ151gC, ISBN: 978-0-521-76222-9.

[76] L. Vandenberghe, "Fast proximal gradient methods," *EE236C course notes, Online, http://www. seas. ucla. edu/vandenbe C*, vol. 236, 2010.

[77] A. S. Nemirovskii and D. B. Yudin, *Problem complexity and method efficiency in optimization* (Wiley-Interscience series in discrete mathematics), engrus. Chichester ; New York: Wiley, 1983, ISBN: 978-0-471-10345-5.

[78] Y. E. Nesterov, "A Method of Solving a Convex Programming Problem with Convergence Rate $\mathcal{O}(1/k^2)$," *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376, 1983. [Online]. Available: http://mi.mathnet.ru/eng/dan46009.

[79] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *submitted to SIAM Journal on Optimization*, vol. 2, no. 3, 2008.

[80] S. R. Becker, E. J. Candès, and M. C. Grant, "Templates for convex cone problems with applications to sparse signal recovery," en, *Mathematical Programming Computation*, vol. 3, no. 3, pp. 165–218, Sep. 2011, ISSN: 1867-2957. DOI: 10.1007/s12532-011-0029-5. [Online]. Available: https://doi.org/10.1007/s12532-011-0029-5.

[81] M. R. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, vol. 4, no. 5, pp. 303–320, 1969, ISSN: 00223239. DOI: 10.1007/BF00927673.

[82] M. Powell, "A method for nonlinear constraints in minimization problems.," in *Optimization: Symposium of the Institute of Mathematics and Its Applications*, 1969, pp. 283–298.

[83] N. Ogura and I. Yamada, "NON-STRICTLY CONVEX MINIMIZATION OVER THE FIXED POINT SET OF AN ASYMPTOTICALLY SHRINKING NONEXPANSIVE MAPPING," en, *Numerical Functional Analysis and Optimization*, vol. 23, no. 1-2, pp. 113–137, Jan. 2002, ISSN: 0163-0563, 1532-2467. DOI: 10.1081/NFA-120003674. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1081/NFA-120003674.

[84] G. B. Passty, "Ergodic convergence to a zero of the sum of monotone operators in Hilbert space," en, *Journal of Mathematical Analysis and Applications*, vol. 72, no. 2, pp. 383–390, Dec. 1979, ISSN: 0022247X. DOI: 10.1016/0022-247X(79)90234-8. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0022247X79902348.

[85] P. L. Combettes and V. R. Wajs, "Signal Recovery by Proximal Forward-Backward Splitting," en, *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, Jan. 2005, ISSN: 1540-3459, 1540-3467. DOI: 10.1137/050626090. [Online]. Available: http://epubs.siam.org/doi/10.1137/050626090.

[86] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009, Publisher: Society for Industrial and Applied Mathematics Publications, ISSN: 19364954. DOI: 10.1137/080716542.

[87] J. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *Journal of Machine Learning Research*, vol. 10, no. 99, pp. 2899–2934, 2009. [Online]. Available: http://jmlr.org/papers/v10/duchi09a.html.

[88] D. W. Peaceman and H. H. Rachford Jr., "The Numerical Solution of Parabolic and Elliptic Differential Equations," *Journal of the Society for Industrial and Applied Mathematics*, vol. 3, no. 1, pp. 28–41, Mar. 1955, ISSN: 0368-4245. DOI: 10.1137/0103003. [Online]. Available: http://epubs.siam.org/doi/10.1137/0103003.

[89] P. L. Lions and B. Mercier, "Splitting Algorithms for the Sum of Two Nonlinear Operators," *SIAM Journal on Numerical Analysis*, vol. 16, no. 6, pp. 964–979, Dec. 1979, Publisher: Society for Industrial and Applied Mathematics, ISSN: 0036-1429. DOI: 10.1137/0716071. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/0716071.

[90] R. B. Kellogg, "A nonlinear alternating direction method," en, *Mathematics of Computation*, vol. 23, no. 105, pp. 23–27, 1969, ISSN: 0025-5718, 1088-6842. DOI: 10.1090/S0025-5718-1969-0238507-3. [Online]. Available: https://www.ams.org/mcom/1969-23-105/S0025-5718-1969-0238507-3/.

[91] J. Douglas and H. H. Rachford, "On the Numerical Solution of Heat Conduction Problems in Two and Three Space Variables," *Transactions of the American Mathematical Society*, vol. 82, no. 2, p. 421, 1956, ISSN: 00029947. DOI: 10.2307/1993056.

[92] M. Yan and W. Yin, "Self Equivalence of the Alternating Direction Method of Multipliers," *ArXiv*, pp. 165–194, 2016, arXiv: 1407.7400. DOI: 10.1007/978-3-319-41589-5_5.

[93] D. Gabay, "Chapter IX Applications of the Method of Multipliers to Variational Inequalities," in *Studies in Mathematics and its Applications*, vol. 15, Issue: C ISSN: 01682024, 1983, pp. 299–331. DOI: 10.1016/S0168-2024(08)70034-1. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0168202408700341.

[94] P. Tseng, "A Modified Forward-Backward Splitting Method for Maximal Monotone Mappings," en, *SIAM Journal on Control and Optimization*, vol. 38, no. 2, pp. 431–446, Jan. 2000, ISSN: 0363-0129, 1095-7138. DOI: 10.1137/S0363012998338806. [Online]. Available: http://epubs.siam.org/doi/10.1137/S0363012998338806.

[95] D. Davis and W. Yin, "A Three-Operator Splitting Scheme and its Optimization Applications," en, *Set-Valued and Variational Analysis*, vol. 25, no. 4, pp. 829–858, Dec. 2017, ISSN: 1877-0541. DOI: 10.1007/s11228-017-0421-z. [Online]. Available: https://doi.org/10.1007/s11228-017-0421-z.

[96] A. Chambolle and T. Pock, "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging," en, *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, May 2011, ISSN: 1573-7683. DOI: 10.1007/s10851-010-0251-1. [Online]. Available: https://doi.org/10.1007/s10851-010-0251-1.

[97] T. Goldstein and S. Osher, "The Split Bregman Method for L1-Regularized Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, Jan. 2009, Publisher: Society for Industrial and Applied Mathematics. DOI: 10.1137/080725891. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/080725891.

[98] E. Esser, X. Zhang, and T. F. Chan, "A General Framework for a Class of First Order Primal-Dual Algorithms for Convex Optimization in Imaging Science," *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 1015–1046, Jan. 2010, Publisher: Society for Industrial and Applied Mathematics. DOI: 10.1137/09076934X. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/09076934X.

[99] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," en, *Advances in Computational Mathematics*, vol. 38, no. 3, pp. 667–681, Apr. 2013, ISSN: 1572-9044. DOI: 10.1007/s10444-011-9254-8. [Online]. Available: https://doi.org/10.1007/s10444-011-9254-8.

[100] L. Condat, "A Primal–Dual Splitting Method for Convex Optimization Involving Lipschitzian, Proximable and Linear Composite Terms," en, *Journal of Optimization Theory and Applications*, vol. 158, no. 2, pp. 460–479, Aug. 2013, ISSN: 1573-2878. DOI: 10.1007/s10957-012-0245-9. [Online]. Available: https://doi.org/10.1007/s10957-012-0245-9.

[101] M. Zhu and T. Chan, "An Efficient Primal-Dual Hybrid Gradient Algorithm For Total Variation Image Restoration," en, UCLA, UCLA CAM Report, May 2008.

[102] L. M. Briceño-Arias and P. L. Combettes, "A Monotone+Skew Splitting Model for Composite Monotone Inclusions in Duality," *SIAM Journal on Optimization*, vol. 21, no. 4, pp. 1230–1250, Oct. 2011, Publisher: Society for Industrial and Applied Mathematics, ISSN: 1052-6234. DOI: 10.1137/10081602X. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/10081602X.

[103] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, Dec. 2010, ISBN: 978-1-4008-3535-5. DOI: 10.1515/9781400835355. [Online]. Available: https://www.degruyter.com/document/doi/10.1515/9781400835355/html.

[104] P. Wang, P. Lin, W. Ren, and Y. Song, "Distributed subgradient-based multiagent optimization with more general step sizes," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 2295–2302, 2018, arXiv: 1602.00048 Publisher: IEEE, ISSN: 00189286. DOI: 10.1109/TAC.2017.2763782.

[105] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An Exact First-Order Algorithm for Decentralized Consensus Optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, Jan. 2015, ISSN: 1052-6234. DOI: 10.1137/14096668X. [Online]. Available: http://epubs.siam.org/doi/10.1137/14096668X.

[106] A. Nedić, A. Olshevsky, and W. Shi, "Achieving Geometric Convergence for Distributed Optimization Over Time-Varying Graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, Jan. 2017, ISSN: 1052-6234. DOI: 10.1137/16M1084316. [Online]. Available: https://epubs.siam.org/doi/10.1137/16M1084316.

[107] J. N. Tsitsiklis, *Problems in decentralized decision making and computation OR - Massachusetts Institute of Technology*, Publication Title: Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, 1984. [Online]. Available: citeulike-article-id:10514594.

[108] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986, ISSN: 15582523. DOI: 10.1109/TAC.1986.1104412.

[109] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997, ISBN: 1-886529-01-9.

[110] C.-H. Zhang and T. Zhang, "A General Theory of Concave Regularization for High-Dimensional Sparse Estimation Problems," *Statistical Science*, vol. 27, no. 4, pp. 576–593, Nov. 2012, Publisher: Institute of Mathematical Statistics, ISSN: 0883-4237, 2168-8745. DOI: 10.1214/12-STS399. [Online]. Available: https://projecteuclid.org/journals/statistical-science/volume-27/issue-4/A-General-Theory-of-Concave-Regularization-for-High-Dimensional-Sparse/10.1214/12-STS399.full.

[111] Y. Zhou, Y. Liang, Y. Yu, W. Dai, and E. P. Xing, "Distributed proximal gradient algorithm for partially asynchronous computer clusters," *Journal of Machine Learning Research*, vol. 19, no. 19, pp. 1–32, 2018. [Online]. Available: http://jmlr.org/papers/v19/17-444.html.

[112] A. A. Goldstein, "Convex programming in Hilbert space," en, *Bulletin of the American Mathematical Society*, vol. 70, no. 5, pp. 709–710, 1964, ISSN: 0273-0979, 1088-9485. DOI: 10.1090/S0002-9904-1964-11178-2. [Online]. Available: https://www.ams.org/bull/1964-70-05/S0002-9904-1964-11178-2/.

[113] E. S. Levitin and B. T. Polyak, "Constrained minimization methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 6, no. 5, pp. 1–50, Jan. 1966, ISSN: 0041-5553. DOI: 10.1016/0041-5553(66)90114-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0041555366901145.

[114] R. T. Rockafellar, "The multiplier method of Hestenes and Powell applied to convex programming," *Journal of Optimization Theory and Applications*, vol. 12, no. 6, pp. 555–562, 1973, ISSN: 00223239. DOI: 10.1007/BF00934777.

[115] R. Rockafellar, "MONOTONE OPERATORS AND AUGMENTED LAGRANGIAN METHODS IN NONLINEAR PROGRAMMING," in *Nonlinear Programming 3*, Elsevier, 1978, pp. 1–25. DOI: 10.1016/B978-0-12-468660-1.50006-2. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B9780124686601500062.

[116] A. Ruszczynski, "On convergence of an augmented lagrangian decomposition method for sparse convex optimization," *Mathematics of Operations Research*, vol. 20, no. 3, pp. 634–656, 1995, ISSN: 0364765X. DOI: 10.1287/moor.20.3.634.

[117] N. Chatzipanagiotis, D. Dentcheva, and M. M. Zavlanos, "An augmented Lagrangian method for distributed optimization," *Mathematical Programming*, vol. 152, no. 1-2, pp. 405–434, 2015, Publisher: Springer Berlin Heidelberg, ISSN: 14364646. DOI: 10.1007/s10107-014-0808-7. [Online]. Available: http://dx.doi.org/10.1007/s10107-014-0808-7.

[118] L. Chen, D. Sun, and K. C. Toh, "A note on the convergence of ADMM for linearly constrained convex optimization problems," *Computational Optimization and Applications*, vol. 66, no. 2, pp. 327–343, 2017, arXiv: 1507.02051 Publisher: Springer US, ISSN: 15732894. DOI: 10.1007/s10589-016-9864-7.

[119] B. He and X. Yuan, "On the $O(1/n)$ Convergence Rate of the Douglas–Rachford Alternating Direction Method," *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, Jan. 2012, ISBN: 2011009111, ISSN: 0036-1429. DOI: 10.1137/110836936. [Online]. Available: http://epubs.siam.org/doi/10.1137/110836936.

[120] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the Linear Convergence of the ADMM in Decentralized Consensus Optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, Apr. 2014, ISSN: 1053-587X. DOI: 10.1109/TSP.2014.2304432.

[121] W. Deng and W. Yin, "On the Global and Linear Convergence of the Generalized Alternating Direction Method of Multipliers," *Journal of Scientific Computing*, vol. 66, no. 3, pp. 889–916, 2016, Publisher: Springer US, ISSN: 08857474. DOI: 10.1007/s10915-015-0048-x.

[122] D. Davis and W. Yin, "Convergence Rate Analysis of Several Splitting Schemes," in *Splitting Methods in Communication, Imaging, Science, and Engineering*, arXiv: 1406.4834, Springer, Jun. 2016, pp. 115–163, ISBN: 978-3-319-41587-1. DOI: 10.1007/978-3-319-41589-5_4. [Online]. Available: http://link.springer.com/10.1007/978-3-319-41589-5_4.

[123] B. S. He, H. Yang, and S. L. Wang, "Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities," *Journal of Optimization Theory and Applications*, vol. 106, no. 2, pp. 337–356, 2000, ISSN: 00223239. DOI: 10.1023/A:1004603514434.

[124] S. L. Wang and L. Z. Liao, "Decomposition method with a variable parameter for a class of monotone variational inequality problems," *Journal of Optimization Theory and Applications*, vol. 109, no. 2, pp. 415–429, 2001, ISSN: 00223239. DOI: 10.1023/A:1017522623963.

[125] J. Giesen and S. Laue, "Combining ADMM and the augmented Lagrangian method for efficiently handling many constraints," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2019-Augus, pp. 4525–4531, 2019, ISBN: 9780999241141, ISSN: 10450823. DOI: 10.24963/ijcai.2019/629.

[126] T. H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 482–497, 2015, arXiv: 1402.6065 Publisher: IEEE, ISSN: 1053587X. DOI: 10.1109/TSP.2014.2367458.

[127] T. H. Chang, "A Proximal Dual Consensus ADMM Method for Multi-Agent Constrained Optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3719–3734, 2016, arXiv: 1409.3307 Publisher: IEEE, ISSN: 1053587X. DOI: 10.1109/TSP.2016.2544743.

[128] Y. Cui, X. Li, D. Sun, and K. C. Toh, "On the Convergence Properties of a Majorized Alternating Direction Method of Multipliers for Linearly Constrained Convex Optimization Problems with Coupled Objective Functions," *Journal of Optimization Theory and Applications*, vol. 169, no. 3, pp. 1013–1041, 2016, arXiv: 1502.00098v1, ISSN: 15732878. DOI: 10.1007/s10957-016-0877-2.

[129] E. K. Ryu, Y. Liu, and W. Yin, "Douglas–Rachford splitting and ADMM for pathological convex optimization," *Computational Optimization and Applications*, vol. 74, no. 3, pp. 747–778, 2019, arXiv: 1801.06618 Publisher: Springer US, ISSN: 15732894. DOI: 10.1007/s10589-019-00130-9. [Online]. Available: https://doi.org/10.1007/s10589-019-00130-9.

[130] D. Han and X. Yuan, "A Note on the Alternating Direction Method of Multipliers," *Journal of Optimization Theory and Applications*, vol. 155, no. 1, pp. 227–238, 2012, ISSN: 00223239. DOI: 10.1007/s10957-012-0003-z.

[131] W. Deng, M. J. Lai, Z. Peng, and W. Yin, "Parallel Multi-Block ADMM with o(1/k) Convergence," *Journal of Scientific Computing*, vol. 71, no. 2, pp. 712–736, 2017, arXiv: 1312.3040 Publisher: Springer US, ISSN: 08857474. DOI: 10.1007/s10915-016-0318-2.

[132] T. Y. Lin, S. Q. Ma, and S. Z. Zhang, "On the Sublinear Convergence Rate of Multi-block ADMM," *Journal of the Operations Research Society of China*, vol. 3, no. 3, pp. 251–274, Aug. 2015, arXiv: 1408.4265, ISSN: 21946698. DOI: 10.1007/s40305-015-0092-0. [Online]. Available: http://arxiv.org/abs/1408.4265.

[133] W. Gao, D. Goldfarb, and F. E. Curtis, "ADMM for multiaffine constrained optimization," *Optimization Methods and Software*, vol. 35, no. 2, pp. 257–303, 2020, arXiv: 1802.09592, ISSN: 10294937. DOI: 10.1080/10556788.2019.1683553.

[134] L. Liu and Z. Han, "Multi-block ADMM for big data optimization in smart grid," in *2015 International Conference on Computing, Networking and Communications, ICNC 2015*, arXiv: 1503.00054, Institute of Electrical and Electronics Engineers Inc., Mar. 2015, pp. 556–561, ISBN: 978-1-4799-6959-3. DOI: 10.1109/ICCNC.2015.7069405.

[135] Z. Li, Q. Guo, H. Sun, and H. Su, "ADMM-based decentralized demand response method in electric vehicle virtual power plant," in *IEEE Power and Energy Society General Meeting*, ISSN: 19449933, vol. 2016-Novem, IEEE Computer Society, Nov. 2016, ISBN: 978-1-5090-4168-8. DOI: 10.1109/PESGM.2016.7741146.

[136] G. Chen and J. Li, "A fully distributed ADMM-based dispatch approach for virtual power plant problems," *Applied Mathematical Modelling*, vol. 58, pp. 300–312, 2018, Publisher: Elsevier Inc., ISSN: 0307904X. DOI: 10.1016/j.apm.2017.06.010. [Online]. Available: https://doi.org/10.1016/j.apm.2017.06.010.

[137] Y. Li, G. Shi, W. Yin, L. Liu, and Z. Han, "A Distributed ADMM Approach with Decomposition-Coordination for Mobile Data Offloading," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2514–2530, 2018, Publisher: IEEE, ISSN: 00189545. DOI: 10.1109/TVT.2017.2760920.

[138] R. Van Parys and G. Pipeleers, "Online distributed motion planning for multi-vehicle systems," *2016 European Control Conference, ECC 2016*, vol. 32, pp. 1580–1585, 2016, ISBN: 9781509025916. DOI: 10.1109/ECC.2016.7810516.

[139] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Cooperative Distributed Collision Avoidance Based on ADMM for Waterborne AGVs," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9335, ISSN: 16113349, Springer Verlag, 2015, pp. 181–194, ISBN: 978-3-319-24263-7. DOI: 10.1007/978-3-319-24264-4_13. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-24264-4_13.

[140] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Fast ADMM for Distributed Model Predictive Control of Cooperative Waterborne AGVs," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1406–1413, Jul. 2017, Publisher: Institute of Electrical and Electronics Engineers Inc., ISSN: 1063-6536. DOI: 10.1109/TCST.2016.2599485. [Online]. Available: http://ieeexplore.ieee.org/document/7572966/.

[141] F. Rey, Z. Pan, A. Hauswirth, and J. Lygeros, "Fully Decentralized ADMM for Coordination and Collision Avoidance," en, in *2018 European Control Conference (ECC)*, Limassol: IEEE, Jun. 2018, pp. 825–830, ISBN: 978-3-9524269-8-2. DOI: 10.23919/ECC.2018.8550245. [Online]. Available: https://ieeexplore.ieee.org/document/8550245/.

[142] D. Luenberger, "An introduction to observers," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 596–602, Dec. 1971, Conference Name: IEEE Transactions on Automatic Control, ISSN: 1558-2523. DOI: 10.1109/TAC.1971.1099826. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1099826.

[143] R. L. Plackett, "Some Theorems in Least Squares," *Biometrika*, vol. 37, no. 1/2, pp. 149–157, 1950, Publisher: [Oxford University Press, Biometrika Trust], ISSN: 0006-3444. DOI: 10.2307/2332158. [Online]. Available: https://www.jstor.org/stable/2332158.

[144] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," en, *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960, ISSN: 0021-9223. DOI: 10.1115/1.3662552. [Online]. Available: https://asmedigitalcollection.asme.org/fluidsengineering/article/82/1/35/397706/A-New-Approach-to-Linear-Filtering-and-Prediction.

[145] S. S. Haykin, *Adaptive filter theory* (Prentice Hall informations and system sciences series), eng, 4. ed., International ed. Upper Saddle River, NJ: Prentice Hall, 2002, ISBN: 978-0-13-048434-5.

[146] P. S. Maybeck, *Stochastic models, estimation, and control* (Mathematics in science and engineering v. 141-3), eng. New York: Academic Press, 1982, ISBN: 978-0-08-096003-6.

[147] E. Hall, "MIT's role in project Apollo: Final report on contracts NAS 9-163 and NAS 94065," Technical report R-700 (Charles Stark Draper Laboratory, MIT, Cambridge), Tech. Rep., 1972.

[148] D. Simon, *Optimal state estimation: Kalman, H [infinity] and nonlinear approaches*, eng. Hoboken, N.J: Wiley-Interscience, 2006, ISBN: 978-0-471-70858-2 978-0-470-04534-3 978-0-470-04533-6 978-1-61583-476-1 978-1-280-50795-3. DOI: 10.1002/0470045345.

[149] D. P. Bertsekas, *Dynamic programming and optimal control* (Athena scientific optimization and computation series), eng, 4th edition. Nashua, NH: Athena scientific, 2017, ISBN: 978-1-886529-43-4 978-1-886529-44-1.

[150] B. Kouvaritakis and M. Cannon, *Model Predictive Control* (Advanced Textbooks in Control and Signal Processing). Cham: Springer International Publishing, 2016, ISBN: 978-3-319-24851-6 978-3-319-24853-0. DOI: 10.1007/978-3-319-24853-0. [Online]. Available: http://link.springer.com/10.1007/978-3-319-24853-0.

[151] J. M. Morales, A. J. Conejo, and J. Pérez-Ruiz, "Economic valuation of reserves in power systems with high penetration of wind power," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 900–910, 2009, Publisher: Institute of Electrical and Electronics Engineers Inc., ISSN: 08858950. DOI: 10.1109/TPWRS.2009.2016598.

[152] A. Nedić, D. P. Bertsekas, and V. S. Borkar, "Distributed asynchronous incremental subgradient methods," *Studies in Computational Mathematics*, vol. 8, no. C, pp. 381–407, 2001, ISSN: 1570579X. DOI: 10.1016/S1570-579X(01)80023-9.

[153] A. Nedić and A. Ozdaglar, "Convergence rate for consensus with delays," *Journal of Global Optimization*, vol. 47, no. 3, pp. 437–456, 2010, ISSN: 09255001. DOI: 10.1007/s10898-008-9370-2.

[154] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," *Proceedings of the IEEE Conference on Decision and Control*, pp. 5451–5452, 2012, arXiv: 1104.5525 Publisher: IEEE ISBN: 9781467320665, ISSN: 01912216. DOI: 10.1109/CDC.2012.6426626.

[155] I. Notarnicola and G. Notarstefano, "Asynchronous Distributed Optimization Via Randomized Dual Proximal Gradient," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2095–2106, 2017, arXiv: 1509.08373 Publisher: IEEE, ISSN: 00189286. DOI: 10.1109/TAC.2016.2607023.

[156] R. Hannah and W. Yin, "On Unbounded Delays in Asynchronous Parallel Fixed-Point Algorithms," *Journal of Scientific Computing*, vol. 76, no. 1, pp. 299–326, 2018, arXiv: 1609.04746 Publisher: Springer US, ISSN: 08857474. DOI: 10.1007/s10915-017-0628-z. [Online]. Available: https://doi.org/10.1007/s10915-017-0628-z.

[157] X. Lian, W. Zhang, C. Zhang, and J. Liu, "Asynchronous decentralized parallel stochastic gradient descent," *35th International Conference on Machine Learning, ICML 2018*, vol. 7, pp. 4745–4767, 2018, arXiv: 1710.06952 ISBN: 9781510867963.

[158] K. Mishchenko, F. Iutzeler, J. Malick, and M.-R. R. Amini, "A delay-tolerant proximal-gradient algorithm for distributed learning," *35th International Conference on Machine Learning, ICML 2018*, vol. 8, pp. 5774–5788, 2018, ISBN: 9781510867963.

[159] R. Zhang and J. T. Kwok, "Asynchronous distributed ADMM for consensus optimization," *31st International Conference on Machine Learning, ICML 2014*, vol. 5, no. 2, pp. 3689–3697, 2014, ISBN: 9781634393973.

[160] L. Fang and Y. Lei, "An Asynchronous Distributed ADMM Algorithm and Efficient Communication Model," *Proceedings - 2016 IEEE 14th International Conference on Dependable, Autonomic and Secure Computing, DASC 2016, 2016 IEEE 14th International Conference on Pervasive Intelligence and Computing, PICom 2016, 2016 IEEE 2nd International Conference on Big Data*, no. 2, pp. 136–140, 2016, Publisher: IEEE ISBN: 9781509040650. DOI: 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.41.

[161] S. Kumar, R. Jain, and K. Rajawat, "Asynchronous Optimization Over Heterogeneous Networks Via Consensus ADMM," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 1, pp. 114–129, 2017, arXiv: 1605.00076, ISSN: 2373776X. DOI: 10.1109/TSIPN.2016.2593896.

[162] S. Jiang, Y. Lei, S. Wang, and D. Wang, "An Asynchronous ADMM Algorithm for Distributed Optimization with Dynamic Scheduling Strategy," in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, IEEE, Aug. 2019, pp. 1–8, ISBN: 978-1-72812-058-4. DOI: 10.1109/HPCC/

SmartCity/DSS.2019.00016. [Online]. Available: https://ieeexplore.ieee.org/document/8855654/.

[163] T. H. Chang, M. Hong, W. C. Liao, and X. Wang, "Asynchronous Distributed ADMM for Large-Scale Optimization- Part I: Algorithm and Convergence Analysis - Online," *arXiv*, p. 38, Sep. 2015, arXiv: 1509.02597. [Online]. Available: http://arxiv.org/abs/1509.02597.

[164] T. H. Chang, M. Hong, W. C. Liao, and X. Wang, "Asynchronous distributed alternating direction method of multipliers: Algorithm and convergence analysis," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2016-May, pp. 4781–4785, 2016, Publisher: IEEE ISBN: 9781479999880, ISSN: 15206149. DOI: 10.1109/ICASSP.2016.7472585.

[165] M. Hong, "A distributed, asynchronous, and incremental algorithm for nonconvex optimization: An ADMM approach," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 935–945, 2018, arXiv: 1412.6058 Publisher: IEEE, ISSN: 23255870. DOI: 10.1109/TCNS.2017.2657460.

[166] I. 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model*. [Online]. Available: https://www.iso.org/standard/20269.html.

[167] L. Ljung, "System Identification," in *Signal Analysis and Prediction*, J. J. Benedetto, A. Procházka, J. Uhlíř, P. W. J. Rayner, and N. G. Kingsbury, Eds., Series Title: Applied and Numerical Harmonic Analysis, Boston, MA: Birkhäuser Boston, 1998, pp. 163–173, ISBN: 978-1-4612-7273-1 978-1-4612-1768-8. DOI: 10.1007/978-1-4612-1768-8_11. [Online]. Available: http://link.springer.com/10.1007/978-1-4612-1768-8_11.

[168] M. Schulze Darup and G. Book, "On Closed-Loop Dynamics of ADMM-Based MPC," in *Lecture Notes in Control and Information Sciences*, vol. 485, arXiv: 1911.02641 ISSN: 16107411, Springer Science and Business Media Deutschland GmbH, 2021, pp. 107–134. DOI: 10.1007/978-3-030-63281-6_5.

[169] A.-s. Esteki, S. S. Kia, and S. Member, "Distributed Optimal Resource Allocation with Time-Varying Quadratic Cost Functions and Resources over Switching Agents," in *2022 European Control Conference (ECC)*, 2022, pp. 441–446, ISBN: 978-3-907144-07-7.

[170] P. Giselsson, M. Fält, and S. Boyd, "Line Search for Averaged Operator Iteration," *ArXiv*, Mar. 2016, arXiv: 1603.06772. [Online]. Available: http://arxiv.org/abs/1603.06772.

[171] H. H. Bauschke, S. M. Moffat, and X. Wang, "Firmly Nonexpansive Mappings and Maximally Monotone Operators: Correspondence and Duality," *Set-Valued and Variational Analysis*, vol. 20, no. 1, pp. 131–153, Mar. 2012, arXiv: 1101.4688, ISSN: 09276947. DOI: 10.1007/s11228-011-0187-7.

[172] M. Vidyasagar, *Nonlinear Systems Analysis*, en, Second. Society for Industrial and Applied Mathematics, Jan. 2002, ISBN: 978-0-89871-526-2 978-0-89871-918-5. DOI: 10.1137/1.9780898719185. [Online]. Available: http://epubs.siam.org/doi/book/10.1137/1.9780898719185.

[173] A. J. King and S. W. Wallace, *Modeling with Stochastic Programming* (Springer Series in Operations Research and Financial Engineering), en. Cham: Springer International Publishing, 2024, ISBN: 978-3-031-54549-8 978-3-031-54550-4. DOI: 10.1007/978-3-031-54550-4. [Online]. Available: https://link.springer.com/10.1007/978-3-031-54550-4.

[174] N. Hermer, D. R. Luke, and A. Sturm, *Nonexpansive Markov Operators and Random Function Iterations for Stochastic Fixed Point Problems*, en, arXiv:2205.15897 [math], Mar. 2023. [Online]. Available: http://arxiv.org/abs/2205.15897.

[175] P. Bianchi, W. Hachem, and F. Iutzeler, "A Coordinate Descent Primal-Dual Algorithm and Application to Distributed Asynchronous Optimization," en, *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2947–2957, Oct. 2016, ISSN: 0018-9286, 1558-2523. DOI: 10.1109/TAC.2015.2512043. [Online]. Available: http://ieeexplore.ieee.org/document/7364172/.

[176] N. Bastianello, L. Madden, R. Carli, and E. Dall'Anese, "A Stochastic Operator Framework for Optimization and Learning With Sub-Weibull Errors," en, *IEEE Transactions on Automatic Control*, pp. 1–16, 2024, ISSN: 0018-9286, 1558-2523, 2334-3303. DOI: 10.1109/TAC.2024.3419186. [Online]. Available: https://ieeexplore.ieee.org/document/10572296/.

[177] P. L. Combettes and J.-C. Pesquet, "Stochastic Quasi-Fejér Block-Coordinate Fixed Point Iterations with Random Sweeping," en, *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 1221–1248, Jan. 2015, ISSN: 1052-6234, 1095-7189. DOI: 10.1137/140971233. [Online]. Available: http://epubs.siam.org/doi/10.1137/140971233.

[178] M. Bravo and R. Cominetti, "Stochastic Fixed-Point Iterations for Nonexpansive Maps: Convergence and Error Bounds," *SIAM Journal on Control and Optimization*, vol. 62, no. 1, pp. 191–219, Feb. 2024, Publisher: Society for Industrial and Applied Mathematics, ISSN: 0363-0129. DOI: 10.1137/22M1515550. [Online]. Available: https://epubs.siam.org/doi/10.1137/22M1515550.

[179] F. E. Browder, "NONEXPANSIVE NONLINEAR OPERATORS IN A BANACH SPACE," en, *Proceedings of the National Academy of Sciences*, vol. 54, no. 4, pp. 1041–1044, Oct. 1965, ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.54.4.1041. [Online]. Available: https://pnas.org/doi/full/10.1073/pnas.54.4.1041.

[180] W. A. Kirk, "A Fixed Point Theorem for Mappings which do not Increase Distances," en, *The American Mathematical Monthly*, vol. 72, no. 9, p. 1004, Nov. 1965, ISSN: 00029890. DOI: 10.2307/2313345. [Online]. Available: https://www.jstor.org/stable/2313345?origin=crossref.

[181] E. Sontag and Y. Wang, "Lyapunov Characterizations of Input to Output Stability," en, *SIAM Journal on Control and Optimization*, vol. 39, no. 1, pp. 226–249, Jan. 2000, ISSN: 0363-0129, 1095-7138. DOI: 10.1137/S0363012999350213. [Online]. Available: http://epubs.siam.org/doi/10.1137/S0363012999350213.

[182] H. K. Khalil, *Nonlinear systems*, en, 3rd ed. Upper Saddle River, N.J: Prentice Hall, 2002, ISBN: 978-0-13-067389-3.

[183] H. H. Bauschke and S. G. Kruk, "Reflection-Projection Method for Convex Feasibility Problems with an Obtuse Cone," en, *Journal of Optimization Theory and Applications*, vol. 120, no. 3, pp. 503–531, Mar. 2004, ISSN: 1573-2878. DOI: 10.1023/B:JOTA.0000025708.31430.22. [Online]. Available: https://doi.org/10.1023/B:JOTA.0000025708.31430.22.

[184] W. Tang and P. Daoutidis, "Data-Driven Control: Overview and Perspectives," in *2022 American Control Conference (ACC)*, ISSN: 2378-5861, Jun. 2022, pp. 1048–1064. DOI: 10.23919/ACC53348.2022.9867266. [Online]. Available: https://ieeexplore.ieee.org/document/9867266/.