Time-Varying Optimization and Its Application to Power System Operation

Thesis by
Yujie Tang

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2019
Defended January 4, 2019

Yujie Tang
ORCID: 0000-0002-4921-8372

ACKNOWLEDGEMENTS

My years at Caltech have truly been a precious experience and have shaped my research style. Foremost, I would like to express my sincere gratitude to my advisor, Professor Steven Low, for his generous support and invaluable guidance through my studies and research. Steven has always inspired me by his passion, intelligence, patience, and hardworking attitude. His philosophy of research has greatly influenced my thoughts and is something I will always admire and try to emulate in my future career.

I would also like to thank Professor Adam Wierman, Professor Venkat Chandrasekaran, and Professor Babak Hassibi for being my thesis committee members and providing brilliant comments and suggestions.

My sincere thanks also go to Professor Emiliano Dall'Anese at University of Colorado Boulder and Andrey Bernstein at National Renewable Energy Laboratory for their strong support of my research. It has always been a great pleasure to work with them.

I greatly appreciate the support from current and previous members of Netlab. My early years at Caltech would not have been smooth without the help from Lingwen Gan, Changhong Zhao, Qiuyu Peng, and Niangjun Chen. Discussions with Daniel Guo and John Pang have always been enlightening and valuable. It is also pleasing to see younger generations of Netlab making great contributions to state-of-the-art research, stimulating me to gain more confidence and diligence.

Finally, I would like to thank my parents Yongcheng Tang and Diqiu Yu for their unconditional love and support throughout my life.

ABSTRACT

The main topic of this thesis is time-varying optimization, which studies algorithms that can track optimal trajectories of optimization problems that evolve with time. A typical time-varying optimization algorithm is implemented in a running fashion in the sense that the underlying optimization problem is updated during the iterations of the algorithm, and is especially suitable for optimizing large-scale fast varying systems. Motivated by applications in power system operation, we propose and analyze first-order and second-order running algorithms for time-varying nonconvex optimization problems.

The first-order algorithm we propose is the regularized proximal primal-dual gradient algorithm, and we develop a comprehensive theory on its tracking performance. Specifically, we provide analytical results in terms of tracking a KKT point, and derive bounds for the tracking error defined as the distance between the algorithmic iterates and a KKT trajectory. We then provide sufficient conditions under which there exists a set of algorithmic parameters that guarantee that the tracking error bound holds. Qualitatively, the sufficient conditions for the existence of feasible parameters suggest that the problem should be "sufficiently convex" around a KKT trajectory to overcome the nonlinearity of the nonconvex constraints. The study of feasible algorithmic parameters motivates us to analyze the continuous-time limit of the discrete-time algorithm, which we formulate as a system of differential inclusions; results on its tracking performance as well as feasible and optimal algorithmic parameters are also derived. Finally, we derive conditions under which the KKT points for a given time instant will always be isolated so that bifurcations or merging of KKT trajectories do not happen.

The second-order algorithms we develop are approximate Newton methods that incorporate second-order information. We first propose the approximate Newton method for a special case where there are no explicit inequality or equality constraints. It is shown that good estimation of second-order information is important for achieving satisfactory tracking performance. We also propose a specific version of the approximate Newton method based on L-BFGS-B that handles box constraints. Then, we propose two variants of the approximate Newton method that handle explicit inequality and equality constraints. The first variant employs penalty functions to obtain a modified version of the original problem, so that the approximate Newton method for the special case can be applied. The second variant can

be viewed as an extension of the sequential quadratic program in the time-varying setting.

Finally, we discuss application of the proposed algorithms to power system operation. We formulate the time-varying optimal power flow problem, and introduce partition of the decision variables that enables us to model the power system by an implicit power flow map. The implicit power flow map allows us to incorporate real-time feedback measurements naturally in the algorithm. The use of real-time feedback measurement is a central idea in real-time optimal power flow algorithms, as it helps reduce the computation burden and potentially improve robustness against model mismatch. We then present in detail two real-time optimal power flow algorithms, one based on the regularized proximal primal-dual gradient algorithm, and the other based on the approximate Newton method with the penalty approach.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Y. Tang, E. Dall'Anese, A. Berstein, and S. Low. Running primal-dual gradient method for time-varying nonconvex problems, 2018, arXiv:1812.00613. URL https://arxiv.org/abs/1812.00613.
Y. Tang participated in formulating the problem and proposing the algorithm, derived the theorerical results, prepared the simulation, and participated in the writing of the manuscript.

[2] Y. Tang, E. Dall'Anese, A. Berstein, and S. H. Low. A feedback-based regularized primal-dual gradient method for time-varying nonconvex optimization. In *Proceedings of the 57th IEEE Conference on Decision and Control (CDC)*, pages 3244–3250, Miami Beach, FL, USA, Dec. 2018. doi:10.1109/CDC.2018.8619225.
Y. Tang participated in formulating the problem, proposing the algorithm, and deriving the theoretical results, prepared the simulation, and participated in the writing of the manuscript.

[3] Y. Tang, K. Dvijotham, and S. Low. Real-time optimal power flow. *IEEE Transactions on Smart Grid*, 8(6):2963–2973, 2017. doi:10.1109/TSG.2017.2704922.
Y. Tang participated in formulating the problem, proposed and analyzed the algorithm, prepared the simulation, and participated in the writing of the manuscript.

[4] Y. Tang and S. Low. Distributed algorithm for time-varying optimal power flow. In *Proceedings of the 56th IEEE Conference on Decision and Control (CDC)*, pages 3264–3270, Melbourne, VIC, Australia, Dec. 2017. doi:10.1109/CDC.2017.8264138.
Y. Tang participated in formulating the problem, proposed the distributed algorithm, prepared the simulation, and participated in the writing of the manuscript.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Overview of Time-Varying Optimization

Suppose we are given a physical or logical system, and for each time $t \in [0, T]$, the optimal operation of the system can be modeled as the following optimization problem:

$$\begin{aligned} \min_{x} \quad & c(x,t) \\ \text{s.t.} \quad & f(x,t) \le 0. \end{aligned} \tag{1.1}$$

Here $x \in \mathbb{R}^n$ is the decision variable that will be applied for operating the system, $c(x,t)$ is the objective function, and $f(x,t)$ gives the constraints at time t. It can be seen that (1.1) gives an optimization problem that evolves with time. We focus on the situations where the maps $x \mapsto c(x,t)$ and $x \mapsto f(x,t)$ will only be revealed at time t and their prior predictions are not available. This is a typical setting of a time-varying optimization problem.

In most situations where digital computers are used to solve this problem, we discretize the period $[0, T]$ by a sequence $(t_\tau)_{\tau=1}^K$ satisfying $0 < t_1 < \ldots < t_K \le T$, and obtain the following sequence of sampled problems:

$$\begin{aligned} \min_{x} \quad & c_\tau(x) \\ \text{s.t.} \quad & f_\tau(x) \le 0, \end{aligned} \tag{1.2}$$

where $\tau \in \{1, \ldots, K\}$ labels the discrete time index, and c_τ and f_τ denote the sampled versions $c(\cdot, t_\tau)$ and $f(\cdot, t_\tau)$. Traditionally, each instance of (1.2) is solved in the *batch* scheme abstracted as follows:

For each $\tau = 1, 2, \ldots, K$,

1. Collect problem data $\mathsf{D}(t_\tau)$ at $t = t_\tau$, and construct the initial iterate z_τ^0.
2. Repeat
$$z_\tau^k = \mathsf{T}\left(z_\tau^{k-1}; \mathsf{D}(t_\tau)\right) \tag{1.3}$$
for each $k = 1, 2, \ldots$ until z_τ^k convergences to some z_τ^∞.
3. Let $x_\tau^* = \Pi z_\tau^\infty$, and apply x_τ^* to the system of interest.

Here the operator T represents a single iteration of some iterative optimization algorithm, z^k denotes the intermediate iterate, $\mathsf{D}(t)$ denotes the problem data (objective function and constraints) at time t, and Π is a canonical projection map that extracts an applicable solution from the intermediate iterate. In order for the batch scheme to work smoothly, the iterations (1.3) for each τ should converge before the next sampling instant $t_{\tau+1}$ arrives. However, when each sampling interval $t_{\tau+1} - t_\tau$ needs to be very small to fully capture fast-varying costs and constraints and closely approximate the continuous-time optimal trajectory, the batch scheme may not be appropriate as the iterations (1.3) may fail to converge within the interval $(t_\tau, t_{\tau+1})$. The batch scheme may also break down if each instance of (1.2) is a large-scale problem or the optimization procedure requires heavy communication over a network in a distributed setting, as limited computation resources or high computational complexity may prevent the iterations (1.3) from converging within the interval $(t_\tau, t_{\tau+1})$.

The operation and control of smart grids is one such example. Technological advances have continuously reduced the cost of sustainable energy, and it is anticipated that future smart grids will incorporate a large number of distributed energy resources, including renewable generations such as solar panels and wind turbines, as well as small-scale, distributed devices with controllable power injections such as smart inverters, smart appliances, electric vehicles, and distributed energy storage to name a few. On the one hand, renewable generations introduce hard-to-predict fluctuations and uncertainties into the power network, making the operation and control of smart grids challenging. On the other hand, the increasing penetration of distributed controllable devices can provide diverse control capabilities that can be potentially utilized to overcome these challenges. In addition, extensive real-time measurement data will become available through the installation of smart meters and other advanced measurement equipment.

The optimal operation of smart grids can be formulated as a time-varying optimal power flow (OPF) problem, which is generally nonconvex. There is extensive literature on traditional OPF algorithms that solve each instance of (1.2) in the batch scheme; see the survey papers [45, 46] and the tutorial papers [65, 66]. However, these traditional algorithms are computationally burdensome if one needs to handle the fast fluctuations and uncertainties introduced by renewable generations, considering that the number of distributed controllable devices will be large in future smart grids. This motivates us to develop novel technologies for the operation and control of smart grids.

The operation of smart grids is not the only case where batch solutions can be difficult to obtain or inappropriate for application due to the time-varying nature of the problems. Similar situations can also occur in communication networks [29, 67], robotic networks [25], social networks [8], sparse signal recovery [6, 9], online learning [68, 96], economics [39], etc.

In time-varying optimization, this issue is resolved by designing algorithms that can be implemented in a *running* fashion, in the sense that the underlying optimization problem changes during the iterations of the algorithm. In other words, the problem data will be constantly updated regardless of whether the iterations converge to an optimal solution, and a sub-optimal solution can be extracted from the iterations at any time when necessary.

Throughout the thesis we consider the situation where each sampling instant t_τ is equal to $\tau\Delta$ for some $\Delta > 0$; in other words, we discretize $[0, T]$ by a uniform sampling interval Δ. Then, a running time-varying optimization algorithm for solving (1.1) can be described by the following procedure:

For each $\tau = 1, 2, \ldots, \lfloor T/\Delta \rfloor$,

1. Collect problem data $\mathsf{D}(\tau\Delta)$ at $t = \tau\Delta$.
2. Compute
$$\begin{aligned} \hat{z}_\tau &= \mathsf{T}(\hat{z}_{\tau-1}; \mathsf{D}(\tau\Delta)), \\ \hat{x}_\tau &= \Pi \hat{z}_\tau. \end{aligned} \tag{1.4}$$
3. Apply $\hat{x}_\tau$ to the system of interest.

Here $\hat{z}_\tau \in \mathbb{R}^d$ denotes the intermediate iterate, $\mathsf{D}(t) \in \mathbb{R}^p$ represents the problem data (parameters that describe the time-varying objective function, constraints, etc.) at time t. The map $\mathsf{T} : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ computes the new iterate from the previous iterate and the problem data, and $\Pi : \mathbb{R}^d \to \mathbb{R}^n$ is a canonical projection map that extracts an applicable solution from the intermediate iterate. The operator T has the following features:

1. The computation of T is relatively inexpensive so that (1.4) can be finished within the interval $(\tau\Delta, (\tau + 1)\Delta)$.
2. Suppose $X^*(t)$ is the set of global optimal solutions to (1.1) at time t which is nonempty. Let T_t denote the map $x \mapsto \mathsf{T}(x; \mathsf{D}(t))$. Then there exist an open

subset $U_t \subseteq \mathbb{R}^d$ with $X^*(t) \subset \Pi[U_t]$ and a set $K_t \supseteq X^*(t)$ with

$$\sup_{x\in K_t} \inf_{x'\in X^*(t)} \|x - x'\| < +\infty,$$

such that for any $z \in U_t$,

$$\Pi \circ \mathsf{T}_t^k(z) \in K_t$$

eventually as $k \to \infty$, where

$$\mathsf{T}_t^k := \underbrace{\mathsf{T}_t \circ \cdots \circ \mathsf{T}_t}_{k \text{ times}}.$$

Roughly speaking, this feature ensures that, if we fix t and run the iteration T_t in the batch scheme from a sufficiently good initial point, then in the long run we will get a good sub-optimal solution to (1.1) at time t. In other words, the iteration (1.4) is able to at least properly handle static problems.

In applications, the iteration (1.4) is carried out immediately after the problem data $\mathsf{D}(\tau\Delta)$ has been collected, and once a single iteration (1.4) is finished, the solution $\hat{x}_\tau$ will be immediately applied to the real world, and one prepares to collect the new problem data at time $t = (\tau + 1)\Delta$. While the solution $\hat{x}_\tau$ is in general only sub-optimal, the problem data is updated frequently to keep pace with the time-varying problem, so that the resulting solutions $\hat{x}_\tau$ will be able to *track* the optimal trajectory.

The simplest time-varying optimization algorithm is perhaps the running gradient descent algorithm for unconstrained time-varying optimization problems, whose iterations are given by

$$\hat{x}_\tau = \hat{x}_{\tau-1} - \alpha\nabla c_\tau(\hat{x}_{\tau-1}).$$

One can readily recognize that this is exactly a single iteration of the gradient descent algorithm for static optimization problems. In fact, many time-varying optimization algorithms are developed in a similar way, where the operator T resembles a single iteration of some existing iterative algorithm for static optimization, as the operator T constructed in such manner will be very likely to possess the aforementioned two features.

Let's look at a toy example that illustrates the advantages of employing running time-varying algorithms over batch solutions. Consider the following time-varying optimization problem

$$\min_{x\in\mathbb{R}^2} \quad c(x,t) = \frac{1}{2}\begin{bmatrix} x_1 - \cos t \\ x_2 - \sin t \end{bmatrix}^T \begin{bmatrix} 3 - 2\cos 2t & 2\sin 2t \\ 2\sin 2t & 3 + 2\cos 2t \end{bmatrix} \begin{bmatrix} x_1 - \cos t \\ x_2 - \sin t \end{bmatrix}$$

for $t \in [0, 2\pi]$, which models the optimization of some fictitious system that is time-varying. It is easy to recognize that the problem is quadratic and convex for each t and the trajectory of optimal solution is given by

$$x^*(t) = \begin{bmatrix} \cos t \\ \sin t \end{bmatrix}.$$

Let us consider two strategies for solving this problem:

1. The batch scheme: The gradient descent algorithm is employed. Computation of one gradient $\nabla_x c(x, t)$ takes a fixed amount of time $\pi/200$. Starting from $t = 0$, we update the problem data, then run the gradient descent algorithm until the ℓ_∞ norm of the gradient is less than 10^{-3}. Immediately after the iteration has converged, we apply the resulting solution to the fictitious system, increase τ by 1, update the problem data, and restart the iterations with the initial point being the previous solution that has just been calculated. For the fictitious system, the applied setpoint does not change until the next solution arrives.

2. The running scheme: The running gradient descent algorithm is employed, and computation of one gradient $\nabla_x c(x, t)$ takes the same fixed amount of time $\pi/200$. After one iteration has been carried out, we immediately apply the iterate to the fictitious system as a sub-optimal solution, increase τ by 1, update the problem data, and compute the next iteration. For the fictitious system, the applied setpoint does not change until the next solution arrives.

The initial point at $t = 0$ is $(1.01, 0)$, and the step size is $1/3$ for both schemes. We assume that apart from the delays caused by the gradient computation, there are no other delays or time spent during the procedure. The two schemes will provide two solution trajectories; we denote the trajectory generated by the batch scheme by $x^{\mathrm{b}}(t)$, and denote the trajectory generated by the running scheme by $x^{\mathrm{r}}(t)$. They are step functions over $t \in [0, 2\pi]$ as can be seen from the setting.

Figure 1.1 shows the curves of the distances to the optimal trajectory $\|x^{\mathrm{b}}(t) - x^*(t)\|$ and $\|x^{\mathrm{r}}(t) - x^*(t)\|$, and the objective value differences $c(x^{\mathrm{b}}(t), t) - c(x^*(t), t)$ and $c(x^{\mathrm{r}}(t), t) - c(x^*(t), t)$. It is apparent that the running scheme achieves much better performance in terms of *tracking* the time-varying optimal solution. In the batch scheme, it takes quite some time for the iterations to converge, and when a batch solution is applied to the system, the optimization problem has already changed considerably, meaning that the applied solution is not up-to-date. But in the running scheme, since we do not wait for the iterations to converge, we can update the

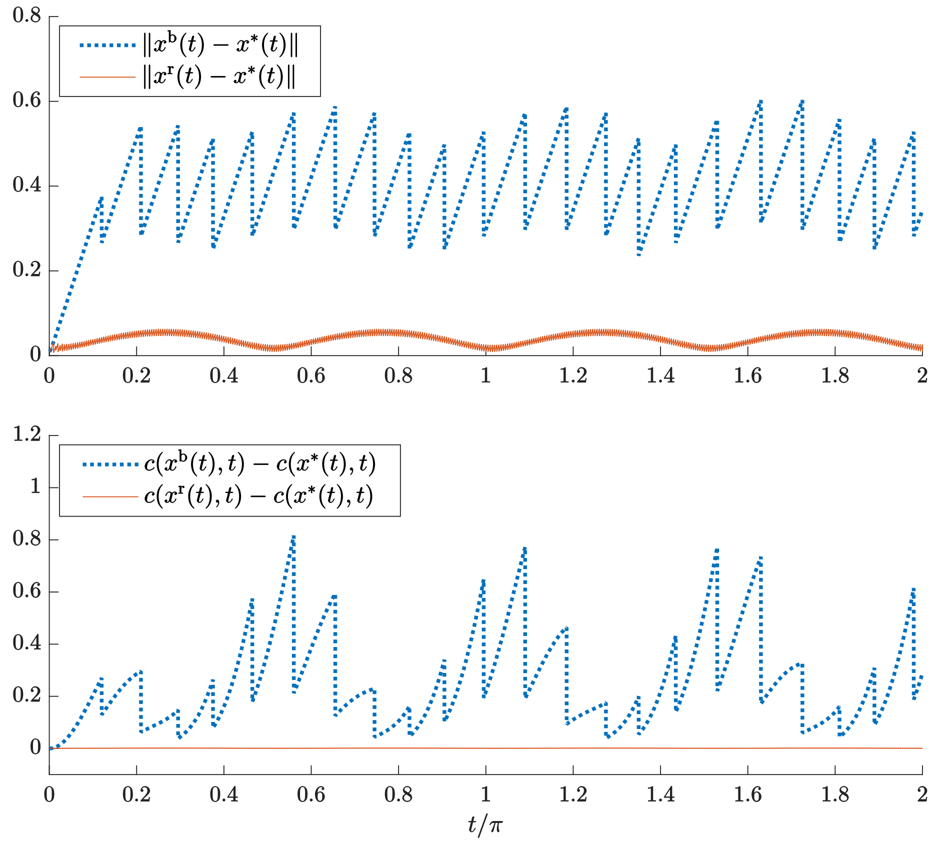

Figure 1.1: The distances to the optimal trajectory $\|x^{\mathrm{b}}(t)-x^*(t)\|$ and $\|x^{\mathrm{r}}(t)-x^*(t)\|$, and the objective value differences $c(x^{\mathrm{b}}(t),t) - c(x^*(t),t)$ and $c(x^{\mathrm{r}}(t),t) - c(x^*(t),t)$ of the two optimization schemes.

problem data much more frequently, and the resulting setpoints are able to keep track of the fast varying problem. While this toy example has a much simplified setting compared to practical scenarios, it demonstrates the benefits of time-varying optimization algorithms that can become significant in appropriate situations.

Performance Evaluation of Time-Varying Optimization Algorithms

As previously discussed, the solutions provided by time-varying optimization algorithms are in general not optimal; instead one focuses on whether and how accurately the solutions can *track* the optimal solution trajectory. *Tracking performance* is central in evaluating time-varying optimization algorithms.

Let us consider the sampled problem (1.2), and denote $\left(x^*_\tau\right)_\tau$ as a sequence of its (local) optimal solutions[1]. Suppose we run some time-varying optimization

[1] We choose x^*_τ arbitrarily when there are multiple local optimal solutions to (1.2).

algorithm and obtain a sequence of solutions denoted by $(\hat{x}_\tau)_\tau$. So far in existing literature, three types of quantities have been proposed as metrics for tracking performance with respect to $\left(x_\tau^*\right)_\tau$.

1. The distance to the optimal solution x_τ^*

$$e_\tau := \left\|\hat{x}_\tau - x_\tau^*\right\|,$$

   where the norm can be arbitrary. When there are explicit equality or inequality constraints and the algorithm also generates dual iterates, we can also evaluate the distance between the primal-dual pairs. This metric has been analyzed in [64, 77, 84, 86–88, 95] for specific time-varying optimization algorithms and employed in [9, 29, 35, 36, 90] for specific applications.

2. The sub-optimality in terms of objective values. Comparison of objective values is standard for evaluating performance of online learning algorithms, and can be also employed in time-varying optimization. There can be two sub-categories in this type of metric:

   a) The difference in objective values

$$e_\tau := c_\tau(\hat{x}_\tau) - c_\tau(x_\tau^*).$$

   The notion of dynamic regret in online learning is closely related to this metric [48, 56, 68, 96, 99].

   b) The ratio of objective values

$$r_\tau := \frac{c_\tau(\hat{x}_\tau)}{c_\tau(x_\tau^*)} \qquad \text{or} \qquad r := \frac{\sum_\tau c_\tau(\hat{x}_\tau)}{\sum_\tau c_\tau(x_\tau^*)}.$$

   Competitive ratio in online convex optimization can be viewed as a variant of this metric [4, 30, 63]. This metric can be employed in situations where the relative gap to the optimal objective value is more relevant than the absolute gap. On the other hand, one usually needs strong assumptions on the objective function to achieve a bounded competitive ratio.

   When the iterates $\hat{x}_\tau$ generated by the algorithm do not strictly satisfy the constraints $f_\tau(\hat{x}_\tau) \leq 0$, we can use some meric function

$$\phi_\tau(x) = c_\tau(x) + \sum_i g\left([f_{\tau,i}(x)]_+\right)$$

   instead of the objective function c_τ, where $[\cdot]_+$ denotes the positive part of a scalar, and $g : \mathbb{R} \to \mathbb{R}_+$ is a non-decreasing function with $g(x) = 0$ for $x \leq 0$.

3. The fixed-point residual
$$e_\tau := \|\hat{x}_\tau - G_\tau(\hat{x}_\tau)\|,$$
where $G_\tau : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous map such that any fixed point of G_τ is an optimal solution (local or global) to (1.2) at time τ. This metric has been introduced and analyzed in [84].

We mention that, in existing literature, researchers have also proposed to compare $\hat{x}_\tau$ with an optimal solution of the problem instance at the next time step $\tau + 1$; in other words, metrics based on

$$\left\|\hat{x}_\tau - x^*_{\tau+1}\right\|, \qquad c_{\tau+1}\left(\hat{x}_\tau\right) - c_{\tau+1}\left(x^*_{\tau+1}\right), \qquad \frac{c_{\tau+1}(\hat{x}_\tau)}{c_{\tau+1}\left(x^*_{\tau+1}\right)}, \qquad \|\hat{x}_\tau - G_{\tau+1}(\hat{x}_\tau)\|$$

have also been proposed. It turns out that it doesn't matter much whether we choose to compare $\hat{x}_\tau$ with x^*_τ or $x^*_{\tau+1}$ in the time-varying optimization setting[2]; results on either one of the choices can usually be transferred to results on the other choice under appropriate conditions.

The three types of metrics are all reasonable quantitative characterizations of the tracking performance mathematically, and depending on specific scenarios where the time-varying optimization algorithms are applied, one can choose different metrics that are more suitable for application. In this thesis, we mostly use the first metric, the distance to the optimal trajectory, as the metric for tracking performance.

1.2 Review of Existing Works

In this section we give a brief and non-exhaustive review of some existing works on time-varying optimization and related topics.

Reference [77] is one of the early papers that consider time-varying optimization problems with a similar setting to the one in this thesis. The paper derived a tracking error bound of the running gradient descent algorithm for unconstrained time-varying convex optimization. Specifically, the paper showed that

$$\limsup_{\tau\to\infty} \left\|\hat{x}_\tau - x^*_\tau\right\| \leq \frac{\rho}{1-\rho} \sup_\tau \left\|x^*_\tau - x^*_{\tau-1}\right\| \tag{1.5}$$

for the running gradient descent algorithm for time-varying unconstrained strongly convex problems, where

$$\Lambda_{\min} = \inf_{\tau,x} \lambda_{\min}\left(\nabla^2 c_\tau(x)\right), \qquad \Lambda_{\max} = \sup_{\tau,x} \lambda_{\max}\left(\nabla^2 c_\tau(x)\right),$$

[2] It does matter, however, in online learning where the actual loss incurred is $c_{\tau+1}(\hat{x}_\tau)$.

$$\rho = \max\{|1 - \alpha\Lambda_{\min}|, |1 - \alpha\Lambda_{\max}|\},$$

and $\alpha \in (0, 2/\Lambda_{\max})$ is the step size. The paper also considered a special case where the tracking error $\|\hat{x}_\tau - x^*_\tau\|$ tends to zero as $\tau \to \infty$. The tracking error bound (1.5) turns out to be one of the most fundamental results in time-varying optimization.

Recent years have witnessed considerable advances in the theory and algorithms of time-varying convex optimization. [64] proposed a decentralized algorithm based on Alternating Direction Method of Multipliers for time-varying unconstrained consensus problems, and also derived a tracking error bound that is similar to (1.5). [86] proposed a running algorithm based on the consensus + innovations method for time-varying constrained consensus problems, and also showed a similar tracking error bound. [87] proposed and analyzed the double smoothing method based on the multiuser optimization algorithm in [60], which is one of the earliest time-varying optimization algorithms that treat explicit inequality constraints. [95] introduced auxiliary variables in the design of the running algorithm for time-varying unconstrained consensus problems over directed networks, and showed convergence to a bounded tracking error. [84] proposed a unified framework for time-varying convex optimization using averaged operators, and derived tracking error bounds for several running algorithms. [16] proposed and analyzed a feedback-based time-varying optimization algorithm based on the primal-dual gradient method, which applies to logical or physical systems with feedback measurements. [34] proposed and analyzed a feedback-based method to regulate the output of a linear time-invariant dynamical system to the optimal solution of a time-varying convex optimization problem. [54] proposed a formulation of time-varying projected gradient dynamics by introducing the notion of temporal tangent cones, and showed existence of solution to the resulting differential equations. [15] developed an algorithmic framework for tracking fixed points of time-varying contraction mappings, and derived tracking error results in the situations where communication delays and packet drops lead to asynchronous algorithmic updates.

We would like to further expand on the results of [84] and [87]. In [84], the author considered the case where the time-varying optimization algorithm can be abstracted as

$$\hat{x}_\tau = T_\tau(\hat{x}_{\tau-1}),$$

where for each τ, $T_\tau : \mathbb{R}^n \to \mathbb{R}^n$ can be represented as

$$T_\tau(x) = (1 - a_\tau)x + a_\tau G_\tau(x), \qquad x \in \mathbb{R}^n$$

for some $a_\tau > 0$ and some nonexpansive operator G_τ (i.e., $\|G_\tau(x_1) - G_\tau(x_2)\| \leq \|x_1 - x_2\|$), and the fixed point of the operator T_τ gives the optimal solution x_τ^*. The paper showed that, if $\sigma := \sup_\tau \|x_\tau^* - x_{\tau-1}^*\| < +\infty$ and $X := \sup_\tau \sup_{x \in \mathbb{R}^n} T_\tau(x) < +\infty$, then the sequence $(\hat{x}_\tau)_\tau$ satisfies

$$\frac{1}{K} \sum_{\tau=1}^{K} \frac{1 - a_\tau}{a_\tau} \|T_\tau(\hat{x}_{\tau-1}) - \hat{x}_{\tau-1}\|^2 \leq \frac{1}{K} \left\|\hat{x}_0 - x_1^*\right\|^2 + \sigma(4X + \sigma).$$

In addition, in the special case where T_τ is a contraction mapping with a uniform contraction coefficient $\rho \in (0, 1)$, it was shown that

$$\left\|\hat{x}_{\tau-1} - x_\tau^*\right\| \leq \rho^{\tau-1} \|\hat{x}_0 - x_1^*\| + \frac{1 - \rho^{\tau-1}}{1 - \rho} \sigma.$$

The paper then applied this bound to several time-varying optimization algorithms. Especially, a bound on the fixed-point residual was established for the running projected gradient descent method that solves

$$\min_{x \in \mathcal{X}(t)} c(x, t),$$

where $\mathcal{X}(t) \subset \mathbb{R}^n$ is compact and uniformly bounded, and $c(\cdot, t)$ is convex and uniformly strongly smooth for each t.

In [87], the authors considered the time-varying convex problem

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & c_\tau(x) \\ \text{s.t.} \quad & f_\tau(x) \leq 0, \end{aligned}$$

where $c_\tau : \mathbb{R}^n \to \mathbb{R}$ is convex, $f_\tau : \mathbb{R}^n \to \mathbb{R}^m$ has convex components, and $\mathcal{X} \subseteq \mathbb{R}^n$ is convex and closed. The paper proposed a running version of the double smoothing algorithm [60] given by

$$\begin{aligned} \hat{x}_\tau &= \mathcal{P}_{\mathcal{X}} \left[\hat{x}_{\tau-1} - \alpha \nabla_x \mathcal{L}_\tau^{\nu,\epsilon}(\hat{x}_{\tau-1}, \hat{\lambda}_{\tau-1})\right], \\ \hat{\lambda}_\tau &= \mathcal{P}_{\mathbb{R}^m} \left[\hat{\lambda}_{\tau-1} + \alpha \nabla_\lambda \mathcal{L}_\tau^{\nu,\epsilon}(\hat{x}_{\tau-1}, \hat{\lambda}_{\tau-1})\right], \end{aligned}$$

where the regularized Lagrangian $\mathcal{L}_\tau^{\nu,\epsilon}$ is defined by

$$\mathcal{L}_\tau^{\nu,\epsilon}(x, \lambda) := c_\tau(x) + \lambda^T f_\tau(x) + \frac{\nu}{2}\|x\|^2 - \frac{\epsilon}{2}\|\lambda\|^2.$$

The paper showed that, when the step size α is sufficiently small, there exists some $\rho < 1$ such that

$$\limsup_{k \to \infty} \left\|\hat{z}_\tau - z_{\tau+1}^\star\right\| \leq \frac{1}{1 - \rho} \sup_\tau \left\|z_{\tau+1}^\star - z_\tau^\star\right\|,$$

where $\hat{z}_\tau = \left(\hat{x}_\tau, \hat{\lambda}_\tau\right)$, and $z^\star_\tau = \left(x^\star_\tau, \lambda^\star_\tau\right)$ denotes the saddle point of the regularized Lagrangian $\mathcal{L}^{\nu,\epsilon}_\tau(x, \lambda)$. The authors also considered how to implement the algorithm in a distribution manner when the cost and constraint functions have special structures.

Time-varying optimization is closely related to *parametric optimization* that has a long history [49, 73, 81]. The formulation of one-parametric optimization problems is almost the same as (1.1), where t is regarded as a one-dimensional parameter that may or may not represent time. Path-following methods (also called continuation methods or homotopy methods) and their varients have been developed to solve these problems [3, 39, 43, 80, 85, 97]. A typical parametric optimization algorithm discretizes the parameter set $[0, T]$ by $0 < t_1 < \ldots < t_K \leq T$, and generates approximate solutions $\hat{x}_\tau$ such that

$$e := \sup_\tau \|\hat{x}_\tau - x^*(t_\tau)\| = O\left(\Delta_{\max}\right), \quad \text{as } \Delta_{\max} \to 0,$$

where $\Delta_{\max} = \max_\tau |t_{\tau+1} - t_\tau|$. The theory of parametric optimization generally focuses on the convergence rate of the approximation error e as $\Delta_{\max} \to 0$. In addition, many parametric optimization algorithms consist of a predictor and a corrector: for each $\tau = 1, \ldots, K$, the corrector utilizes the problem data at $t = t_\tau$ to move the prediction produced by the previous iteration towards the optimal trajectory, and the predictor then estimates the tangent vector $dz^*(t)/dt$ at $t = t_\tau$ to provide a prediction of the optimal solution at the subsequent parameter value. This predictor-corrector procedure assumes knowledge of how the cost and constraint functions evolve with time, which is different from the time-varying optimization setting discussed in this thesis. The theory and algorithms of parametric optimization provide important insights on and tools for the study of time-varying optimization.

We would also like to mention that, in online learning, theories on dynamic regret have been developed, which are closely related to time-varying optimization [21, 48, 51, 56, 62, 68, 96, 99]. In online learning, for each time step τ, a player chooses a strategy $\hat{x}_\tau$ from some feasible set $\mathcal{X}$ by an online learning algorithm, and suffers some loss $c_{\tau+1}(\hat{x}_\tau)$ [3]. To evaluate the performance of the online learning algorithm in the time-varying setting, the dynamic regret is proposed that compares the cumulative losses of the online player with the losses of the best possible

[3] We label the time indices in a fashion that is different from online learning literature but similar to the time-varying optimization setting.

responses

$$\sum_{\tau=1}^{K} c_\tau\left(\hat{x}_{\tau-1}\right) - c_\tau\left(x_\tau^*\right).$$

Particularly, researchers are interested in bounding the growth rate of the dynamic regret as $K \to \infty$ over a specific class of loss functions, and usually focus on the situations where the dynamic regret achieves sublinear growth.

For example, the pioneering paper [99] formulated the online convex programming problem in which $\mathcal{X} \subseteq \mathbb{R}^n$ is convex and compact and c_τ is convex and differentiable for all τ. The paper proposed the online projected gradient descent algorithm whose iterations are given by

$$\hat{x}_\tau = \mathcal{P}_\mathcal{X}\left[\hat{x}_{\tau-1} - \alpha \nabla c_\tau(\hat{x}_{\tau-1})\right].$$

The paper showed that, for the online projected gradient descent algorithm, the dynamic regret can be upper bounded by

$$\sum_{\tau=1}^{K} c_\tau\left(\hat{x}_{\tau-1}\right) - c_\tau\left(x_\tau^*\right) \leq \frac{7R^2}{4\alpha} + \frac{R}{\alpha}\sum_{\tau=1}^{K-1}\left\|x_{\tau+1}^* - x_\tau^*\right\| + \frac{G\alpha}{2}K,$$

where

$$R := \sup_{x_1, x_2 \in \mathcal{X}} \|x_2 - x_1\|, \qquad G := \sup_{\tau}\sup_{x \in \mathcal{X}} \|\nabla c_\tau(x)\|.$$

In [68], the authors considered an online learning problem where $\mathcal{X} \subseteq \mathbb{R}^n$ is convex and compact and the loss function c_τ is uniformly strongly convex, i.e., there exists $\mu > 0$ such that

$$c_\tau(x_2) \geq c_\tau(x_1) + \nabla c_\tau(x_1)^T(x_2 - x_1) + \frac{\mu}{2}\|x_2 - x_1\|^2, \qquad \forall x_1, x_2 \in \mathcal{X}.$$

It was shown that an improved bound on the dynamic regret can be derived for the online projected gradient descent algorithm when the step size α is sufficiently small:

$$\sum_{\tau=1}^{K} c_\tau\left(\hat{x}_{\tau-1}\right) - c_\tau\left(x_\tau^*\right) \leq G\left(\frac{1}{1-\rho}\sum_{\tau=1}^{K-1}\left\|x_{\tau+1}^* - x_\tau^*\right\| + \frac{\|\hat{x}_0 - x_1^*\|}{1-\rho}\right),$$

where

$$\rho = \sqrt{1 - \alpha\mu}, \qquad G = \sup_{\tau}\sup_{x \in \mathcal{X}} \|\nabla c_\tau(x)\|.$$

A recent paper [48] considered an online learning problem where the loss function could be nonconvex. Specifically, while $\mathcal{X} \subseteq \mathbb{R}^n$ was still assumed to be convex and compact, each loss function $c_\tau : \mathcal{X} \to \mathbb{R}$ was assumed to be *weakly pseudo-convex*: there exists $M > 0$ such that

$$c_\tau(x) - c_\tau(x_\tau^*) \leq \begin{cases} M \dfrac{\nabla c_\tau(x)^T (x - x_\tau^*)}{\|\nabla c_\tau(x)\|}, & \nabla c_\tau(x) \neq 0, \\ 0, & \nabla c_\tau(x) = 0 \end{cases}$$

for any $x \in \mathcal{X}$ and any $x_\tau^* \in \arg\min_{u \in \mathcal{X}} c_\tau(u)$. The paper showed that, if the strategies $\hat{x}_\tau$ are generated by the *online normalized gradient descent* algorithm

$$\hat{x}_\tau = \mathcal{P}_\mathcal{X} \left[\hat{x}_{\tau-1} - \eta g_\tau \right], \qquad g_\tau := \begin{cases} \dfrac{\nabla c_\tau(\hat{x}_{\tau-1})}{\|\nabla c_\tau(\hat{x}_{\tau-1})\|}, & \nabla c_\tau(\hat{x}_{\tau-1}) \neq 0, \\ 0, & \nabla c_\tau(\hat{x}_{\tau-1}) = 0, \end{cases}$$

where η is some positive constant, then the following bound on the dynamic regret holds under certain conditions:

$$\sum_{\tau=1}^{K} c_\tau(\hat{x}_{\tau-1}) - c_\tau(x_\tau^*) \leq \frac{M}{2\eta} \left(4R^2 + \eta^2 K + 6R \sum_{\tau=1}^{K-1} \left\| x_{\tau+1}^* - x_\tau^* \right\| \right),$$

where $R := \sup_{x \in \mathcal{X}} \|x\|$. Further investigation of these examples and other related works suggests that, although time-varying optimization and online learning have different perspectives and settings, the mathematics behind the theories of dynamic regret can be very relevant for the research on time-varying optimization.

Time-varying optimization has also found its place in various applications. While the original formulation was for static problems, the network flow control algorithm in [67] is essentially implemented in a running fashion and can be easily extended to time-varying situations. In [29], the primal-dual saddle point dynamics was applied in time-varying wireless systems, with theoretical analysis on the tracking performance. In [9], the running version of the iterative soft-thresholding algorithm and its continuous-time counterpart for sparse signal recovery were analyzed under the time-varying setting. In [8], the authors proposed a running stochastic gradient descent method for topology tracking in social networks, though no theoretical guarantee on tracking performance has been provided.

With the technological advances in sustainable energy and smart grids, recent research has found it necessary to consider power system operation in the time-varying setting. Reference [24] proposed an online algorithm based on the dual subgradient ascent method, and [17, 47] proposed online algorithms based on the projected gradient descent method for the operation of distribution networks, though no theoretical results were developed for the time-varying setting. [35, 36] employed the double smoothing method [87] and a linearized power flow model for real-time operation of distribution networks, and [14] presented a more comprehensive framework that can handle a wider range of controllable power devices. [52, 55] proposed continuous-time online algorithms that employ projected gradient dynamics on the power flow manifold.

1.3 Organization of the Thesis

Chapter 2: First-Order Algorithms for Time-Varying Optimization

In Chapter 2, we propose a first-order time-varying optmization algorithm, which we call the *regularized proximal primal-dual gradient algorithm*. The regularization comes in the form of a strongly concave term in the dual vector variable that is added to the Lagrangian function [50, 58, 60]. The strongly concave regularization term plays a critical role in establishing contraction-like behavior of the proposed algorithm. However, as an artifact of this regularization, existing works for time-invariant convex programs [60], time-varying convex programs [16], and for static nonconvex problems [50] could prove that gradient-based iterative methods approach an approximate KKT point. On the other hand, in Chapter 2 we provide analytical results in terms of tracking a KKT point (as opposed to an approximate KKT point) of (1.2) and provide bounds for the distance of the algorithmic iterates from a KKT trajectory. The bounds are obtained by finding conditions under which the regularized proximal primal-dual gradient step exhibits a contraction-like behavior. The bounds are directly related to the maximum temporal variability of a KKT trajectory, and also depend on pertinent algorithmic parameters such as the step size and the regularization coefficient.

We then provide sufficient conditions for the existence of algorithmic parameters that guarantee bounded tracking error for sufficiently small sampling interval. From a qualitative standpoint, the sufficient conditions for the existence of feasible parameters suggest that the problem should be “sufficiently convex” around a KKT trajectory to overcome the nonlinearity of the nonconvex constraints.

The study of feasible algorithmic parameters suggests analyzing the continuous-time limit of the proposed regularized proximal primal-dual gradient algorithm; see, e.g., [31, 32, 78, 79, 93] and pertinent references therein for continuous-time algorithmic platforms. We show that the continuous-time counterpart of the discrete-time algorithm is given by a system of differential inclusions that can be viewed as a generalization of *perturbed sweeping processes* [2, 28]. The tracking performance of the system of differential inclusions is analytically established; the continuous-time tracking error bound shares a similar form with the discrete-time tracking error bound. Then, we provide sufficient conditions for the existence of feasible algorithmic parameters, and also analyze the existence and properties of the optimal algorithmic parameters that minimize the tracking error bound.

Finally, we derive conditions under which the KKT points for a given time will always be isolated; that is, bifurcations or merging of KKT trajectories do not happen.

While most existing works on time-varying optimization focus on problems that are globally convex, our study does not assume global convexity of the objective and constraint functions, which can be particularly useful for time-varying nonconvex problems such as the time-varying optimal power flow problem. Part of these results have been reported in [88, 89].

Chapter 3: Second-Order Algorithms for Time-Varying Optimization

In Chapter 3, we consider second-order algorithms for time-varying optimization. By "second-order" algorithms, we mean that for each τ, not only the current function value and gradient information is used, but also the exact or approximate curvature information is employed, which is similar to Newton and quasi-Newton methods in static optimization.

We first propose the *approximate Newton method* for a special case where the constraints in (1.2) appear in the form $x \in \mathcal{X}_\tau$ for some closed and convex $\mathcal{X}_\tau$. The approximate Newton method is a natural extension of (quasi-)Newton method to the time-varying setting. It is shown that the tracking error is directly affected by how well we can approximate the curvature of the objective function c_τ. We also propose a specific version of the approximate Newton method based on L-BFGS-B [26, 69] that handles box constraints.

Then, we propose two variants of the approximate Newton method that can handle nonlinear constraints that are formulated as explicit equalities and inequalities. The first variant employs penalty functions [19, 72] to obtain a modified version of the original problem, so that the approximate Newton method for the special case discussed above can be applied. We investigate the relationship between the penalty functions and regularization on the dual variables, and derive bounds on the difference between optimal solutions of the penalized problem and the original problem. The second variant can be viewed as an extension of sequential quadratic programming [23] in the time-varying setting, with regularization on the dual variable just like the regularized proximal primal-dual gradient algorithm. We perform a direct analysis of the tracking error with respect to the optimal trajectory and discuss its implications.

Finally, we use a toy example to compare first-order and second-order methods, in order to have a better understanding of how to appropriately choose between these two types of methods in practical scenarios.

As mentioned in Section 1.2, parametric optimization is closely related to time-varying optimization, and many path-following algorithms utilize curvature information in a very similar fashion to the approximate Newton methods [39, 43, 49, 97]. On the other hand, parametric optimization generally focuses on whether and how fast the trajectory generated by path-following algorithms will converge to the optimal trajectory as the sampling interval goes to zero. This is different from our setting where we assume a given and fixed sampling interval, considering that the delays caused by computation, communication, etc. will prevent the sampling interval from being arbitrarily small in practice.

Similar to Chapter 2, we do not explicitly assume global convexity for the time-varying optimization problem in our study. Part of these results have been reported in [90, 91].

Chapter 4: Applications in Power System Operation

In Chapter 4, we discuss the application of time-varying optimization algorithms to power system operation, motivated by the consideration that batch solutions are inappropriate when one needs to optimize over a large number of distributed controllable devices to handle fast-timescale fluctuations and uncertainties introduced by distributed energy resources in future smart grids.

We formulate the time-varying optimal power flow problem, where the decision variables are subject to the power flow equations. We partition the decision variables into two groups, and replace the power flow equations by an implicit power flow map between the two groups of variables. We present sufficient conditions for the existence of the implicit power flow map, and discuss the advantages of introducing implicit power flow map to utilize real-time feedback measurements. The use of real-time feedback measurements is a central idea in real-time optimal power flow algorithms [35, 36, 90, 91] (also see [47] for static optimization), which not only greatly reduces the computation burden, but also improves robustness against model mismatch.

Then we present two real-time optimal power flow algorithms in detail. The first algorithm applies the regularized primal-dual gradient algorithm to the real-time operation of a distribution feeder. Specifically, we show how real-time measurement data can be naturally incorporated in the computation, and analyze the effect of employing approximate Jacobian of the implicit power flow map in the algorithm theoretically. The second algorithm applies the approximate Newton method with penalty functions to the time-varying optimal power flow problem. Second-order real-time optimal power flow method was first proposed by [90] and here we present the distributed implementation proposed in [91].

Chapter 5: Concluding Remarks on Future Directions

We conclude the thesis by remarks on some future directions that are worth exploration.

1.4 Notations and Terminologies

Sets and Functions

For any set A, its power set will be denoted by 2^A, and the Cartesian product $\underbrace{A \times \cdots \times A}_{n \text{ times}}$ will be denoted by A^n.

Let $(A_i)_{i=1}^n$ be a finite sequence of sets, and let S be a nonempty subset of $A_1 \times \cdots \times A_n$. A map $\pi : S \to \bigcup_{i=1}^n A_i$ is called a canonical projection if there exists $j \in \{1, \ldots, n\}$ such that

$$\pi(x_1, \ldots, x_n) = x_j, \qquad \forall (x_1, \ldots, x_n) \in S.$$

We denote $\mathbb{R}_+ = [0, +\infty)$ and $\mathbb{R}_{++} = (0, +\infty)$.

Suppose $f : X \to Y$ where X and Y are any sets. For any subset $A \subseteq Y$, $f^{-1}[A]$ denotes the preimage $\{x \in X : f(x) \in A\}$.

For a real-valued function f defined on an interval $D \subseteq \mathbb{R}$, we say that it is nondecreasing if $\forall x_1, x_2 \in D$, $x_1 < x_2 \Rightarrow f(x_1) \leq f(x_2)$, and say that it is (strictly) increasing if the inequality is strict. The notion of nonincreasing and (strictly) decreasing functions are defined similarly. We say that f is unimodal if there exists $x_0 \in D$ such that f is strictly decreasing on $D \cap (-\infty, x_0]$ and is strictly increasing on $D \cap [x_0, +\infty)$.

For $f : \mathbb{R}^n \to \mathbb{R}^m$, the inequality $f(x) \leq 0$ means $-f(x) \in \mathbb{R}_+^m$.

For any $x \in \mathbb{R}$, $\lfloor x \rfloor$ denotes the largest integer n such that $n \le x$, and $\lceil x \rceil$ denotes the smallest integer n such that $n \ge x$.

Linear Algebra

We use

$$x = (x_1, \ldots, x_n) = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

to denote vector x in $\mathbb{R}^n$. The Euclidean norm on $\mathbb{R}^n$ will be simply denoted by

$$\|x\| := \sqrt{\sum_{i=1}^{n} x_i^2}, \qquad x \in \mathbb{R}^n.$$

The identity matrix will be denoted by I, or $I_n \in \mathbb{R}^{n\times n}$ when the dimension needs to be specified to avoid confusion.

For any matrix $M = \left(M_{ij}\right) \in \mathbb{R}^{m\times n}$. $\|M\|$ denotes the operator norm

$$\|M\| := \sup_{x\neq 0} \frac{\|Mx\|}{\|x\|}.$$

When M is a square matrix, we use $\lambda_{\max}(M)$ and $\lambda_{\min}(M)$ to denote the largest and the smallest eigenvalues of M respectively.

For two real symmetric matrix A and B, the expression $A \preceq B$ means $B - A$ is positive semidefinite, and $A \prec B$ means $B - A$ is positive definite.

Real Analysis

For a topological space X and a subset $A \subseteq X$, the interior of A will be denoted by $\operatorname{int} A$, and the closure of A will be denoted by $\operatorname{cl} A$.

The closed unit ball in $\mathbb{R}^n$ centered at the origin will be denoted by $\mathcal{B}_n$. For any two subsets A, B of $\mathbb{R}^n$ and any $\alpha, \beta \in \mathbb{R}$, we define

$$\alpha A + \beta B := \{\alpha x + \beta y : x \in A, y \in B\}.$$

As an example, $\{x\} + r\mathcal{B}_n$ is the closed ball in $\mathbb{R}^n$ of radius r centered at x.

Suppose $f : X \to \mathbb{R} \cup \{+\infty\}$ where X is a topological space. We say that f is lower semicontinuous if $\liminf_{n\to\infty} f(x_n) \ge f(x)$ for any $x \in X$ and any sequence $(x_n) \subset X$ that converges to x. Equivalently, f is lower semicontinuous if $\{x \in X : f(x) > a\}$ is an open subset of X for all $a \in \mathbb{R}$. A function $f : X \to \mathbb{R} \cup \{-\infty\}$

is upper semicontinuous if $-f$ is lower semicontinuous. A function $f : X \to \mathbb{R}$ is continuous if and only if it is both lower and upper semicontinuous. See [57, Section 12] for more details.

Let $f : I \to \mathbb{R}$ be a function defined on an interval I of $\mathbb{R}$. The essential supremum of $f(t)$ over $t \in I$ is defined by

$$\operatorname*{ess\,sup}_{t\in I} f(t) := \inf\left\{M \in \mathbb{R} : \text{Leb}\left(f^{-1}[(M, +\infty)]\right) = 0\right\},$$

where Leb denotes the Lebesgue measure on $\mathbb{R}$.

Let $f : [a, b] \to \mathbb{R}^m$ be a vector-valued function whose components are absolutely continuous. Then there exists a Lebegue integrable function from $[a, b]$ to $\mathbb{R}^m$, which we denote by Df, such that

$$f(t) - f(a) = \int_a^t Df(s)\,ds$$

for all $t \in [a, b]$, and Df is unique up to a Lebesgue null set. We will also denote $\frac{d}{dt}f(t) := Df(t)$. Furthermore, f is Lipschitz continuous on $[a, b]$ if and only if

$$\operatorname*{ess\,sup}_{t\in[0,T]} \|Df(t)\| < +\infty,$$

or in other words, $Df \in L^\infty([a, b])$, and in this case,

$$\sup_{\substack{t_1,t_2\in[a,b],\\ t_1\neq t_2}} \frac{\|f(t_2) - f(t_1)\|}{|t_2 - t_1|} = \operatorname*{ess\,sup}_{t\in[a,b]} \|Df(t)\|.$$

See [44, Section 3.5] for more details.

Differential Calculus

Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable on $\mathbb{R}^n$. The gradient of f at $x \in \mathbb{R}^n$ will be denoted by $\nabla f(x)$ which is an n-dimensional column vector. If f is twice differentiable, the Hessian of f at $x \in \mathbb{R}^n$ will be denoted by $\nabla^2 f(x)$.

Let $f(x, y)$ be a function from $\mathbb{R}^n \times Y$ to $\mathbb{R}$ where Y is any set, and for some $y_0 \in Y$, $f(x, y_0)$ is differentiable with respect to x. The (partial) gradient of $f(x, y)$ with respect to x at $(x_0, y_0) \in \mathbb{R}^n \times Y$ will be denoted by $\nabla_x f(x_0, y_0)$. If $f(x, y_0)$ is twice differentiable with respect to x, the (partial) Hessian of $f(x, y)$ with respect to x at $(x_0, y_0) \in \mathbb{R}^n \times Y$ will be denoted by $\nabla^2_{xx} f(x_0, y_0)$.

Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a differentiable vector-valued function. We will denote its components by $f_1, \ldots, f_m$. The Jacobian of f at $x \in \mathbb{R}^n$ will be denoted by

$$J_f(x) := \begin{bmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_n(x)^T \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

Let $f(x, y)$ be a vector-valued function from $\mathbb{R}^n \times Y$ to $\mathbb{R}^m$ where Y is any set. Suppose for some $y_0 \in Y$, $f(x, y_0)$ is differentiable with respect to x. The (partial) Jacobian of $f(x, y)$ with respect to x at $(x_0, y_0) \in \mathbb{R}^n \times Y$ will be denoted by

$$J_{f,x}(x_0, y_0) := \begin{bmatrix} \nabla_x f_1(x_0, y_0)^T \\ \vdots \\ \nabla_x f_m(x_0, y_0)^T \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

The following equalities can be shown by straightforward calculations.

Lemma 1.1. *1. Suppose $f : \mathbb{R}^n \to \mathbb{R}^m$ is continuously differentiable. Then for any $x, y \in \mathbb{R}^n$,*

$$f(y) = f(x) + \left(\int_0^1 J_f(x + \theta(y - x))\, d\theta\right)(y - x). \tag{1.6}$$

2. Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable. Then for any $x, y \in \mathbb{R}^n$,

$$\begin{aligned} f(y) &= f(x) + \nabla f(x)^T (y - x) \\ &\quad + \frac{1}{2}(y - x)^T \left(\int_0^1 2(1 - \theta)\nabla^2 f(x + \theta(y - x))\, d\theta\right)(y - x). \end{aligned} \tag{1.7}$$

Convex Analysis

Let $C \subseteq \mathbb{R}^n$ be a convex set. The indicator function of C will be defined and denoted by

$$I_C(x) = \begin{cases} 0, & x \in C, \\ +\infty, & x \notin C. \end{cases}$$

The relative interior of C will be denoted by $\operatorname{relint} C$. The normal cone of C at $x \in C$ is defined by

$$N_C(x) := \{y \in \mathbb{R}^n : y^T(z - x) \le 0, \ \forall z \in C\}.$$

When C is closed, for any $x \in \mathbb{R}^n$, it is known [20, Proposition 2.2.1] that there exists a unique vector that minimizes $\|y - x\|$ over all $y \in C$. We call this vector the projection of x into C, and denote it by

$$\mathcal{P}_C(x) := \arg\min_{y\in C} \|y - x\|.$$

Furthermore, we have the following proposition.

Proposition 1.1 ([20, Proposition 2.2.1])**.** *Suppose $C \subseteq \mathbb{R}^n$ is convex and closed.*

1. *For any $x \in \mathbb{R}^n$,*

$$y = \mathcal{P}_C(x) \quad \Longleftrightarrow \quad (x-y)^T(z-y) \le 0,\ \forall z \in C \quad \Longleftrightarrow \quad x - y \in N_C(y).$$

2. *$\mathcal{P}_C$ is nonexpansive, i.e., $\|\mathcal{P}_C(x) - \mathcal{P}_C(y)\| \le \|x - y\|$ for any $x, y \in \mathbb{R}^n$.*

For a convex cone $C \subseteq \mathbb{R}^n$, its polar cone is defined by

$$C^\circ := \{y \in \mathbb{R}^n : y^T z \le 0,\ \forall z \in C\}.$$

For any subset A of $\mathbb{R}^n$, its convex hull will be denoted by conv A, and its affine hull will be denoted by aff A.

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function. The domain of f is defined by

$$\mathrm{dom}(f) := \{x \in \mathbb{R}^n : f(x) < +\infty\}.$$

We say that f is proper if $\mathrm{dom}(f)$ is nonempty. We say that f is closed if the epigraph

$$\mathrm{epi}(f) := \{(x, y) \in \mathbb{R}^n \times \mathbb{R} : y \ge f(x)\}$$

is closed.

Proposition 1.2 ([82, Theorem 7.1])**.** *A proper convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is lower semicontinuous if and only if it is closed.*

The subdifferential of f at $x_0 \in \mathrm{dom}(f)$ is defined by

$$\partial f(x_0) := \left\{u \in \mathbb{R}^n : f(x) - f(x_0) \ge u^T(x - x_0),\ \forall x \in \mathbb{R}^n\right\}.$$

The definition indicates that $\partial f(x_0)$ is the intersection of a collection of closed half-spaces in $\mathbb{R}^n$, and therefore is convex and closed.

Suppose $g(x, y)$ is a function from $\mathbb{R}^n \times Y$ to $\mathbb{R} \cup \{+\infty\}$ where Y is any set, and for some fixed $y_0 \in Y$, $g(x, y_0)$ is a proper convex function of x. We denote the partial subdifferential of $g(x, y)$ with respect to x at $(x_0, y_0) \in \mathbb{R}^n \times Y$ by

$$\partial_x g(x_0, y_0) := \left\{u \in \mathbb{R}^n : g(x, y_0) - g(x_0, y_0) \ge u^T(x - x_0),\ \forall x \in \mathbb{R}^n\right\}.$$

Chapter 2

FIRST-ORDER ALGORITHMS FOR TIME-VARYING OPTIMIZATION

2.1 Problem Formulation

Let us consider the following time-varying optimization problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & c(x,t)+h(x,t) \\ \text{s.t.} \quad & f^c(x,t)+f^{nc}(x,t)\le 0, \\ & f^{eq}(x,t)=0. \end{aligned} \tag{2.1}$$

Here $t \in [0,T]$ labels time and $T > 0$ is a fixed constant that represents the length of the period we consider, $c : \mathbb{R}^n\times\mathbb{R} \to \mathbb{R}$, $h : \mathbb{R}^n\times\mathbb{R} \to \mathbb{R}\cup\{+\infty\}$, $f^c : \mathbb{R}^n\times\mathbb{R} \to \mathbb{R}^m$, $f^c : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ and $f^{eq} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^{m'}$. For each fixed $t \in [0,T]$, $c(\cdot,t)$, $f^c(\cdot,t)$, $f^{nc}(\cdot,t)$ and $f^{eq}(\cdot,t)$ are twice continuously differentiable, $f^c(\cdot,t)$ has convex components, and $h(\cdot,t)$ is a closed proper convex function with a closed domain. We denote the domain of $h(\cdot,t)$ by

$$\mathrm{dom}_t(h) := \{x \in \mathbb{R}^n : h(x,t) < +\infty\}.$$

In addition, we also assume that $\nabla^2_{xx}c(x,t)$, $\nabla^2_{xx}f^c_i(x,t)$, $\nabla^2_{xx}f^{nc}_i(x,t)$ and $\nabla^2_{xx}f^{eq}_j(x,t)$ for each $i = 1,\dots,m$ and $j = 1,\dots,m'$ are continuous over $\mathbb{R}^n \times [0,T]$. We let $f^{in} := f^c + f^{nc}$.

We assume that (2.1) is feasible for all $t \in [0,T]$, and that there exists a Lipschitz continuous trajectory of primal-dual pair $z^* = (x^*,\lambda^*,\mu^*) : [0,T] \to \mathbb{R}^n\times\mathbb{R}^m_+\times\mathbb{R}^{m'}$ such that for each $t \in [0,T]$, $z^*(t) = (x^*(t),\lambda^*(t),\mu^*(t))$ satisfies

$$(x^*(t),\lambda^*(t),\mu^*(t)) \in \mathrm{dom}_t(h)\times\mathbb{R}^m_+\times\mathbb{R}^{m'}, \tag{2.2a}$$

$$-\nabla_x c(x^*(t),t) - \begin{bmatrix} J_{f^{in},x}(x^*(t),t) \\ J_{f^{eq},x}(x^*(t),t) \end{bmatrix}^T \begin{bmatrix} \lambda^*(t) \\ \mu^*(t) \end{bmatrix} \in \partial_x h(x^*(t),t), \tag{2.2b}$$

$$f^{in}(x^*(t),t) \in N_{\mathbb{R}^m_+}(\lambda^*(t)), \tag{2.2c}$$

$$f^{eq}(x^*(t),t) = 0. \tag{2.2d}$$

The following proposition justifies (2.2) as KKT conditions for (2.1) under appropriate constraint qualification conditions.

Proposition 2.1. *Suppose $\bar{x}$ is a local minimum of the following optimization problem*

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c(x) + h(x) \\ s.t. \quad & f(x) \in C, \end{aligned} \tag{2.3}$$

where $c : \mathbb{R}^n \to \mathbb{R}$ and $f : \mathbb{R}^n \to \mathbb{R}^p$ are continuously differentiable, $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a closed proper convex function with a closed domain, and $C \subseteq \mathbb{R}^p$ is a closed convex cone. Suppose the following constraint qualification condition is satisfied:

$$\text{There is no } \lambda \in C^\circ \backslash \{0\} \text{ such that } \lambda^T f(\bar{x}) = 0 \text{ and } -J_f(\bar{x})^T \lambda \in N_{\mathrm{dom}(h)}(\bar{x}). \tag{2.4}$$

Then there exists $\bar{\lambda} \in C^\circ$ such that

$$-\nabla c(\bar{x}) - J_f(\bar{x})^T \bar{\lambda} \in \partial h(\bar{x}), \tag{2.5a}$$

$$f(\bar{x}) \in N_{C^\circ}(\bar{\lambda}). \tag{2.5b}$$

The set of optimal Lagrange multipliers

$$\Lambda = \left\{ \bar{\lambda} \in C^\circ : \bar{\lambda} \text{ satisfies } (2.5) \right\}$$

is convex and compact.

The proof is given in Appendix 2.A for completeness.

Remark 2.1. The constraint qualification in Proposition (2.4) can be viewed as a generalization of the Mangasarian-Fromovitz constraint qualification (MFCQ) [19]. Indeed, in the case where $h = 0$ and $C = -\mathbb{R}^m_+ \times \{0\}^{m'}$, we can prove that MFCQ is equivalent to (2.4). Let $\mathcal{I} \subseteq \{1, \ldots, m\}$ denote the index set of active constraints at $\bar{x}$, and denote $\mathcal{E} = \{m+1, \ldots, m'\}$.

1. Suppose MFCQ holds at $\bar{x}$, and $u \in \mathbb{R}^n$ satisfies $J_{f_\mathcal{I}}(\bar{x})u < 0$ and $J_{f_\mathcal{E}}(\bar{x})u = 0$. Suppose $\lambda \in \mathbb{R}^m_+ \times \mathbb{R}^{m'}$ satisfies $\lambda^T f(\bar{x}) = 0$ and $J_f(\bar{x})^T \lambda = 0$. Then we have $\lambda_i = 0$ for $i \notin \mathcal{I} \cup \mathcal{E}$, and

$$0 = \lambda^T J_f(\bar{x})u = \lambda_\mathcal{I}^T J_{f_\mathcal{I}}(\bar{x})u + \lambda_\mathcal{E}^T J_{f_\mathcal{E}}(\bar{x})u = \lambda_\mathcal{I}^T J_{f_\mathcal{I}}(\bar{x})u.$$

   Considering that $J_{f_\mathcal{I}}(\bar{x})u < 0$ while $\lambda_\mathcal{I} \geq 0$, we get $\lambda_\mathcal{I} = 0$. Thus $J_f(\bar{x})^T \lambda = J_{f_\mathcal{E}}(\bar{x})^T \lambda_\mathcal{E} = 0$. MFCQ requires that $J_{f_\mathcal{E}}(\bar{x})$ has linearly independent row vectors, which implies that $\lambda_\mathcal{E} = 0$. Now we see that $\lambda = 0$. Therefore MFCQ implies the constraint qualification (2.4).

2. Suppose that MFCQ does not hold at $\bar{x}$. If the row vectors of $J_{f_{\mathcal{E}}}(\bar{x})$ are linearly dependent, then there exists $\mu \in \mathbb{R}^{m'}\backslash\{0\}$ such that $J_{f_{\mathcal{E}}}(\bar{x})^T\mu = 0$, and (2.4) is obviously violated. If there is no $u \in \mathbb{R}^n$ satisfying $J_{f_{\mathcal{I}}}(\bar{x})u < 0$ and $J_{f_{\mathcal{E}}}(\bar{x})u = 0$, we let

$$S_1 = \left\{ \begin{bmatrix} J_{f_{\mathcal{I}}}(\bar{x}) \\ J_{f_{\mathcal{E}}}(\bar{x}) \end{bmatrix} u : u \in \mathbb{R}^n \right\}, \qquad S_2 = \left\{ (z, 0) \in \mathbb{R}^{|\mathcal{I}|} \times \mathbb{R}^{m'} : z \in -\mathbb{R}^{|\mathcal{I}|}_{++} \right\}.$$

Obviously S_1 and S_2 are convex sets, and since $S_1 \cap S_2 = \emptyset$, by the separating hyperplane theorem, there exists some nonzero $(v_1, v_2) \in \mathbb{R}^{|\mathcal{I}|} \times \mathbb{R}^{m'}$ such that $v_1^T z \geq 0$ for all $z \in -\mathbb{R}^{|\mathcal{I}|}_{++}$, and

$$v_1^T J_{f_{\mathcal{I}}}(\bar{x})u + v_2^T J_{f_{\mathcal{E}}}(\bar{x})u \leq 0, \quad \forall u \in \mathbb{R}^n.$$

By setting $u = J_{f_{\mathcal{I}}}(\bar{x})^T v_1 + J_{f_{\mathcal{E}}}(\bar{x})^T v_2$ in the above inequality, we see that $J_{f_{\mathcal{I}}}(\bar{x})^T v_1 + J_{f_{\mathcal{E}}}(\bar{x})^T v_2 = 0$. On the other hand, $v_1^T z \geq 0$ for all $z \in -\mathbb{R}^{|\mathcal{I}|}_{++}$ implies $v_1 \geq 0$. We then see that if we define $\lambda \in \mathbb{R}^m_+ \times \mathbb{R}^{m'}$ by $\lambda_{\mathcal{I}} = v_1$, $\lambda_{\mathcal{E}} = v_2$ and $\lambda_i = 0$ for any $i \notin \mathcal{I} \cup \mathcal{E}$, the constraint qualification (2.4) is violated by λ.

We now see that the constraint qualification (2.4) is equivalent to MFCQ. ■

Remark 2.2. In general, there can be multiple KKT points of (2.1) that move in $\mathbb{R}^n \times \mathbb{R}^m_+ \times \mathbb{R}^{m'}$ as time proceeds, which form multiple trajectories that can appear, terminate, bifurcate or merge during the period $(0, T)$. Reference [49] presented a very comprehensive theory of the structures and singularities of trajectories of KKT points for time-varying optimization problems. In [39], the authors show that strong regularity for generalized equations is a key concept for establishing the existence of Lipschitz continuous KKT trajectories over a given finite period. Here, we arbitrarily select one of these trajectories that is well defined and Lipschitz continuous for $t \in [0, T]$, denote it by $z^*(t)$, and mainly focus on this trajectory in most part of our study. ■

As a special case of (2.1), if we set $h(x, t)$ to be the indicator function

$$h(x,t) = \begin{cases} 0, & x \in \mathcal{X}(t), \\ +\infty, & \text{otherwise}, \end{cases} \tag{2.6}$$

where $\mathcal{X}(t) \subset \mathbb{R}^n$ is convex and compact for each $t \in [0, T]$, then (2.1) includes the following problem as a special case

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & c(x,t) \\ \text{s.t.} \quad & f^c(x,t) + f^{nc}(x,t) \le 0, \\ & f^{eq}(x,t) = 0, \\ & x \in \mathcal{X}(t), \end{aligned} \tag{2.7}$$

and in this case, the subdifferential $\partial_x h(x,t)$ is the normal cone $N_{\mathcal{X}(t)}(x)$ as can be seen by

$$\begin{aligned} y \in \partial_x h(x,t) \quad & \Longleftrightarrow \quad h(z,t) \ge h(x,t) + y^T(z-x), \quad \forall z \in \mathbb{R}^n \\ & \Longleftrightarrow \quad 0 \ge y^T(z-x), \quad \forall z \in \mathcal{X}(t) \\ & \Longleftrightarrow \quad y \in N_{\mathcal{X}(t)}(x). \end{aligned}$$

Proximal Operator

We define the proximal operator to a convex function $g : \mathbb{R}^k \to \mathbb{R} \cup \{+\infty\}$ by

$$\operatorname{prox}_g(x) := \operatorname*{arg\,min}_{y\in\mathbb{R}^k} \; g(y) + \frac{1}{2}\|y-x\|^2. \tag{2.8}$$

The proximal operator can be viewed as a generalization of projection onto convex sets. Indeed, if we take g to be the indicator function of a convex set $C \subseteq \mathbb{R}^k$, the proximal operator is just the projection onto C.

The following lemma summarizes the properties of the proximal operator that will be frequently used in subsequent sections.

Lemma 2.1 ([74]). *Let $g : \mathbb{R}^k \to \mathbb{R} \cup \{+\infty\}$ be convex and proper. Then*

1. $\operatorname{prox}_g(x)$ *is well-defined for all $x \in \mathbb{R}^k$.*
2. $y = \operatorname{prox}_g(x)$ *if and only if $x - y \in \partial g(y)$.*
3. prox_g *is nonexpansive: for any $x, y \in \mathbb{R}^k$, $\left\|\operatorname{prox}_g(x) - \operatorname{prox}_g(y)\right\| \le \|x-y\|$.*

We can use the second property in Lemma 2.1 to rewrite two of the KKT conditions (2.2b) and (2.2c) equivalently in the form of a fixed-point equation

$$x^*(t) = \operatorname{prox}_{\alpha h(\cdot,t)}\left[x^*(t) - \alpha\left(\nabla_x c(x^*(t),t) + \begin{bmatrix} J_{f^{in},x}(x^*(t),t) \\ J_{f^{eq},x}(x^*(t),t)\end{bmatrix}^T \begin{bmatrix} \lambda^*(t) \\ \mu^*(t)\end{bmatrix}\right)\right], \tag{2.9a}$$

$$\lambda^*(t) = \mathcal{P}_{\mathbb{R}^m_+}\left[\lambda^*(t) + \beta f^{in}(x^*(t),t)\right], \tag{2.9b}$$

where α and β are any positive real number.

2.2 Regularized Proximal Primal-Dual Gradient Algorithm

We first discretize the time domain $[0, T]$ so that discrete-time algorithms can be proposed and implemented. Let $\Delta > 0$ be the sampling interval. Let $\mathcal{T} := \{1, 2, \ldots, \lfloor T/\Delta \rfloor\}$ be the set of discrete time indices, and we denote

$$\begin{aligned} c_\tau(x) &= c(x, \tau\Delta), & h_\tau(x) &= h(x, \tau\Delta), \\ f_\tau^c(x) &= f^c(x, \tau\Delta), & f_\tau^{nc}(x) &= f^{nc}(x, \tau\Delta), \\ f_\tau^{in}(x) &= f^{in}(x, \tau\Delta) & f_\tau^{eq}(x) &= f^{eq}(x, \tau\Delta), \end{aligned}$$

for each $\tau \in \mathcal{T}$. The sampled version of the KKT trajectory $z^*(t)$ will be denoted by $z_\tau^* = (x_\tau^*, \lambda_\tau^*, \mu_\tau^*)$ for $\tau \in \mathcal{T}$.

Let $\hat{z}_0 = (\hat{x}_0, \hat{\lambda}_0, \hat{\mu}_0) \in \mathrm{dom}_0(h) \times \mathbb{R}_+^m \times \mathbb{R}^{m'}$ be the initial point. The regularized proximal primal-dual gradient algorithm produces a primal-dual pair $(\hat{x}_\tau, \hat{\lambda}_\tau, \hat{\mu}_\tau)$ iteratively by

$$\hat{x}_\tau = \mathrm{prox}_{\alpha h_\tau}\left[\hat{x}_{\tau-1} - \alpha\left(\nabla c_\tau(\hat{x}_{\tau-1}) + \begin{bmatrix} J_{f_\tau^{in}}(\hat{x}_{\tau-1}) \\ J_{f_\tau^{eq}}(\hat{x}_{\tau-1}) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_{\tau-1} \\ \hat{\mu}_{\tau-1} \end{bmatrix}\right)\right], \tag{2.10a}$$

$$\hat{\lambda}_\tau = \mathcal{P}_{\mathbb{R}_+^m}\left[\hat{\lambda}_{\tau-1} + \eta\alpha\left(f_\tau^{in}(\hat{x}_{\tau-1}) - \epsilon\big(\hat{\lambda}_{\tau-1} - \lambda_{\mathrm{prior}}\big)\right)\right], \tag{2.10b}$$

$$\hat{\mu}_\tau = \hat{\mu}_{\tau-1} + \eta\alpha\left(f_\tau^{eq}(\hat{x}_{\tau-1}) - \epsilon\big(\hat{\mu}_{\tau-1} - \mu_{\mathrm{prior}}\big)\right) \tag{2.10c}$$

for each $\tau \in \mathcal{T}$. Here $\alpha > 0$, $\eta > 0$, $\epsilon > 0$, $\lambda_{\mathrm{prior}} \in \mathbb{R}_+^m$ and $\mu_{\mathrm{prior}} \in \mathbb{R}^{m'}$ are parameters of the algorithm.

The regularized proximal primal-dual gradient algorithm is closely related to the following saddle point problem

$$\min_{x \in \mathbb{R}^n} \max_{\lambda \in \mathbb{R}_+^m, \mu \in \mathbb{R}^{m'}} \mathcal{L}_\tau^\epsilon(x, \lambda, \mu), \tag{2.11}$$

where $\mathcal{L}_\tau^\epsilon(x, \lambda, \mu)$ denotes the regularized Lagrangian defined by

$$\begin{aligned} \mathcal{L}_\tau^\epsilon(x, \lambda, \mu) = {} & c_\tau(x) + h_\tau(x) + \lambda^T f_\tau^{in}(x) + \mu^T f_\tau^{eq}(x) \\ & - \frac{\epsilon}{2}\left(\left\|\lambda - \lambda_{\mathrm{prior}}\right\|^2 + \left\|\mu - \mu_{\mathrm{prior}}\right\|^2\right). \end{aligned} \tag{2.12}$$

We can see that there is an additional term $-\frac{\epsilon}{2}\left(\left\|\lambda - \lambda_{\mathrm{prior}}\right\|^2 + \left\|\mu - \mu_{\mathrm{prior}}\right\|^2\right)$ compared to the original Lagrangian function, which represents regularization that drives the dual variables towards $\big(\lambda_{\mathrm{prior}}, \mu_{\mathrm{prior}}\big)$. Then, (2.10) can be viewed as applying a single iteration of proximal gradient descent on the primal variable and projected gradient ascent on the dual variables in (2.12), with step sizes being α for the primal update and $\eta\alpha$ for the dual updates respectively. The parameter ϵ then controls the amount of regularization on the dual variables. The constant vectors λ_{prior} and μ_{prior} can be viewed as prior estimates of the optimal dual variables, and can be set to zero if such prior estimates cannot be obtained[1].

It can be seen that the regularized Lagrangian (2.12) is strongly concave with respect to the dual variables, which potentially improves the convergence behavior of primal-dual gradient methods. On the other hand, even if we solve the saddle point problem (2.11) exactly, the resulting solution will be in general different from the KKT points of (2.1), which suggests that there could be further sub-optimality introduced by regularization in the time-varying setting. This is indeed the case as will be seen in Section 2.3.

We denote the primal-dual pair produced by (2.10) as $\hat{z}_\tau = (\hat{x}_\tau, \hat{\lambda}_\tau, \hat{\mu}_\tau)$. We define the norm

$$\|z\|_\eta := \sqrt{\|x\|^2 + \eta^{-1}\|\lambda\|^2 + \eta^{-1}\|\nu\|^2}$$

for $z = (x, \lambda, \nu) \in \mathbb{R}^{n+m+m'}$.

2.3 Tracking Performance

The regularized proximal primal-dual gradient algorithm proposed in the previous section is expected to produce a sequence of primal-dual pairs that are sufficiently close to the KKT points for each time instant; in other words, the algorithm should be able to *track* the KKT points $z^*_\tau = (x^*_\tau, \lambda^*_\tau, \mu^*_\tau)$ for each τ. In this section, we study the tracking performance of the regularized proximal primal-dual gradient algorithm.

We define the *tracking error* to be

$$\left\|\hat{z}_\tau - z^*_\tau\right\|_\eta = \sqrt{\|\hat{x}_\tau - x^*_\tau\|^2 + \eta^{-1}\left\|\hat{\lambda}_\tau - \lambda^*_\tau\right\|^2 + \eta^{-1}\left\|\hat{\mu}_\tau - \mu^*_\tau\right\|^2},$$

which represents the distance between the KKT point and the solution produced by (2.10); a small tracking error implies good tracking performance with respect to $\left(z^*_\tau\right)_\tau$. We are interested in factors that affect the tracking error, and especially conditions under which a bounded tracking error can be guaranteed.

[1] The dual update (2.10b) can also be equivalently written as

$$\hat{\lambda}_\tau = \mathcal{P}_{\mathbb{R}^m_+}\left[\hat{\lambda}_{\tau-1} + \eta\alpha\left(\tilde{f}^{in}_\tau(\hat{x}_{\tau-1}) - \epsilon\hat{\lambda}_{\tau-1}\right)\right]$$

with $\tilde{f}^{in}_\tau(x) = f^{in}_\tau(x) + \epsilon\lambda_{\text{prior}}$. In other words, we can also view (2.10b) as employing no prior estimate of λ^*_τ but a more conservative version of the inequality constraint given by $f^{in}_\tau(x) + \epsilon\lambda_{\text{prior}} \le 0$.

Before proceeding, we first define some quantities that will be used in our analysis. Let

$$\sigma_\eta := \sup_{\substack{t_1,t_2\in[0,T],\\ t_1\neq t_2}} \frac{\|z^*(t_2)-z^*(t_1)\|_\eta}{|t_2-t_1|} = \operatorname*{ess\,sup}_{t\in[0,T]} \left\|\frac{d}{dt}z^*(t)\right\|_\eta,$$

i.e., the maximum speed of the KKT point with respect to the norm $\|\cdot\|_\eta$; hence $\left\|z^*_\tau - z^*_{\tau-1}\right\|_\eta \leq \sigma_\eta\Delta$ for each τ. We assume that $\sigma_\eta > 0$ for some (and thus for any) $\eta > 0$. Intuitively, the slower $z^*(t)$ moves, the more likely it is to obtain a smaller tracking error. We then define

$$M_d := \sup_{t\in[0,T]} \left\|\begin{bmatrix}\lambda^*(t)-\lambda_{\text{prior}}\\ \mu^*(t)-\mu_{\text{prior}}\end{bmatrix}\right\|, \tag{2.13}$$

$$M_{nc}(\delta) := \sup_{t\in[0,T]}\sup_{u:\|u\|\leq\delta} \left\|D^2_{xx}\begin{bmatrix}f^{nc}(x^*(t)+u,t)\\ f^{eq}(x^*(t)+u,t)\end{bmatrix}\right\|, \tag{2.14}$$

$$M_c(\delta) := \sup_{t\in[0,T]}\sup_{u:\|u\|\leq\delta} \left\|D^2_{xx}f^c(x^*(t)+u,t)\right\|, \tag{2.15}$$

$$L_f(\delta) := \sup_{t\in[0,T]}\sup_{u:\|u\|\leq\delta} \left\|\begin{bmatrix}J_{f^{in},x}(x^*(t)+u,t)\\ J_{f^{eq},x}(x^*(t)+u,t)\end{bmatrix}\right\|, \tag{2.16}$$

$$D(\delta,\eta) := \sqrt{\eta}L_f(\delta) + M_c(\delta)\sup_{t\in[0,T]}\|\lambda^*(t)\|. \tag{2.17}$$

Here for a vector-valued $f(x,t)$ that is twice continuously differentiable in x for a fixed t, we use $D^2_{xx}f(x,t)$ to denote the bilinear map that maps a pair of vectors (h_1,h_2) to a vector whose i'th entry is given by $h_2^T\nabla^2_{xx}f_i(x,t)h_1$, i.e.,

$$D^2_{xx}f(x,t):\quad (h_1,h_2)\mapsto \left(h_2^T\nabla^2_{xx}f_i(x,t)h_1\right)_i,$$

and $\left\|D^2_{xx}f(x,t)\right\|$ is defined by

$$\left\|D^2_{xx}f(x,t)\right\| := \sup_{h_1,h_2\neq 0}\frac{\left\|D^2_{xx}f(x,t)(h_1,h_2)\right\|}{\|h_1\|\|h_2\|} = \sup_{\|h_1\|=\|h_2\|=1}\left\|D^2_{xx}f(x,t)(h_1,h_2)\right\|.$$

Intuitively, $\left\|D^2_{xx}f(x,t)\right\|$ characterizes the nonlinearity of function f with respect to x at time t. Specifically, we have the following lemma, whose proof is given in Appendix 2.A.

Lemma 2.2. *Let $t\in[0,T]$ be fixed. Then for any x_1, x_2,*

$$\begin{aligned}&\left\|J_{f,x}(x_2,t)-J_{f,x}(x_1,t)\right\|\\ &\leq \|x_2-x_1\|\cdot\sup_{\theta\in[0,1]}\left\|D^2_{xx}f(x_1+\theta(x_2-x_1),t)\right\|,\end{aligned} \tag{2.18}$$

and

$$\begin{aligned} &\left\| f(x_2,t) - f(x_1,t) - J_{f,x}(x_1,t)(x_2 - x_1) \right\| \\ \le\ & \frac{1}{2}\|x_2 - x_1\|^2 \cdot \sup_{\theta\in[0,1]} \left\| D^2_{xx} f(x_1 + \theta(x_2 - x_1), t) \right\|. \end{aligned} \tag{2.19}$$

The argument $\delta \in (0,+\infty)$ in the definitions (2.14)–(2.17) represents the radius of the ball centered at $x^*(t)$, i.e., the local region around $x^*(t)$ we are interested in.

Let the "nonconvex Lagrangian component" be

$$\begin{aligned} \mathcal{L}^{nc}(x,\lambda,\mu,t) &:= c(x,t) + \lambda^T f^{nc}(x,t) + \mu^T f^{eq}(x,t), \\ \mathcal{L}^{nc}_\tau(x,\lambda,\mu) &:= \mathcal{L}^{nc}(x,\lambda,\mu,\tau\Delta). \end{aligned}$$

We also define

$$\overline{H}_{\mathcal{L}^{nc}}(u,t) := \int_0^1 \nabla^2_{xx}\mathcal{L}^{nc}(x^*(t) + \theta u, \lambda^*(t), \mu^*(t), t)\, d\theta, \tag{2.20}$$

$$\overline{H}_{f_i^c}(u,t) := \int_0^1 2(1-\theta)\nabla^2_{xx} f_i^c(x^*(t) + \theta u, t)\, d\theta, \tag{2.21}$$

and

$$\rho^{(\mathrm{P})}(\delta,\alpha,\eta,\epsilon) := \sup_{\substack{t\in[0,T],\\ u:\|u\|\le\delta}} \left\| \left(I - \alpha\overline{H}_{\mathcal{L}^{nc}}(u,t)\right)^2 - \alpha(1-\eta\alpha\epsilon)\sum_{i=1}^m \lambda_i^*(t)\overline{H}_{f_i^c}(u,t) \right\|,$$

$$\begin{aligned} \rho(\delta,\alpha,\eta,\epsilon) := \Bigg[& \max\left\{\rho^{(\mathrm{P})}(\delta,\alpha,\eta,\epsilon), (1-\eta\alpha\epsilon)^2\right\} + \alpha(1-\eta\alpha\epsilon)\frac{\sqrt{\eta}\delta M_{nc}(\delta)}{2} \\ & + \alpha^2\left(2\sup_{\substack{t\in[0,T],\\ u:\|u\|\le\delta}} \left\|\eta\epsilon I - \overline{H}_{\mathcal{L}^{nc}}(u,t)\right\| D(\delta,\eta) + D^2(\delta,\eta)\right)\Bigg]^{1/2}, \end{aligned}$$

$$\kappa(\delta,\alpha,\eta,\epsilon) := \max\left\{1,\ \frac{1-\eta\alpha\epsilon}{\rho(\delta,\alpha,\eta,\epsilon)},\ \frac{\sqrt{\eta}\alpha L_f(\delta)}{\rho(\delta,\alpha,\eta,\epsilon)}\right\}.$$

Here, $\overline{H}_{\mathcal{L}^{nc}}(u,t)$ is the averaged Hessian matrix of the "nonconvex Lagrangian component" around $x^*(t)$ along the vector u, and $\overline{H}_{f_i^c}(u,t)$ is the averaged Hessian matrix of the convex part of the i'th constraint along the vector u. The quantity $\rho(\delta,\alpha,\eta,\epsilon)$, as we will show later, can be viewed as the contraction coefficient of a single step of the algorithm.

Remark 2.3. Although the algorithm (2.10) runs in the discrete time domain, the definitions above take supremum over the continuous time domain $[0,T]$, as some of these quantities will be employed again to study the continuous-time limit of the algorithm (2.10) in Section 2.4. The results in Theorem 2.1 will still follow if we replace all the supremum over $t \in [0, T]$ by supremum over $t \in \tau\mathcal{T}$ in the definitions above. ■

The following lemma is critical in establishing the tracking error bound of the algorithm (2.10), whose proof will be presented in Appendix 2.A.

Lemma 2.3. *Let $\tau \in \mathcal{T}$ be arbitrary. If $\left\|\hat{z}_{\tau-1} - z_\tau^*\right\|_\eta \leq \delta$ and $\hat{z}_\tau$ is generated by* (2.10)*, then*

$$\left\|\hat{z}_\tau - z_\tau^*\right\|_\eta \leq \rho(\delta, \alpha, \eta, \epsilon) \left\|\hat{z}_{\tau-1} - z_\tau^*\right\|_\eta + \kappa(\delta, \alpha, \eta, \epsilon)\sqrt{\eta}\alpha\epsilon M_d, \tag{2.22}$$

where $\kappa(\delta, \alpha, \eta, \epsilon)$ is upper bounded by $\sqrt{2}$ and satisfies

$$\lim_{\alpha\to 0^+} \kappa(\delta, \alpha, \eta, \epsilon) = 1.$$

Now we present one of the main results of this chapter, which establishes the tracking error bound of the regularized proximal primal-dual gradient algorithm (2.10).

Theorem 2.1. *Suppose there exist $\delta > 0$, $\alpha > 0$, $\eta > 0$ and $\epsilon > 0$ such that*

$$\sigma_\eta \Delta \leq (1 - \rho(\delta, \alpha, \eta, \epsilon))\, \delta - \kappa(\delta, \alpha, \eta, \epsilon)\sqrt{\eta}\alpha\epsilon M_d. \tag{2.23a}$$

Let the initial point $\hat{z}_0 = (\hat{x}_0, \hat{\lambda}_0, \hat{\mu}_0)$ be sufficiently close to the KKT point z_1^* so that

$$\left\|\hat{z}_0 - z_1^*\right\|_\eta \leq \delta. \tag{2.23b}$$

Then the sequence $(\hat{z}_\tau)_{\tau\in\mathcal{T}}$ produced by the regularized proximal primal-dual gradient algorithm (2.10) *satisfies*

$$\begin{aligned}\left\|\hat{z}_\tau - z_\tau^*\right\|_\eta \leq{} & \frac{\rho(\delta, \alpha, \eta, \epsilon)\, \sigma_\eta\Delta + \kappa(\delta, \alpha, \eta, \epsilon)\sqrt{\eta}\alpha\epsilon M_d}{1 - \rho(\delta, \alpha, \eta, \epsilon)} \\ & + \rho^\tau(\delta, \alpha, \eta, \epsilon) \left(\left\|\hat{z}_0 - z_1^*\right\|_\eta - \frac{\sigma_\eta\Delta + \kappa(\delta, \alpha, \eta, \epsilon)\sqrt{\eta}\alpha\epsilon M_d}{1 - \rho(\delta, \alpha, \eta, \epsilon)}\right)\end{aligned} \tag{2.24}$$

for all $\tau \in \mathcal{T}$.

Moreover, we have

$$\lim_{\alpha\to 0^+} \kappa\,(\delta, \alpha, \eta, \epsilon) = 1 \qquad \textit{and} \qquad \kappa\,(\delta, \alpha, \eta, \epsilon) \leq \sqrt{2}.$$

Note that Lemma 2.3 asserts that if the radius δ is chosen such that $\rho := \rho(\delta, \alpha, \eta, \epsilon) < 1$, iteration (2.10) is an approximate local contraction with coefficient ρ and error term $e := \kappa(\delta, \alpha, \eta, \epsilon)\sqrt{\eta}\alpha\epsilon M_\lambda$. The idea of the proof of Theorem 2.1 is then based on showing that condition (2.23a) is sufficient to guarantee that the trajectory generated by the algorithm is confined within the contraction region at every time step. Note that (2.23a) implies

$$B(z_\tau^*, \rho\delta + e) \subseteq B(z_{\tau+1}^*, \delta), \tag{2.25}$$

where $B(z, \delta)$ is the ball centered at z with radius δ (with respect to the norm $\|\cdot\|_\eta$). Then, it is possible to show by induction that if $\hat{z}_0 \in B(z_1^*, \delta)$, then $\hat{z}_{\tau-1} \in B(z_\tau^*, \delta)$ for all τ. The idea of condition (2.25) is illustrated in Figure 2.1 and the formal proof of Theorem 2.1 is given below.

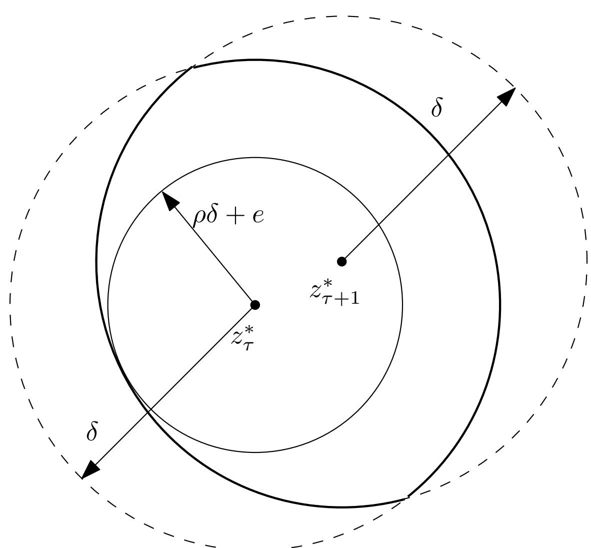

Figure 2.1: Illustration of condition (2.25). This condition is essentially on: (i) time-variability of a KKT point, (ii) extent of the contraction, and (iii) maximum error in each iteration. Note that in the error-less static case (i.e., $\sigma_\eta = e = 0$), this condition is trivially satisfied.

Proof of Theorem 2.1. For notational simplicity, we just use ρ to denote $\rho(\delta, \alpha, \eta, \epsilon)$ and use κ to denote $\kappa(\delta, \alpha, \eta, \epsilon)$. The condition (2.23b) guarantees that we can use Lemma 2.3 to get

$$\left\|\hat{z}_1 - z_1^*\right\|_\eta \leq \rho \left\|\hat{z}_0 - z_1^*\right\|_\eta + \kappa\sqrt{\eta}\alpha\epsilon M_d,$$

which shows that (2.24) holds for $\tau = 1$. Now suppose that (2.24) holds for some $\tau \in \mathcal{T}$. We have

$$
\begin{aligned}
&\left\|\hat{z}_\tau - z^*_{\tau+1}\right\|_\eta \leq \left\|\hat{z}_\tau - z^*_\tau\right\|_\eta + \left\|z^*_\tau - z^*_{\tau+1}\right\|_\eta \\
\leq\ & \frac{\rho\sigma_\eta\Delta + \kappa\sqrt{\eta}\alpha\epsilon M_d}{1-\rho} + \rho^\tau\left(\left\|\hat{z}_0 - z^*_1\right\|_\eta - \frac{\sigma_\eta\Delta + \kappa\sqrt{\eta}\alpha\epsilon M_d}{1-\rho}\right) + \sigma_\eta\Delta \\
=\ & (1-\rho^\tau)\,\frac{\sigma_\eta\Delta + \kappa\sqrt{\eta}\alpha\epsilon M_d}{1-\rho} + \rho^\tau\left\|\hat{z}_0 - z^*_1\right\|_\eta .
\end{aligned}
\tag{2.26}
$$

The condition (2.23a) implies

$$
\rho < 1 \qquad \text{and} \qquad \delta \geq \frac{\sigma_\eta\Delta + \kappa\sqrt{\eta}\alpha\epsilon M_d}{1-\rho},
$$

and therefore

$$
\left\|\hat{z}_\tau - z^*_{\tau+1}\right\|_\eta \leq (1-\rho^\tau)\,\delta + \rho^\tau\delta \leq \delta.
$$

We can then use Lemma 2.3 and (2.26) to get

$$
\begin{aligned}
&\left\|\hat{z}_{\tau+1} - z^*_{\tau+1}\right\|_\eta \leq \rho\left\|\hat{z}_\tau - z^*_{\tau+1}\right\|_\eta + \kappa\sqrt{\eta}\alpha\epsilon M_d \\
\leq\ & \rho\left((1-\rho^\tau)\,\frac{\sigma_\eta\Delta + \kappa\sqrt{\eta}\alpha\epsilon M_d}{1-\rho} + \rho^\tau\left\|\hat{z}_0 - z^*_1\right\|_\eta\right) + \kappa\sqrt{\eta}\alpha\epsilon M_d \\
=\ & \frac{\rho\sigma_\eta\Delta + \kappa\sqrt{\eta}\alpha\epsilon M_d}{1-\rho} + \rho^{\tau+1}\left(\left\|\hat{z}_0 - z^*_1\right\|_\eta - \frac{\sigma_\eta\Delta + \kappa\sqrt{\eta}\alpha\epsilon M_d}{1-\rho}\right),
\end{aligned}
$$

and by induction we get (2.24) for all $\tau \in \mathcal{T}$. □

It can be seen that the tracking error bound in (2.24) consists of a constant term and a term that decays geometrically with τ, as the condition (2.23a) implies $\rho\,(\delta, \alpha, \eta, \epsilon) < 1$. We call the constant term

$$
\frac{\rho(\delta,\alpha,\eta,\epsilon)\,\sigma_\eta\Delta + \kappa(\delta,\alpha,\eta,\epsilon)\sqrt{\eta}\alpha\epsilon M_\lambda}{1-\rho(\delta,\alpha,\eta,\epsilon)}
\tag{2.27}
$$

the *eventual tracking error bound*, which can be further split into two parts:

1. The first part
$$
\frac{\rho(\delta,\alpha,\eta,\epsilon)}{1-\rho(\delta,\alpha,\eta,\epsilon)}\sigma_\eta\Delta
$$
is proportional to σ_η, the maximum speed of the KKT point. In time-varying optimization, such terms are common in the tracking error bound [35, 68, 84, 90]. This term also decreases as one reduces the sampling interval Δ.

2. The second part
$$\frac{\kappa(\delta,\alpha,\eta,\epsilon)\sqrt{\eta}\alpha\epsilon M_d}{1-\rho(\delta,\alpha,\eta,\epsilon)}$$
is proportional to M_d, the maximum distance between the optimal Lagrange multiplier $(\lambda^*(t),\mu^*(t))$ and the prior estimate $(\lambda_{\text{prior}},\mu_{\text{prior}})$. This term represents the discrepancy introduced by adding regularization on the dual variable; similar behavior has also been observed in [60].

In addition, the first part has a multiplicative factor $\rho(\delta,\alpha,\eta,\epsilon)/(1-\rho(\delta,\alpha,\eta,\epsilon))$, while the second part has a multiplicative factor $1/(1-\rho(\delta,\alpha,\eta,\epsilon))$, which are all strictly increasing in $\rho(\delta,\alpha,\eta,\epsilon)$. This implies that a smaller $\rho(\delta,\alpha,\eta,\epsilon)$ will lead to better tracking performance. The condition (2.23a) is also more likely to be satisfied when $\rho(\delta,\alpha,\eta,\epsilon)$ is smaller.

Feasible Parameters

In order that (2.23a) can be satisfied and the bound (2.27) can be as small as possible, one needs to find an appropriate set of the parameters α,η,ϵ. However, the expression that defines $\rho(\delta,\alpha,\eta,\epsilon)$ is rather complicated, making it difficult to analyze how to achieve a smaller bound (2.27); even the existence of parameters that can guarantee the condition (2.23a) is not readily available. In this section, we give a preliminary study of the conditions under which (2.23a) can be satisfied.

Definition 2.1. We say that $(\delta,\alpha,\eta,\epsilon)\in\mathbb{R}^4_{++}$ is a tuple of *feasible parameters* if

$$(1-\rho(\delta,\alpha,\eta,\epsilon))\,\delta-\kappa(\delta,\alpha,\eta,\epsilon)\sqrt{\eta}\alpha\epsilon M_d>0. \tag{2.28}$$

It can be seen that, if (2.28) is satisfied, then for sufficiently small sampling interval Δ the condition (2.23a) can be satisfied; otherwise (2.23a) cannot be satisfied no matter how much one reduces the sampling interval Δ. The quantity δ has been added to the tuple of parameters for convenience.

Define

$$\Lambda_m(\delta):=\inf_{t\in[0,T]}\inf_{u:\|u\|\le\delta}\lambda_{\min}\left(\overline{H}_{\mathcal{L}^{nc}}(u,t)+\frac{1}{2}\sum_{i=1}^{m}\lambda_i^*(t)\overline{H}_{f_i^c}(u,t)\right), \tag{2.29}$$

$$\Lambda_M(\delta):=\sup_{t\in[0,T]}\sup_{u:\|u\|\le\delta}\lambda_{\max}\left(\overline{H}_{\mathcal{L}^{nc}}(u,t)+\frac{1}{2}\sum_{i=1}^{m}\lambda_i^*(t)\overline{H}_{f_i^c}(u,t)\right). \tag{2.30}$$

Roughly speaking, $\Lambda_m(\delta)$ characterizes how convex the problem (2.1) is in the neighborbood of $x^*(t)$. It's easy to see that $\Lambda_m(\delta)$ is a nonincreasing function in δ.

Lemma 2.4. *1. Suppose X and Y are metric spaces with X being compact, and $f : X \times Y \to \mathbb{R}$ is a continuous function. Let $g : Y \to \mathbb{R}$ be given by*

$$g(y) = \sup_{x \in X} f(x, y).$$

Then g is a continuous function.

2. Suppose $f : R\mathcal{B}_n \times V \to \mathbb{R}$ is a continuous function for some $R > 0$ and V is a metric space. Define $g : (0, R) \times V \to \mathbb{R}$ by

$$g(r, v) = \sup_{u : \|u\| \leq r} f(u, v)$$

Then g is a continuous function on $(0, R) \times V$.

The proof of Lemma 2.4 will be given in Appendix 2.A.

Theorem 2.2. *Suppose there exists some $\bar{\delta} > 0$ such that*

$$\Lambda_m(\bar{\delta}) > M_d \, M_{nc}(\bar{\delta}). \tag{2.31}$$

Let

$$S_{\text{fp}} := \left\{(\delta, \alpha, \eta, \epsilon) \in \mathbb{R}^4_{++} : (1 - \rho(\delta, \alpha, \eta, \epsilon))\, \delta - \kappa(\delta, \alpha, \eta, \epsilon)\sqrt{\eta}\alpha\epsilon M_d > 0\right\}.$$

Then S_{fp} is a non-empty open subset of $\mathbb{R}^4_{++}$.

Proof. Let $R > \bar{\delta}$ be arbitrary. We first show that $\overline{H}_{\mathcal{L}^{nc}}(u, t)$ and each $\overline{H}_{f_i^c}(u, t)$ are continuous over $(u, t) \in R\mathcal{B}_n \times [0, T]$. Indeed, by their definitions (2.20) and (2.21), any entry of these matrices can be written in the form

$$\int_0^1 g(\theta, u, t)\, d\theta,$$

where g is some continuous function over $(\theta, u, t) \in [0, 1] \times R\mathcal{B}_n \times [0, T]$ as we have assumed the continuity of $\nabla^2_{xx} c(x, t)$ and each $\nabla^2_{xx} f_i^c(x, t)$, $\nabla^2_{xx} f_i^{nc}(x, t)$, $\nabla^2_{xx} f_j^{eq}(x, t)$. Since $[0, 1] \times R\mathcal{B}_n \times [0, T]$ is a compact set, $g(\theta, u, t)$ is bounded above. By the dominated convergence theorem, $\int_0^1 g(\theta, u, t)\, d\theta$ is continuous over $(u, t) \in R\mathcal{B}_n \times [0, T]$.

As a consequence of the continuity of $\overline{H}_{\mathcal{L}^{nc}}(u, t)$ and each $\overline{H}_{f_i^c}(u, t)$, $\Lambda_M(\delta)$ is finite for all $\delta \leq R$.

Let

$$f_R(\delta, \alpha, \eta, \epsilon) := (1 - \rho(\delta, \alpha, \eta, \epsilon))\delta - \kappa(\delta, \alpha, \eta, \epsilon)\sqrt{\eta}\alpha\epsilon M_d$$

for $(\delta, \alpha, \eta, \epsilon) \in (0, R) \times \mathbb{R}^3_{++}$, and let us consider the Taylor expansion of $f_R(\delta, \alpha, \eta, \epsilon)$ with respect to α as $\alpha \to 0^+$. We have

$$\begin{aligned} &\left(I - \alpha \overline{H}_{\mathcal{L}^{nc}}(u,t)\right)^2 - \alpha(1 - \eta\alpha\epsilon) \sum_{i=1}^{m} \lambda_i^*(t) \overline{H}_{f_i^c}(u,t) \\ &= I - 2\alpha \left(\overline{H}_{\mathcal{L}^{nc}}(u,t) + \frac{1}{2} \sum_{i=1}^{m} \lambda_i^*(t) \overline{H}_{f_i^c}(u,t)\right) + \alpha^2 Q(u,t,\eta,\epsilon), \end{aligned} \tag{2.32}$$

where $Q(u,t,\eta,\epsilon)$ is some positive semidefinite matrix that depends continuously on (u,t,η,ϵ). It can be checked that for $\alpha < (2\Lambda_M(R))^{-1}$, we have

$$0 \prec 2\alpha \left(\overline{H}_{\mathcal{L}^{nc}}(u,t) + \frac{1}{2} \sum_{i=1}^{m} \lambda_i^*(t) \overline{H}_{f_i^c}(u,t)\right) \prec I$$

whenever $\|u\| < R$, and so

$$\begin{aligned} &\left\| I - 2\alpha \left(\overline{H}_{\mathcal{L}^{nc}}(u,t) + \frac{1}{2} \sum_{i=1}^{m} \lambda_i^*(t) \overline{H}_{f_i^c}(u,t)\right) \right\| \\ &= 1 - 2\alpha \lambda_{\min} \left(\overline{H}_{\mathcal{L}^{nc}}(u,t) + \frac{1}{2} \sum_{i=1}^{m} \lambda_i^*(t) \overline{H}_{f_i^c}(u,t)\right) \end{aligned}$$

whenever $\|u\| < R$ and $\alpha < (2\Lambda_M(R))^{-1}$. By (2.32), we get

$$\begin{aligned} &\sup_{t \in [0,T]} \sup_{u:\|u\| \le \delta} \left\| \left(I - \alpha \overline{H}_{\mathcal{L}^{nc}}(u,t)\right)^2 - \alpha(1 - \eta\alpha\epsilon) \sum_{i=1}^{m} \lambda_i^*(t) \overline{H}_{f_i^c}(u,t) \right\| \\ &= 1 - 2\alpha \Lambda_m(\delta) + O\left(\alpha^2\right) \end{aligned}$$

as $\alpha \to 0^+$ for any $\delta \in (0, R)$. Consequently, by Taylor expansion of $\rho(\delta, \alpha, \eta, \epsilon)$ with respect to α and using the properties of $\kappa(\delta, \alpha, \eta, \epsilon)$, we can show that

$$\rho(\delta, \alpha, \eta, \epsilon) = 1 - \alpha \left(\min\{\Lambda_m(\delta), \eta\epsilon\} - \frac{\sqrt{\eta}}{4} \delta M_{nc}(\delta)\right) + O\left(\alpha^2\right) \tag{2.33}$$

and

$$\begin{aligned} &f_R(\delta, \alpha, \eta, \epsilon) \\ &= \alpha \left(\delta \left(\min\{\Lambda_m(\delta), \eta\epsilon\} - \frac{\sqrt{\eta}}{4} \delta M_{nc}(\delta)\right) - \sqrt{\eta}\epsilon M_d\right) + O\left(\alpha^2\right). \end{aligned} \tag{2.34}$$

Now we consider two cases:

1. $M_d > 0$: Let $\delta_0 = \bar{\delta}$ and

$$\eta_0 = \left(\frac{2\Lambda_m(\delta_0)}{\delta_0 M_{nc}(\delta_0)}\right)^2, \qquad \epsilon_0 = \frac{\Lambda_m(\delta_0)}{\eta_0}.$$

   By (2.34),

$$f_R(\delta_0, \alpha, \eta_0, \epsilon_0) = \alpha\frac{\bar{\delta}}{2}\left(\Lambda_m(\bar{\delta}) - M_d M_{nc}(\bar{\delta})\right) + O(\alpha^2). \tag{2.35}$$

   Togeher with the condition (2.31), we can see from (2.35) there exists a sufficiently small $\alpha_0 > 0$ such that $f_R(\delta_0, \alpha_0, \eta_0, \epsilon_0)$ is positive, and consequently S_{fp} is non-empty.

2. $M_d = 0$: Let $\eta_0 > 0$ be arbitrary. Consider the function

$$g(\delta) = \Lambda_m(\delta) - \frac{\sqrt{\eta_0}}{4}\delta M_{nc}(\delta).$$

   By the monotonicity of $\Lambda_m(\delta)$ and $M_{nc}(\delta)$,

$$\lim_{\delta\to 0^+} g(\delta) = \lim_{\delta\to 0^+} \Lambda_m(\delta) \geq \Lambda_m\left(\bar{\delta}\right) > 0.$$

   Therefore there exists some $\delta_0 \in \left(0, \bar{\delta}\right]$ such that $g(\delta_0) > 0$.

   Now let $\epsilon_0 = \eta_0^{-1}\Lambda_m(\delta_0)$. By (2.34),

$$f_R(\delta_0, \alpha, \eta_0, \epsilon_0) = \alpha\delta_0 g(\delta_0) + O(\alpha^2). \tag{2.36}$$

   Therefore we can find some $\alpha_0 > 0$ such that $f_R(\delta_0, \alpha_0, \eta_0, \epsilon_0)$ is positive, and consequently S_{fp} is non-empty.

Finally, by Lemma 2.4, it can be seen that $f_R(\delta, \alpha, \eta, \epsilon)$ is a continuous function over $(\delta, \alpha, \eta, \epsilon) \in (0, R) \times \mathbb{R}^3_{++}$. Therefore the set

$$S_{\text{fp}} \cap \left((0, R) \times \mathbb{R}^3_{++}\right) = \{(\delta, \alpha, \eta, \epsilon) \in (0, R) \times \mathbb{R}^3_{++} : f_R(\delta, \alpha, \eta, \epsilon) > 0\}$$

is an open subset of $\mathbb{R}^4_{++}$, and consequently

$$S_{\text{fp}} = \bigcup_{R > \bar{\delta}} S_{\text{fp}} \cap \left((0, R) \times \mathbb{R}^3_{++}\right)$$

is an open subset of $\mathbb{R}^4_{++}$. □

The condition (2.31) for the existence of feasible parameters can be intuitively interpreted as follows: The problem should be sufficiently convex around the KKT trajectory to overcome the nonlinearity of the nonconvex constraints. It should be emphasized that this is only a sufficient condition.

The parameters $(\delta_0, \alpha_0, \eta_0, \epsilon_0)$ given in the proof are in general not the optimal choice of parameters. However, they do provide some insights on how to choose parameters in practical situations:

1. Since ϵ controls the amount of regularization on the dual variable, a smaller ϵ can help reduce the inaccuracy introduced by the regularization. On the other hand, if ϵ is two small, one could have $\rho^{(\mathrm{P})}(\delta, \alpha, \eta, \epsilon) < (1 - \eta\alpha\epsilon)^2$, and the contraction coefficient $\rho(\delta, \alpha, \eta, \epsilon)$ can be too close to 1, which degrades the overall tracking performance. In the proof we set $\epsilon_0 = \eta_0^{-1}\Lambda_m(\delta_0)$ so that $\rho^{(\mathrm{P})}(\delta_0, \alpha_0, \eta_0, \epsilon_0) \approx (1 - \eta_0\alpha_0\epsilon_0)^2$.
2. It can be seen that, in the case where the constraints and objective are all convex, if we choose ϵ as previously mentioned, then the "contraction coefficient" $\rho(\delta, \alpha, \eta, \epsilon)$ has a particular form

   $$\rho(\delta, \alpha, \eta, \epsilon) \approx 1 - \alpha\Lambda_m(\delta) + \alpha^2 \cdot (\text{additional terms}),$$

   where those "additional terms" are mainly related to the Lipschitz constants of various functions. Similar forms have appeared in [35, 60, 87]. For the nonconvex case, the coefficient of the first-order term will be reduced, but Theorem 2.2 guarantees that it can be made positive by appropriate choice of parameters under the condition (2.31).

   Because of the "additional terms", we could choose α to be sufficiently small so that $\rho(\delta, \alpha, \eta, \epsilon)$ is less than 1, but an excessively small α can also degrade the tracking performance. In practice the step size can be chosen heuristically by experiments or simulations. Adaptive step sizes will be left for future study.

In the proof of Theorem 2.2, we consider the asymptotic behavior of $\rho(\delta, \alpha, \eta, \epsilon)$ as the step size α approaches zero, which greatly simplifies the analysis. It is well known that when the step size is very small, the classical projected gradient descent can be viewed as good approximation of the continuous-time projected gradient flow [71] which has simpler analysis but still provides valuable results for understanding the discrete-time counterpart. This observation suggests that by studying the continuous-time limit of (2.10), we may get a better understanding of the discrete-time algorithm.

2.4 Continuous-Time Limit

In this section, we study the continuous-time limit of the regularized proximal primal-dual gradient algorithm (2.10).

Let $\hat{z}_0 = (\hat{x}_0, \hat{\lambda}_0, \hat{\mu}_0)$ be the initial primal-dual pair. For each $K \in \mathbb{N}$, let $\hat{z}_\tau^{(K)} = \left(\hat{x}_\tau^{(K)}, \hat{\lambda}_\tau^{(K)}, \hat{\mu}_\tau^{(K)}\right)$, $\tau = 0, 1, 2, \ldots, K$ be the sequence produced by (2.10) with sampling interval $\Delta = \Delta_K := T/K$ and step size $\alpha = \Delta_K \beta$ for some fixed $\beta > 0$. In other words, we let $\hat{z}_0^{(K)} = \hat{z}_0$ and $\hat{z}_\tau^{(K)} = \left(\hat{x}_\tau^{(K)}, \hat{\lambda}_\tau^{(K)}, \hat{\mu}_\tau^{(K)}\right)$ with

$$\hat{x}_\tau^{(K)} = \mathrm{prox}_{\Delta_K \beta h_\tau}\left[\hat{x}_{\tau-1}^{(K)} - \Delta_K \beta \left(\nabla c_\tau\left(\hat{x}_{\tau-1}^{(K)}\right) + \begin{bmatrix} J_{f_\tau^{in}}\left(\hat{x}_{\tau-1}^{(K)}\right) \\ J_{f_\tau^{eq}}\left(\hat{x}_{\tau-1}^{(K)}\right)\end{bmatrix}^T \begin{bmatrix} \lambda_{\tau-1}^{(K)} \\ \hat{\mu}_{\tau-1}^{(K)} \end{bmatrix}\right)\right], \tag{2.37a}$$

$$\hat{\lambda}_\tau^{(K)} = \mathcal{P}_{\mathbb{R}_+^m}\left[\hat{\lambda}_{\tau-1}^{(K)} + \Delta_K \eta \beta \left(f_\tau^{in}\left(\hat{x}_{\tau-1}^{(K)}\right) - \epsilon\left(\hat{\lambda}_{\tau-1}^{(K)} - \lambda_{\text{prior}}\right)\right)\right], \tag{2.37b}$$

$$\hat{\mu}_\tau^{(K)} = \hat{\mu}_{\tau-1}^{(K)} + \Delta_K \eta \beta \left(f_\tau^{eq}\left(\hat{x}_{\tau-1}^{(K)}\right) - \epsilon\left(\hat{\mu}_{\tau-1}^{(K)} - \mu_{\text{prior}}\right)\right), \tag{2.37c}$$

for $\tau = 1, 2, \ldots, K$. Define

$$\hat{z}^{(K)}(t) = \frac{\tau\Delta_K - t}{\Delta_K}\hat{z}_{\tau-1}^{(K)} + \frac{t - (\tau-1)\Delta_K}{\Delta_K}\hat{z}_\tau^{(K)} \tag{2.38}$$

if $t \in [(\tau-1)\Delta_K, \tau\Delta_K]$ for each $\tau = 1, \ldots, K$. It can be seen that $\hat{z}^{(K)}$ is a linear interpolation of the set of points $\left\{\left(\tau\Delta_K, \hat{z}_\tau^{(K)}\right)\right\}$, and we are interested in the behavior of $\hat{z}^{(K)}$ when $K \to \infty$.

Definition 2.2. Given $S : [0, T] \to 2^{\mathbb{R}^k}$, we say that $S(t)$ is κ-Lipschitz in t if

$$d_H\left(S(t_1), S(t_2)\right) \le \kappa |t_1 - t_2|$$

for all $t_1, t_2 \in [0, T]$, where and d_H denotes the Hausdorff distance of two sets:

$$d_H(A, B) := \max\left\{\sup_{x\in A}\inf_{y\in B}\|x - y\|, \sup_{y\in B}\inf_{x\in A}\|x - y\|\right\}.$$

Theorem 2.3. *Suppose the following conditions hold:*

1. $\mathrm{graph}_t(h)$ *is* κ_1*-Lipschitz in* t *for some* $\kappa_1 > 0$*, where*

$$\mathrm{graph}_t(h) := \{(x, h(x,t)) : x \in \mathrm{dom}_t(h)\}$$

denotes the graph of the function $x \mapsto h(x, t)$ *for each* $t \in [0, T]$*;*

2. *there exists* $\ell > 0$ *such that*

$$\sup_{x_1, x_2 \in \mathrm{dom}_t(h)} \frac{|h(x_2, t) - h(x_1, t)|}{\|x_2 - x_1\|} \le \ell, \qquad \forall t \in [0, T]; \tag{2.39}$$

3. *$\nabla_x c(x,t)$ is continuous on the set $\bigcup_{t\in[0,T]} \mathrm{dom}_t(h) \times [0,T]$, and there exists some $\kappa_2 > 0$ such that*

$$\|\nabla_x c(x,t)\| \le \kappa_2(1+\|x\|), \quad \forall (x,t) \in \bigcup_{t\in[0,T]} \mathrm{dom}_t(h) \times [0,T]; \tag{2.40}$$

4. *$f^{in}(x,t)$ and $f^{eq}(x,t)$ are continuous, and $J_{f^{in},x}(x,t)$ and $J_{f^{eq},x}(x,t)$ are bounded and continuous over $(x,t) \in \bigcup_{t\in[0,T]} \mathrm{dom}_t(h) \times [0,T]$. Moreover, there exists some integrable function $l : [0,T] \to \mathbb{R}_+$ such that for each $t \in [0,T]$*

$$\begin{aligned}
\sup_{x\in \mathrm{dom}_t(h)} \left\|\nabla^2_{xx} c(x,t)\right\| &\le l(t), \\
\sup_{x\in \mathrm{dom}_t(h)} \left\|\nabla^2_{xx} f^{in}_j(x,t)\right\| &\le l(t), \qquad \forall j = 1,\dots m, \\
\sup_{x\in \mathrm{dom}_t(h)} \left\|\nabla^2_{xx} f^{eq}_j(x,t)\right\| &\le l(t), \qquad \forall j = 1,\dots m'.
\end{aligned} \tag{2.41}$$

We keep T constant and let $K \to \infty$. Then the sequence of trajectories $\left(\hat{z}^{(K)}\right)_{K\in\mathbb{N}}$ defined in (2.38) *converges uniformly to some Lipschitz continuous $\hat{z} = \left(\hat{x}, \hat{\lambda}, \hat{\mu}\right)$ that satisfies*

$$-\frac{d}{dt}\hat{x}(t) - \beta\left(\nabla_x c(\hat{x}(t),t) + \begin{bmatrix} J_{f^{in},x}(\hat{x}(t),t) \\ J_{f^{eq},x}(\hat{x}(t),t)\end{bmatrix}^T \begin{bmatrix}\hat{\lambda}(t) \\ \hat{\mu}(t)\end{bmatrix}\right) \in \beta\partial_x h(\hat{x}(t),t), \tag{2.42a}$$

$$-\frac{d}{dt}\hat{\lambda}(t) + \eta\beta\left(f^{in}(\hat{x}(t),t) - \epsilon\left(\hat{\lambda}(t) - \lambda_{\mathrm{prior}}\right)\right) \in N_{\mathbb{R}^m_+}\left(\hat{\lambda}(t)\right), \tag{2.42b}$$

$$-\frac{d}{dt}\hat{\mu}(t) + \eta\beta\left(f^{eq}(\hat{x}(t),t) - \epsilon\left(\hat{\mu}(t) - \mu_{\mathrm{prior}}\right)\right) = 0 \tag{2.42c}$$

for almost all $t \in [0,T]$. Moreover, the differential inclusions (2.42) *have a unique absolutely continuous solution given any initial condition $\hat{z}(0) = \hat{z}_0$.*

The proof of Theorem 2.3 is given in Appendix 2.A.

Remark 2.4. Theorem 2.3 indicates that a continuous-time counterpart of the discrete-time algorithm (2.10) is given by the differential inclusions (2.42). In the special case where h is the indicator function of a compact convex set $\mathcal{X}(t)$ as given in (2.6), the differential inclusions (2.42) can be written in the form

$$-\frac{d}{dt}\hat{z}(t) + \Phi(\hat{z}(t),t) \in N_{C(t)}(\hat{z}(t)), \tag{2.43}$$

where $C(t) = \mathcal{X}(t) \times \mathbb{R}^m_+ \times \mathbb{R}^{m'}$ is convex and $\Phi : \mathbb{R}^n \times \mathbb{R}^m_+ \times \mathbb{R}^{m'} \to \mathbb{R}^n \times \mathbb{R}^m_+ \times \mathbb{R}^{m'}$ gives the gradient step of the primal and dual variables. This form of differential inclusions has been studied under the name *perturbed sweeping processes* in the literature [28], and the discrete-time algorithm has been called the catching algorithm of the perturbed sweeping processes.

It should be noted that, when the convex set $C(t)$ is time-varying, the perturbed sweeping process (2.43) in general cannot be equivalently written in the form

$$\frac{d}{dt}\hat{z}(t) = \mathcal{P}_{T_{C(t)}(\hat{z}(t))}\left[\Phi(\hat{z}(t), t)\right], \tag{2.44a}$$

where $T_{C(t)}$ denotes the tangent cone of $C(t)$, or

$$\frac{d}{dt}\hat{z}(t) = \lim_{s\to 0^+} \frac{\mathcal{P}_{C(t)}\left[\hat{z}(t) + s\Phi(\hat{z}(t), t)\right] - \hat{z}(t)}{s} \tag{2.44b}$$

as an ordinal projected dynamical system, as there may not exist solutions on $[0, T]$ that satisfy these equations almost everywhere. A simple example is given by $C(t) = \{(x_1, x_2) \in \mathbb{R}^2 : x_1 \geq t\}$ with $\Phi(z, t) = 0$. It can be seen that under the initial condition $(x_1(0), x_2(0)) = (0, 0)$, (2.43) admits the solution $x_1(t) = t, x_2(t) = 0$, but (2.44a) and (2.44b) do not have solutions. In [54], the authors introduced a formulation similar to (2.44a) based on the notion of temporal tangent cones, which is a generalization of tangent cones in time-varying situations. ■

Remark 2.5. Among the four conditions listed in Theorem 2.3, the first three conditions and the continuity of f^{in}, f^{eq}, $J_{f^{in},x}$ and $J_{f^{eq},x}$ guarantee that $\left(\hat{z}^{(K)}\right)_{K\in\mathbb{N}}$ will always have a convergent subsequence, any any convergent subsequence will converge uniformly to an absolutely continuous solution to (2.42), while by adding the existence of the function $l(t)$ that satisfies (2.41), we can further derive the uniqueness of the absolutely continuous solution to (2.42) and the convergence of the sequence $\left(\hat{z}^{(K)}\right)_{K\in\mathbb{N}}$ itself. ■

Tracking Performance

After showing that the differential inclusions (2.42) give the correct continuous-time limit of the algorithm (2.10), we proceed to study the tracking performance under the continuous-time limit.

We first present a Gronwall-type lemma, whose proof is given in Appendix 2.A.

Lemma 2.5. *Let $v(t)$ be a nonnegative absolutely continuous function that satisfies*

$$\frac{1}{2}\frac{d}{dt}\left(v^2(t)\right) \leq av(t) - bv^2(t)$$

for almost all $t \in [0, T]$, where a and b are nonnegative constants. Then

$$v(t) \leq e^{-bt}v(0) + \frac{a}{b}(1 - e^{-bt}).$$

The following theorem then establishes the tracking error bound of the differential inclusions (2.42), which is the continuous-time counterpart of Theorem 2.1. It characterizes the behavior of the discrete-time algorithm (2.10) under the continuous-time limit.

Theorem 2.4. *Suppose there exists $\delta > 0$, $\beta > 0$, $\eta > 0$ and $\epsilon > 0$ such that*

$$\beta^{-1}\sigma_\eta < \delta\,\gamma(\delta,\eta,\epsilon) - \sqrt{\eta}\epsilon M_d, \tag{2.45}$$

where

$$\gamma(\delta,\eta,\epsilon) := \min\{\Lambda_m(\delta), \eta\epsilon\} - \frac{\sqrt{\eta}}{4}\delta M_{nc}(\delta). \tag{2.46}$$

Let $\hat{z}(t)$ be a Lipschitz continuous solution to (2.42) *with $\|\hat{z}(0) - z^*(0)\|_\eta < \delta$. Then for all $t \in [0,T]$,*

$$\begin{aligned}\|\hat{z}(t) - z^*(t)\|_\eta < {} & \frac{\beta^{-1}\sigma_\eta + \sqrt{\eta}\epsilon M_d}{\gamma(\delta,\eta,\epsilon)} \\ & + e^{-\beta\gamma(\delta,\eta,\epsilon)\,t}\left(\|\hat{z}(0) - z^*(0)\|_\eta - \frac{\beta^{-1}\sigma_\eta + \sqrt{\eta}\epsilon M_d}{\gamma(\delta,\eta,\epsilon)}\right).\end{aligned} \tag{2.47}$$

Proof. First of all, we notice that (2.42a) implies

$$\begin{aligned}&\beta\left(h(x^*(t),t) - h(\hat{x}(t),t)\right) \\ &\geq (x^*(t) - \hat{x}(t))^T\left(-\frac{d}{dt}\hat{x}(t) - \beta\nabla_x c(\hat{x}(t),t) - \beta\begin{bmatrix} J_{f^{in},x}(\hat{x}(t),t) \\ J_{f^{eq},x}(\hat{x}(t),t)\end{bmatrix}^T \begin{bmatrix}\hat{\lambda}(t) \\ \hat{\mu}(t)\end{bmatrix}\right) \\ &= (x^*(t) - \hat{x}(t))^T\Bigg(- \frac{d}{dt}\hat{x}(t) - \beta\nabla_x\mathcal{L}^{nc}(\hat{x}(t),\lambda^*(t),\mu^*(t),t) \\ &\qquad - \beta J_{f^c,x}(\hat{x}(t),t)^T\hat{\lambda}(t) + \beta\begin{bmatrix} J_{f^{nc},x}(\hat{x}(t),t) \\ J_{f^{eq},x}(\hat{x}(t),t)\end{bmatrix}^T\begin{bmatrix}\lambda^*(t) - \hat{\lambda}(t) \\ \mu^*(t) - \hat{\mu}(t)\end{bmatrix}\Bigg),\end{aligned}$$

and by (2.2b),

$$\begin{aligned}&\beta(h(\hat{x}(t),t) - h(x^*(t),t)) \\ &\geq (\hat{x}(t) - x^*(t))^T\left(-\beta\nabla_x c(x^*(t),t) - \beta\begin{bmatrix} J_{f^{in},x}(x^*(t),t) \\ J_{f^{eq},x}(x^*(t),t)\end{bmatrix}^T\begin{bmatrix}\lambda^*(t) \\ \mu^*(t)\end{bmatrix}\right) \\ &= (\hat{x}(t) - x^*(t))^T\left(-\beta\nabla_x\mathcal{L}^{nc}(x^*(t),\lambda^*(t),\mu^*(t),t) - \beta J_{f^c,x}(x^*(t),t)^T\lambda^*(t)\right).\end{aligned}$$

Thus

$$
\begin{aligned}
&(\hat{x}(t)-x^*(t))^T \frac{d}{dt}\hat{x}(t) \\
\leq\ & -\beta(\hat{x}(t)-x^*(t))^T\Bigg(\nabla_x \mathcal{L}^{nc}(\hat{x}(t),\lambda^*(t),\mu^*(t),t) - \nabla_x \mathcal{L}^{nc}(x^*(t),\lambda^*(t),\mu^*(t),t) \\
&+ J_{f^c,x}(\hat{x}(t),t)^T\hat{\lambda}(t) - J_{f^c,x}(x^*(t),t)^T\lambda^*(t) + \begin{bmatrix} J_{f^{nc},x}(\hat{x}(t),t) \\ J_{f^{eq},x}(\hat{x}(t),t)\end{bmatrix}^T \begin{bmatrix}\hat{\lambda}(t)-\lambda^*(t) \\ \hat{\mu}(t)-\mu^*(t)\end{bmatrix}\Bigg) \\
=\ & -\beta(\hat{x}(t)-x^*(t))^T B_{\mathcal{L}^{nc}}(t)\,(\hat{x}(t)-x^*(t)) \\
&- \beta(\hat{x}(t)-x^*(t))^T\Bigg(J_{f^c,x}(\hat{x}(t),t)^T\hat{\lambda}(t) - J_{f^c,x}(x^*(t),t)^T\lambda^*(t) \\
&\qquad + \begin{bmatrix} J_{f^{nc},x}(\hat{x}(t),t) \\ J_{f^{eq},x}(\hat{x}(t),t)\end{bmatrix}^T \begin{bmatrix}\hat{\lambda}(t)-\lambda^*(t) \\ \hat{\mu}(t)-\mu^*(t)\end{bmatrix}\Bigg),
\end{aligned}
$$

where $B_{\mathcal{L}^{nc}}(t) := \overline{H}_{\mathcal{L}^{nc}}(\hat{x}(t)-x^*(t),t)$. Then, by (2.42b),

$$
0 \geq (\lambda^*(t)-\hat{\lambda}(t))^T\left(-\frac{d}{dt}\hat{\lambda}(t) + \eta\beta\left(f^{in}(\hat{x}(t),t) - \epsilon\big(\hat{\lambda}(t)-\lambda_{\text{prior}}\big)\right)\right),
$$

and by (2.2c),

$$
0 \geq \eta\beta\big(\hat{\lambda}(t)-\lambda^*(t)\big)^T f^{in}(x^*(t),t),
$$

which lead to

$$
\begin{aligned}
&\big(\hat{\lambda}(t)-\lambda^*(t)\big)^T\frac{d}{dt}\hat{\lambda}(t) \\
&\leq \eta\beta\big(\hat{\lambda}(t)-\lambda^*(t)\big)^T\left(f^{in}(\hat{x}(t),t) - f^{in}(x^*(t),t)\right) \\
&\quad - \eta\beta\epsilon\left\|\hat{\lambda}(t)-\lambda^*(t)\right\|^2 - \eta\beta\epsilon\big(\hat{\lambda}(t)-\lambda^*(t)\big)^T(\lambda^*(t)-\lambda_{\text{prior}}).
\end{aligned}
$$

Similarly by (2.42c) and (2.2d), we have

$$
\begin{aligned}
&(\hat{\mu}(t)-\mu^*(t))^T\frac{d}{dt}\hat{\mu}(t) \\
&= \beta\eta(\hat{\mu}(t)-\mu^*(t))^T\left(f^{eq}(\hat{x}(t),t) - f^{eq}(x^*(t),t)\right) \\
&\quad - \eta\beta\epsilon\left\|\hat{\mu}(t)-\mu^*(t)\right\|^2 - \eta\beta\epsilon\big(\hat{\mu}(t)-\mu^*(t)\big)^T(\mu^*(t)-\mu_{\text{prior}}).
\end{aligned}
$$

Then it can be seen that

$$
\begin{aligned}
&\frac{1}{2}\frac{d}{dt}\|\hat{z}(t)-z^*(t)\|_\eta^2 \\
&= (\hat{x}(t)-x^*(t))^T\frac{d}{dt}\hat{x}(t)+\eta^{-1}\left(\hat{\lambda}(t)-\lambda^*(t)\right)^T\frac{d}{dt}\hat{\lambda}(t)+\eta^{-1}(\hat{\mu}(t)-\mu^*(t))^T\frac{d}{dt}\hat{\mu}(t) \\
&\quad -(\hat{z}(t)-z^*(t))^T\begin{bmatrix} I_n & \\ & \eta^{-1}I_{m+m'}\end{bmatrix}\frac{d}{dt}z^*(t) \\
&\le -\beta(\hat{x}(t)-x^*(t))^T B_{\mathcal{L}^{nc}}(t)(\hat{x}(t)-x^*(t)) \\
&\quad -\beta\epsilon\left(\|\hat{\lambda}(t)-\lambda^*(t)\|^2+\|\hat{\mu}(t)-\mu^*(t)\|^2\right) \\
&\quad -\beta(\hat{x}(t)-x^*(t))^T\left(J_{f^c,x}(\hat{x}(t),t)^T\hat{\lambda}(t)-J_{f^c,x}(x^*(t),t)^T\lambda^*(t)\right) \\
&\quad -\beta(\hat{x}(t)-x^*(t))^T\begin{bmatrix} J_{f^{nc},x}(\hat{x}(t),t) \\ J_{f^{eq},x}(\hat{x}(t),t)\end{bmatrix}^T\begin{bmatrix}\hat{\lambda}(t)-\lambda^*(t) \\ \hat{\mu}(t)-\mu^*(t)\end{bmatrix} \\
&\quad +\beta\begin{bmatrix} f^{in}(\hat{x}(t),t)-f^{in}(x^*(t),t) \\ f^{eq}(\hat{x}(t),t)-f^{eq}(x^*(t),t)\end{bmatrix}^T\begin{bmatrix}\hat{\lambda}(t)-\lambda^*(t) \\ \hat{\mu}(t)-\mu^*(t)\end{bmatrix} \\
&\quad -\beta\epsilon\begin{bmatrix}\lambda^*(t)-\lambda_{\text{prior}} \\ \mu^*(t)-\mu_{\text{prior}}\end{bmatrix}^T\begin{bmatrix}\hat{\lambda}(t)-\lambda^*(t) \\ \hat{\mu}(t)-\mu^*(t)\end{bmatrix}+\sigma_\eta\|\hat{z}(t)-z^*(t)\|_\eta,
\end{aligned}
$$

in which

$$
\begin{aligned}
&\quad -(\hat{x}(t)-x^*(t))^T\left(J_{f^c,x}(\hat{x}(t),t)^T\hat{\lambda}(t)-J_{f^c,x}(x^*(t),t)^T\lambda^*(t)\right) \\
&\quad -(\hat{x}(t)-x^*(t))^T\begin{bmatrix} J_{f^{nc},x}(\hat{x}(t),t) \\ J_{f^{eq},x}(\hat{x}(t),t)\end{bmatrix}^T\begin{bmatrix}\hat{\lambda}(t)-\lambda^*(t) \\ \hat{\mu}(t)-\mu^*(t)\end{bmatrix} \\
&\quad +\begin{bmatrix} f^{in}(\hat{x}(t),t)-f^{in}(x^*(t),t) \\ f^{eq}(\hat{x}(t),t)-f^{eq}(x^*(t),t)\end{bmatrix}^T\begin{bmatrix}\hat{\lambda}(t)-\lambda^*(t) \\ \hat{\mu}(t)-\mu^*(t)\end{bmatrix} \\
&= -\hat{\lambda}(t)^T\left(f^c(x^*(t),t)-f(\hat{x}(t),t)-J_{f^c,x}(\hat{x}(t),t)(x^*(t)-\hat{x}(t))\right) \\
&\quad -\lambda^*(t)^T\left(f^c(\hat{x}(t),t)-f(x^*(t),t)-J_{f^c,x}(x^*(t),t)(\hat{x}(t)-x^*(t))\right) \\
&\quad +\begin{bmatrix} f^{nc}(\hat{x}(t),t)+J_{f^{nc},x}(\hat{x}(t),t)(x^*(t)-\hat{x}(t))-f^{nc}(x^*(t),t) \\ f^{eq}(\hat{x}(t),t)+J_{f^{eq},x}(\hat{x}(t),t)(x^*(t)-\hat{x}(t))-f^{eq}(x^*(t),t)\end{bmatrix}^T\begin{bmatrix}\hat{\lambda}(t)-\lambda^*(t) \\ \hat{\mu}(t)-\mu^*(t)\end{bmatrix}.
\end{aligned}
$$

Since

$$
f^c(x^*(t),t)-f(\hat{x}(t),t)-J_{f^c,x}(\hat{x}(t),t)(x^*(t)-\hat{x}(t))\ge 0
$$

by the convexity of f^c, and

$$
\begin{aligned}
&\lambda^*(t)^T\left(f^c(\hat{x}(t),t)-f(x^*(t),t)-J_{f^c,x}(x^*(t),t)(\hat{x}(t)-x^*(t))\right) \\
&= (\hat{x}(t)-x^*(t))^T\left(\frac{1}{2}\sum_{i=1}^m\lambda_i^*(t)B_{f_i^c}(t)\right)(\hat{x}(t)-x^*(t)),
\end{aligned}
$$

where we denote $B_{f_i^c}(t) := \overline{H}_{f_i^c}(\hat{x}(t) - x^*(t), t)$, we can see that

$$
\begin{aligned}
&\frac{1}{2}\frac{d}{dt}\|\hat{z}(t) - z^*(t)\|_\eta^2 \\
\leq\ & -\beta(\hat{x}(t) - x^*(t))^T \left(B_{\mathcal{L}^{nc}}(t) + \frac{1}{2}\sum_{i=1}^m \lambda_i^*(t) B_{f_i^c}(t)\right)(\hat{x}(t) - x^*(t)) \\
& - \beta\epsilon\left(\left\|\hat{\lambda}(t) - \lambda^*(t)\right\|^2 + \|\hat{\mu}(t) - \mu^*(t)\|^2\right) \\
& + \beta \begin{bmatrix} f^{nc}(\hat{x}(t), t) + J_{f^{nc},x}(\hat{x}(t), t)(x^*(t) - \hat{x}(t)) - f^{nc}(x^*(t), t) \\ f^{eq}(\hat{x}(t), t) + J_{f^{eq},x}(\hat{x}(t), t)(x^*(t) - \hat{x}(t)) - f^{eq}(x^*(t), t) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}(t) - \lambda^*(t) \\ \hat{\mu}(t) - \mu^*(t) \end{bmatrix} \\
& - \beta\epsilon \begin{bmatrix} \lambda^*(t) - \lambda_{\text{prior}} \\ \mu^*(t) - \mu_{\text{prior}} \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}(t) - \lambda^*(t) \\ \hat{\mu}(t) - \mu^*(t) \end{bmatrix} + \sigma_\eta \|\hat{z}(t) - z^*(t)\|_\eta .
\end{aligned} \tag{2.48}
$$

Let κ_1 be the Lipschitz constant of $\hat{z}(t)$ with respect to the norm $\|\cdot\|_\eta$. Define

$$
\tilde{\Delta} := \frac{1}{2(\kappa_1 + \sigma)}\left(\delta - \max\left\{\|\hat{z}(0) - z^*(0)\|_\eta\,, \frac{\beta^{-1}\sigma_\eta + \sqrt{\eta}\epsilon M_d}{\gamma(\delta, \eta, \epsilon)}\right\}\right).
$$

We prove by induction that

$$
\|\hat{z}(t) - z^*(t)\|_\eta \leq \max\left\{\|\hat{z}(0) - z^*(0)\|_\eta, \frac{\beta^{-1}\sigma_\eta + \sqrt{\eta}\epsilon M_d}{\gamma(\delta, \eta, \epsilon)}\right\} \tag{2.49}
$$

for $t \in \left[(k-1)\tilde{\Delta}, k\tilde{\Delta}\right] \cap [0, T]$ for each $k = 1, \ldots, \lceil T/\tilde{\Delta}\rceil$. Obviously (2.49) holds for $t = 0$. Now assume that (2.49) holds for $t = k\tilde{\Delta}$. Then we have

$$
\begin{aligned}
\|\hat{z}(t) - z^*(t)\|_\eta &\leq \left\|\hat{z}(t) - \hat{z}(k\tilde{\Delta})\right\|_\eta + \left\|\hat{z}(k\tilde{\Delta}) - z^*(k\tilde{\Delta})\right\|_\eta + \left\|z^*(k\tilde{\Delta}) - z^*(t)\right\|_\eta \\
&\leq (\kappa_1 + \sigma_\eta)\tilde{\Delta} + \left\|\hat{z}(k\tilde{\Delta}) - z^*(k\tilde{\Delta})\right\|_\eta < \delta
\end{aligned}
$$

for any $t \in \left[k\tilde{\Delta}, (k+1)\tilde{\Delta}\right] \cap [0, T]$. Therefore by the definition of $M_{nc}(\delta)$ and (2.19),

$$
\begin{aligned}
&\left\|\begin{bmatrix} f^{nc}(\hat{x}(t), t) + J_{f^{nc},x}(\hat{x}(t), t)(x^*(t) - \hat{x}(t)) - f^{nc}(x^*(t), t) \\ f^{eq}(\hat{x}(t), t) + J_{f^{eq},x}(\hat{x}(t), t)(x^*(t) - \hat{x}(t)) - f^{eq}(x^*(t), t) \end{bmatrix}\right\| \\
\leq\ & \frac{M_{nc}(\delta)}{2}\|\hat{x}(t) - x^*(t)\|^2
\end{aligned} \tag{2.50}
$$

for $t \in \left[k\tilde{\Delta}, (k+1)\tilde{\Delta}\right] \cap [0, T]$. Moreover, by Young's inequality,

$$
\begin{aligned}
\|\hat{x}(t) - x^*(t)\| \left\|\begin{bmatrix} \hat{\lambda}(t) - \lambda^*(t) \\ \hat{\mu}(t) - \mu^*(t) \end{bmatrix}\right\| &\leq \frac{1}{2}\left(\sqrt{\eta}\,\|\hat{x}(t) - x^*(t)\|^2 + \frac{1}{\sqrt{\eta}}\left\|\begin{bmatrix} \hat{\lambda}(t) - \lambda^*(t) \\ \hat{\mu}(t) - \mu^*(t) \end{bmatrix}\right\|^2\right) \\
&= \frac{\sqrt{\eta}}{2}\|\hat{z}(t) - z^*(t)\|_\eta^2 .
\end{aligned} \tag{2.51}
$$

Combining (2.50) and (2.51) with (2.48), we see that

$$\begin{aligned}
&\frac{1}{2}\frac{d}{dt}\|\hat{z}(t) - z^*(t)\|_\eta^2 \\
\le\ & -\beta(\hat{x}(t) - x^*(t))^T \left(B_{\mathcal{L}^{nc}}(t) + \frac{1}{2}\sum_{i=1}^{m} \lambda_i^*(t) B_{f_i^c}(t) \right) (\hat{x}(t) - x^*(t)) \\
& - \beta\epsilon \left(\left\|\hat{\lambda}(t) - \lambda^*(t)\right\|^2 + \|\hat{\mu}(t) - \mu^*(t)\|^2 \right) \\
& + \frac{\beta M_{nc}(\delta)}{2} \|\hat{x}(t) - x^*(t)\|^2 \left\| \begin{bmatrix} \hat{\lambda}(t) - \lambda^*(t) \\ \hat{\mu}(t) - \mu^*(t) \end{bmatrix} \right\| \\
& - \beta\epsilon \begin{bmatrix} \lambda^*(t) - \lambda_{\text{prior}} \\ \mu^*(t) - \mu_{\text{prior}} \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}(t) - \lambda^*(t) \\ \hat{\mu}(t) - \mu^*(t) \end{bmatrix} + \sigma_\eta \|\hat{z}(t) - z^*(t)\|_\eta \\
\le\ & -\beta \left(\min\{\Lambda_m(\delta), \eta\epsilon\} - \frac{\sqrt{\eta}}{4}\delta M_{nc}(\delta) \right) \|\hat{z}(t) - z^*(t)\|_\eta^2 \\
& + \beta \left(\beta^{-1}\sigma_\eta + \sqrt{\eta}\epsilon M_d \right) \|\hat{z}(t) - z^*(t)\|_\eta
\end{aligned} \tag{2.52}$$

for $t \in \left[k\tilde{\Delta}, (k+1)\tilde{\Delta}\right] \cap [0, T]$. Then by the condition (2.45), Lemma 2.5 implies

$$\begin{aligned}
\|\hat{z}(t) - z^*(t)\|_\eta \le\ & e^{-\beta\gamma(\delta,\eta,\epsilon)(t - k\tilde{\Delta})} \left(\left\|\hat{z}\left(k\tilde{\Delta}\right) - z^*\left(k\tilde{\Delta}\right)\right\|_\eta - \frac{\beta^{-1}\sigma_\eta + \sqrt{\eta}\epsilon M_d}{\gamma(\delta, \eta, \epsilon)} \right) \\
& + \frac{\beta^{-1}\sigma_\eta + \sqrt{\eta}\epsilon M_d}{\gamma(\delta, \eta, \epsilon)}
\end{aligned} \tag{2.53}$$

for $t \in \left[k\tilde{\Delta}, (k+1)\tilde{\Delta}\right] \cap [0, T]$. Now, if $\left\|\hat{z}\left(k\tilde{\Delta}\right) - z^*\left(k\tilde{\Delta}\right)\right\|_\eta$ is less than or equal to $\left(\beta^{-1}\sigma_\eta + \sqrt{\eta}\epsilon M_d\right)/\gamma(\delta, \eta, \epsilon)$, then (2.53) shows that

$$\|\hat{z}(t) - z^*(t)\|_\eta \le \frac{\beta^{-1}\sigma_\eta + \sqrt{\eta}\epsilon M_d}{\gamma(\delta, \zeta, \nu, \epsilon)}, \qquad t \in \left[k\tilde{\Delta}, (k+1)\tilde{\Delta}\right] \cap [0, T],$$

while if $\left\|\hat{z}\left(k\tilde{\Delta}\right) - z^*\left(k\tilde{\Delta}\right)\right\|_\eta$ is greater than $\left(\beta^{-1}\sigma_\eta + \sqrt{\eta}\epsilon M_d\right)/\gamma(\delta, \eta, \epsilon)$, then (2.53) with $t = k\tilde{\Delta}$ and (2.49) imply

$$\begin{aligned}
\|\hat{z}(t) - z^*(t)\|_\eta &\le \left\|\hat{z}\left(k\tilde{\Delta}\right) - z^*\left(k\tilde{\Delta}\right)\right\|_\eta \\
&\le \|\hat{z}(0) - z^*(0)\|_\eta, \qquad t \in \left[k\tilde{\Delta}, (k+1)\tilde{\Delta}\right] \cap [0, T],
\end{aligned}$$

and we can see that (2.49) for $t \in \left[k\tilde{\Delta}, (k+1)\tilde{\Delta}\right] \cap [0, T]$. By induction (2.49) holds for all $t \in [0, T]$, and particularly we get

$$\|\hat{z}(t) - z^*(t)\|_\eta < \delta$$

for all $t \in [0, T]$. This suggests that (2.52) holds for all $t \in [0, T]$, and finally by Lemma 2.5, we get the desired bound on $\|\hat{z}(t) - z^*(t)\|_\eta$. □

It is interesting to compare Theorem 2.1 and Theorem 2.4. The Taylor expansion (2.33) shows that

$$\rho(\delta,\alpha,\eta,\epsilon) = 1-\alpha\gamma(\delta,\eta,\epsilon)+O\big(\alpha^2\big).$$

Therefore if we set $\alpha = \Delta\beta$ in the condition (2.23a) and let $\Delta \to 0^+$, we will recover the condition (2.45) except the strictness of the inequality. Furthermore, for any $t \in [0,T]$, we have

$$\begin{aligned}\rho^{\lfloor t/\Delta\rfloor}(\delta,\Delta\beta,\eta,\epsilon) &= \Big(1-\Delta\beta\gamma(\delta,\eta,\epsilon)+O\big(\Delta^2\big)\Big)^{\lfloor t/\Delta\rfloor}\\ &= e^{-\beta\gamma(\delta,\eta,\epsilon)t}+O(\Delta),\end{aligned}$$

from which we can also recover (2.47). These observations partially justify that (2.42) indeed gives the correct continuous-time limit of the discrete-time algorithm (2.10).

Notice that the continuous-time tracking error bound (2.47) shares a similar form with the discrete-time tracking error bound (2.24): a constant term

$$\frac{\beta^{-1}\sigma_\eta+\sqrt{\eta}\epsilon M_\lambda}{\gamma(\delta,\eta,\epsilon)}, \tag{2.54}$$

which we still call the *eventual tracking error bound* in the continuous-time limit, plus something that decays exponentially with τ. The eventual tracking error bound can also be split into two parts, the first part $\beta^{-1}\sigma_\eta/\gamma(\delta,\eta,\epsilon)$ being proportional to σ_η, and the second part $\sqrt{\eta}\epsilon M_\lambda/\gamma(\delta,\eta,\epsilon)$ representing the discrepancy introduced by regularization.

Feasible and Optimal Parameters

As can be seen, $\gamma(\delta,\zeta,\epsilon)$ and the bound (2.54) are much easier to analyze than $\rho(\delta,\alpha,\zeta,\epsilon)$ and (2.27). This enables us not only to discuss the existence of feasible parameters but also the structure of the optimal parameters.

Theorem 2.5. *Let*

$$\mathscr{A}_{\mathrm{fp}}(\delta,\beta) := \left\{(\eta,\epsilon)\in\mathbb{R}^2_{++} : \beta^{-1}\sigma_\eta < \delta\,\gamma(\delta,\eta,\epsilon)-\sqrt{\eta}\epsilon M_d\right\}.$$

1. Suppose $\Lambda_m\big(\bar\delta\big) > M_d M_{nc}\big(\bar\delta\big)$ *for some* $\bar\delta > 0$. *Then the set*

$$\mathscr{S}_{\mathrm{fp}} := \left\{(\delta,\beta,\eta,\epsilon)\in\mathbb{R}^4_{++} : (\eta,\epsilon)\in\mathscr{A}_{\mathrm{fp}}(\delta,\beta)\right\}$$

is a nonempty open subset of $\mathbb{R}^4_{++}$.

2. *Let* $\beta > 0$ *and* $\delta > 0$ *be fixed such that* $\mathscr{A}_{\mathrm{fp}}(\delta, \beta)$ *is nonempty, and suppose* $M_d M_{nc}(\delta) > 0$. *Then the minimizer of* (2.54) *over* $(\eta, \epsilon) \in \mathscr{A}_{\mathrm{fp}}(\delta, \beta)$ *exists and is unique, and is equal to* $(\Lambda_m(\delta)/\epsilon^*, \epsilon^*)$ *where* ϵ^* *is the unique minimizer of the unimodal function*

$$b_{\delta,\beta}(\epsilon) := \frac{\beta^{-1}\sigma_{\epsilon^{-1}\Lambda_m(\delta)} + \sqrt{\epsilon\Lambda_m(\delta)}M_d}{\Lambda_m(\delta) - \delta M_{nc}(\delta)\sqrt{\epsilon^{-1}\Lambda_m(\delta)}/4}, \qquad \epsilon > \frac{1}{\Lambda_m(\delta)}\left(\frac{\delta M_{nc}(\delta)}{4\Lambda_m(\delta)}\right)^2.$$

The proof is postponed to Appendix 2.A.

The first part of Theorem 2.5 is the continuous-time counterpart of Theorem 2.2, and the proof uses the same approach as well. Then in Part 2, we proved that when β and δ are fixed, there exists a unique optimal (η^*, ϵ^*) that minimizes the eventual tracking error bound (2.54). We have also shown that η^* is equal to $\Lambda_m(\delta)/\epsilon^*$, which has been suggested in Section 2.3 and now justified in the continuous-time limit. The fact that ϵ^* is the unique minimizer of the unimodal function $b_{\delta,\beta}(\epsilon)$ quantitatively characterizes the trade-off in choosing the regularization parameter ϵ: more regularization makes the Lagrangian better conditioned in the dual variables, but introduces additional errors as a side effect.

We fix β and δ in Part 2 of Theorem 2.5, as a larger β or a smaller δ will always lead to a smaller bound, while in applications β usually cannot be arbitrarily chosen because of practical limitations (e.g., computation or communication delays), and an excessively small δ can also result in violating the condition (2.45).

It should be noted that, the optimal (η^*, ϵ^*) that minimizes the bound (2.54) may not be the optimal parameters that minimizes the tracking error itself, as (2.54) is only an upper bound which can be loose in certain situations. However, the analysis presented here will still be of value and can serve as a guide for choosing the parameters in practice.

Remark 2.6. In the second part of Theorem 2.5, we only consider the case where neither M_d and $M_{nc}(\delta)$ is zero. If one of M_d and $M_{nc}(\delta)$ is zero, the optimal (η^*, ϵ^*) may not satisfy the structure stated in Theorem 2.5, but the analysis is similar and no harder which we omit here. ■

Isolation of the KKT Trajectory

In Section 2.1, we remarked that there could be multiple trajectories of KKT points, and $z^*(t) = (x^*(t), \lambda^*(t), \mu^*(t))$ is only one of these trajectories that is chosen arbitrarily. Then we analyzed the tracking performance of the algorithm (2.10) and its continuous-time limit (2.42), and showed that under the conditions (2.23) or (2.45) a bounded tracking error can be achieved. On the other hand, if the KKT trajectory $z^*(t)$ bifurcates into two or more branches at some time $\tilde{t} \in [0, T]$ and these branches become far away as time proceeds, then we have no way to identify from Theorem 2.1 or Theorem 2.4 which trajectory the algorithm will track. It is also possible that two KKT trajectories come very close to each other at some time, and Theorem 2.1 or Theorem 2.4 cannot distinguish which trajectory the algorithm will track afterwards. Fortunately, as the following theorems show, such possibilities will not occur in some sense under certain conditions.

Theorem 2.6a. *Suppose for some $\delta > 0$ and $\eta > 0$,*

$$\Lambda_m(\delta) - \frac{\sqrt{\eta}}{2}\delta M_{nc}(\delta) > 0. \tag{2.55}$$

Then there is no KKT point in the set

$$\left\{z = (x, \lambda, \mu) : 0 < \|z - z^*(t)\|_\eta \le \delta,\ x \neq x^*(t)\right\}$$

for each $t \in [0, T]$.

In particular, (2.55) *holds if the condition* (2.45) *holds for some $\delta \le 2\eta^{-1/2} M_d$.*

Theorem 2.6b. *Let $z^*_{(i)}(t) = \left(x^*_{(i)}(t), \lambda^*_{(i)}(t), \mu^*_{(i)}(t)\right)$, $i = 1, 2$, be two Lipschitz continuous trajectories of KKT points of* (2.1) *over $t \in [0, T]$, and for each $i = 1, 2$, define $\Lambda^{(i)}_m(\delta)$, $M^{(i)}_{nc}(\delta)$ for the associated trajectories by* (2.29) *and* (2.14)*. Suppose there exist $\delta_{(1)} > 0$, $\delta_{(2)} > 0$ and $\eta > 0$ such that*

$$\Lambda^{(i)}_m\left(\delta_{(i)}\right) - \frac{\sqrt{\eta}}{2}\delta_{(i)} M^{(i)}_{nc}\left(\delta_{(i)}\right) > 0 \tag{2.56}$$

for $i = 1, 2$, and for some $t \in [0, T]$, $x^*_{(1)}(t) \neq x^*_{(2)}(t)$. Then

$$\left\|z^*_{(1)}(t) - z^*_{(2)}(t)\right\|_\eta > \delta_{(1)} + \delta_{(2)}.$$

Proof of Theorem 2.6a. Suppose for some $t \in [0, T]$, there is another KKT point $z^+ = (x^+, \lambda^+, \mu^+)$ satisfying

$$0 < \|z^+ - z^*(t)\|_\eta \le \delta \qquad \text{and} \qquad x^+ \neq x^*(t).$$

By the KKT condition (2.2b), we have

$$\begin{aligned} & h(x^+) - h(x^*(t)) \\ \ge{} & (x^+ - x^*(t))^T \left(-\nabla_x c(x^*(t), t) - \begin{bmatrix} J_{f^{in},x}(x^*(t), t) \\ J_{f^{eq},x}(x^*(t), t) \end{bmatrix}^T \begin{bmatrix} \lambda^*(t) \\ \mu^*(t) \end{bmatrix} \right) \\ ={} & (x^+ - x^*(t))^T \big(- \nabla_x \mathcal{L}^{nc}(x^*(t), \lambda^*(t), \mu^*(t), t) - J_{f^c,x}(x^*(t), t)^T \lambda^*(t) \big), \end{aligned}$$

and

$$
\begin{aligned}
&h(x^*(t)) - h(x^+) \\
&\geq (x^*(t) - x^+)^T \left(-\nabla_x c(x^+, t) - \begin{bmatrix} J_{f^{in},x}(\hat{x}^+, t) \\ J_{f^{eq},x}(\hat{x}^+, t) \end{bmatrix}^T \begin{bmatrix} \lambda^+ \\ \mu^+ \end{bmatrix} \right) \\
&= (x^*(t) - x^+)^T \Bigg(- \nabla_x \mathcal{L}^{nc}(x^+, \lambda^*(t), \mu^*(t), t) \\
&\qquad - J_{f^c,x}(x^+, t)^T \lambda^+ + \begin{bmatrix} J_{f^{nc},x}(x^+, t) \\ J_{f^{eq},x}(x^+, t) \end{bmatrix}^T \begin{bmatrix} \lambda^*(t) - \lambda^+ \\ \mu^*(t) - \mu^+ \end{bmatrix} \Bigg).
\end{aligned}
$$

Thus

$$
\begin{aligned}
0 &\geq (x^+ - x^*(t))^T \Bigg(\nabla_x \mathcal{L}^{nc}(x^+, \lambda^*(t), \mu^*(t), t) - \nabla_x \mathcal{L}^{nc}(x^*(t), \lambda^*(t), \mu^*(t), t) \\
&\qquad + J_{f^c,x}(x^+, t)^T \lambda^+ - J_{f^c,x}(x^*(t), t)^T \lambda^*(t) + \begin{bmatrix} J_{f^{nc},x}(x^+, t) \\ J_{f^{eq},x}(x^+, t) \end{bmatrix}^T \begin{bmatrix} \lambda^+ - \lambda^*(t) \\ \mu^+ - \mu^*(t) \end{bmatrix} \Bigg) \\
&= (x^+ - x^*(t))^T B_{\mathcal{L}^{nc}}(t)(x^+ - x^*(t)) \\
&\qquad + (x^+ - x^*(t))^T \left(J_{f^c,x}(x^+, t)^T \lambda^+ - J_{f^c,x}(x^*(t), t) \lambda^*(t) \right) \\
&\qquad + (x^+ - x^*(t))^T \begin{bmatrix} J_{f^{nc},x}(x^+, t) \\ J_{f^{eq},x}(x^+, t) \end{bmatrix}^T \begin{bmatrix} \lambda^+ - \lambda^*(t) \\ \mu^+ - \mu^*(t) \end{bmatrix},
\end{aligned}
$$

where $B_{\mathcal{L}^{nc}}(t) := \overline{H}_{\mathcal{L}^{nc}}(x^+ - x^*(t), t)$. Also,

$$
\begin{aligned}
&(\lambda^+ - \lambda^*(t))^T \left(f^{in}(x^+, t) - f^{in}(x^*(t), t) \right) \\
&= - \lambda^{+T} f^{in}(x^*(t), t) - \lambda^*(t)^T f^{in}(x^+, t) \geq 0
\end{aligned}
$$

by the complementary slackness condition, and

$$
(\mu^+ - \mu^*(t))^T (f^{eq}(x^+, t) - f^{eq}(x^*(t), t) = 0.
$$

Therefore

$$
\begin{aligned}
0 &\geq (x^+ - x^*(t))^T B_{\mathcal{L}^{nc}}(t)(x^+ - x^*(t)) \\
&\qquad + (x^+ - x^*(t))^T \left(J_{f^c,x}(x^+, t)^T \lambda^+ - J_{f^c,x}(x^*(t), t) \lambda^*(t) \right) \\
&\qquad + \begin{bmatrix} \lambda^+ - \lambda^*(t) \\ \mu^+ - \mu^*(t) \end{bmatrix}^T \left(\begin{bmatrix} J_{f^{nc},x}(x^+, t) \\ J_{f^{eq},x}(x^+, t) \end{bmatrix} (x^+ - x^*(t)) - \begin{bmatrix} f^{in}(x^+, t) - f^{in}(x^*(t), t) \\ f^{eq}(x^+, t) - f^{eq}(x^*(t), t) \end{bmatrix} \right).
\end{aligned}
$$

Notice that

$$
\begin{aligned}
&(x^+ - x^*(t))^T \left(J_{f^c,x}(x^+,t)^T\lambda^+ - J_{f^c,x}(x^*(t),t)\lambda^*(t)\right) \\
&\quad + \begin{bmatrix}\lambda^+ - \lambda^*(t)\\ \mu^+ - \mu^*(t)\end{bmatrix}^T \left(\begin{bmatrix}J_{f^{nc},x}(x^+,t)\\ J_{f^{eq},x}(x^+,t)\end{bmatrix}(x^+ - x^*(t)) - \begin{bmatrix}f^{in}(x^+,t) - f^{in}(x^*(t),t)\\ f^{eq}(x^+,t) - f^{eq}(x^*(t),t)\end{bmatrix}\right) \\
&= \lambda^{+T}\left(f^c(x^*(t),t) - f^c(x^+,t) - J_{f^c,x}(x^+,t)(x^*(t) - x^+)\right) \\
&\quad + \lambda^*(t)^T\left(f^c(x^+,t) - f^c(x^*(t),t) - J_{f^c,x}(x^*(t),t)(x^+ - x^*(t))\right) \\
&\quad - \begin{bmatrix}\lambda^+ - \lambda^*(t)\\ \mu^+ - \mu^*(t)\end{bmatrix}^T \begin{bmatrix}f^{nc}(x^+,t) + J_{f^{nc},x}(x^+,t)(x^*(t) - x^+) - f^{nc}(x^*(t),t)\\ f^{eq}(x^+,t) + J_{f^{eq},x}(x^+,t)(x^*(t) - x^+) - f^{eq}(x^*(t),t)\end{bmatrix} \\
&\geq (x^+ - x^*(t))^T \left(\frac{1}{2}\sum_{i=1}^m \lambda_i^*(t) B_{f_i^c}(t)\right)(x^+ - x^*(t)) \\
&\quad - \begin{bmatrix}\lambda^+ - \lambda^*(t)\\ \mu^+ - \mu^*(t)\end{bmatrix}^T \begin{bmatrix}f^{nc}(x^+,t) + J_{f^{nc},x}(x^+,t)(x^*(t) - x^+) - f^{nc}(x^*(t),t)\\ f^{eq}(x^+,t) + J_{f^{eq},x}(x^+,t)(x^*(t) - x^+) - f^{eq}(x^*(t),t)\end{bmatrix},
\end{aligned}
$$

where we denote $B_{f_i^c}(t) := \overline{H}_{f_i^c}(x^+ - x^*(t), t)$. Therefore

$$
\begin{aligned}
0 &\geq (x^+ - x^*(t))^T \left(B_{\mathcal{L}^{nc}}(t) + \frac{1}{2}\sum_{i=1}^m \lambda_i^*(t) B_{f_i^c}(t)\right)(x^+ - x^*(t)) \\
&\quad - \begin{bmatrix}f^{nc}(x^+,t) + J_{f^{nc},x}(x^+,t)(x^*(t) - x^+) - f^{nc}(x^*(t),t)\\ f^{eq}(x^+,t) + J_{f^{eq},x}(x^+,t)(x^*(t) - x^+) - f^{eq}(x^*(t),t)\end{bmatrix}^T \begin{bmatrix}\lambda^+ - \lambda^*(t)\\ \mu^+ - \mu^*(t)\end{bmatrix} \\
&\geq \Lambda_m(\delta)\|x^+ - x^*(t)\|^2 - \frac{M_{nc}(\delta)}{2}\|x^+ - x^*(t)\|^2 \left\|\begin{bmatrix}\lambda^+ - \lambda^*(t)\\ \mu^+ - \mu^*(t)\end{bmatrix}\right\| \\
&\geq \left(\Lambda_m(\delta) - \frac{\sqrt{\eta}}{2}\delta M_{f^{nc}}(\delta)\right)\|x^+ - x^*(t)\|^2.
\end{aligned}
$$

However, if (2.55) holds, the right-hand side of the above inequality is then positive, leading to a contradiction.

Now we prove that (2.55) holds if (2.45) holds for some $\delta \leq 2\eta^{-1/2}M_d$. We have

$$
\min\{\Lambda_m(\delta), \eta\epsilon\} \leq \frac{\Lambda_m(\delta) + \eta\epsilon}{2},
$$

and so

$$
\Lambda_m(\delta) \geq 2\min\{\Lambda_m(\delta), \eta\epsilon\} - \eta\epsilon.
$$

On the other hand, (2.45) implies that

$$
\begin{aligned}
\min\{\Lambda_m(\delta), \eta\epsilon\} &> \frac{\sqrt{\eta}}{4}\delta M_{nc}(\delta) + \delta^{-1}\left(\beta^{-1}\sigma_\eta + \sqrt{\eta}\epsilon M_d\right) \\
&> \frac{\sqrt{\eta}}{4}\delta M_{nc}(\delta) + \delta^{-1}\sqrt{\eta}\epsilon M_d.
\end{aligned}
$$

Since $\delta \leq 2\eta^{-1/2}M_d$, we have $\delta^{-1}\sqrt{\eta}\epsilon M_d \geq \eta\epsilon/2$, and so

$$\Lambda_m(\delta) > 2\left(\frac{\sqrt{\eta}}{4}\delta M_{nc}(\delta) + \delta^{-1}\sqrt{\eta}\epsilon M_d\right) - \eta\epsilon \geq \frac{\sqrt{\eta}}{2}\delta M_{nc}(\delta).$$

□

Proof of Theorem 2.6b. We temporarily denote $z^*_{(i)}(t) = \big(x^*_{(i)}(t), \lambda^*_{(i)}(t), \mu^*_{(i)}(t)\big)$ by $z_{(i)} = \big(x_{(i)}, \lambda_{(i)}, \mu_{(i)}\big)$ for notational simplicity. Let

$$\tilde{\theta}_1 := \frac{\delta_{(1)}}{\delta_{(1)} + \delta_{(2)}}, \qquad \tilde{\theta}_2 := \frac{\delta_{(2)}}{\delta_{(1)} + \delta_{(2)}}, \qquad \tilde{z} := \tilde{\theta}_2 z_{(1)} + \tilde{\theta}_1 z_{(2)}.$$

Then it can be checked that

$$z_{(2)} - z_{(1)} = -\frac{z_{(1)} - \tilde{z}}{\tilde{\theta}_1} = \frac{z_{(2)} - \tilde{z}}{\tilde{\theta}_2} \tag{2.57}$$

and

$$\left\|\tilde{z} - z_{(1)}\right\|_\eta \leq \delta_{(1)}, \qquad \left\|\tilde{z} - z_{(2)}\right\|_\eta \leq \delta_{(2)}.$$

Now by (2.2b), we have

$$h\big(x_{(2)}, t\big) - h\big(x_{(1)}, t\big) \geq \big(x_{(2)} - x_{(1)}\big)^T \left(-\nabla_x c\big(x_{(1)}, t\big) - \begin{bmatrix} J_{f^{in},x}\big(x_{(1)}, t\big) \\ J_{f^{eq},x}\big(x_{(1)}, t\big) \end{bmatrix}^T \begin{bmatrix} \lambda^*_{(1)} \\ \mu^*_{(1)} \end{bmatrix}\right),$$

$$h\big(x_{(1)}, t\big) - h\big(x_{(2)}, t\big) \geq \big(x_{(1)} - x_{(2)}\big)^T \left(-\nabla_x c\big(x_{(2)}, t\big) - \begin{bmatrix} J_{f^{in},x}\big(x_{(2)}, t\big) \\ J_{f^{eq},x}\big(x_{(2)}, t\big) \end{bmatrix}^T \begin{bmatrix} \lambda^*_{(2)} \\ \mu^*_{(2)} \end{bmatrix}\right).$$

Taking their sum, we get

$$\begin{aligned}
0 \geq{}& \big(x_{(2)} - x_{(1)}\big)^T \Bigg(\nabla_x c\big(x_{(2)}, t\big) - \nabla_x c\big(x_{(1)}, t\big) \\
&+ \begin{bmatrix} J_{f^{in},x}\big(x_{(2)}, t\big) \\ J_{f^{eq},x}\big(x_{(2)}, t\big) \end{bmatrix}^T \begin{bmatrix} \lambda_{(2)} \\ \mu_{(2)} \end{bmatrix} - \begin{bmatrix} J_{f^{in},x}\big(x_{(1)}, t\big) \\ J_{f^{eq},x}\big(x_{(1)}, t\big) \end{bmatrix}^T \begin{bmatrix} \lambda_{(1)} \\ \mu_{(1)} \end{bmatrix}\Bigg) \\
={}& \big(x_{(2)} - x_{(1)}\big)^T \Bigg(\big(\nabla_x c\big(x_{(2)}, t\big) - \nabla_x c\big(\tilde{x}, t\big)\big) - \big(\nabla_x c\big(x_{(1)}, t\big) - \nabla_x c\big(\tilde{x}, t\big)\big) \\
&+ \left(\begin{bmatrix} J_{f^{in},x}\big(x_{(2)}, t\big) \\ J_{f^{eq},x}\big(x_{(2)}, t\big) \end{bmatrix}^T \begin{bmatrix} \lambda_{(2)} \\ \mu_{(2)} \end{bmatrix} - \begin{bmatrix} J_{f^{in},x}\big(\tilde{x}, t\big) \\ J_{f^{eq},x}\big(\tilde{x}, t\big) \end{bmatrix}^T \begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu} \end{bmatrix} \right) \\
&- \left(\begin{bmatrix} J_{f^{in},x}\big(x_{(1)}, t\big) \\ J_{f^{eq},x}\big(x_{(1)}, t\big) \end{bmatrix}^T \begin{bmatrix} \lambda_{(1)} \\ \mu_{(1)} \end{bmatrix} - \begin{bmatrix} J_{f^{in},x}\big(\tilde{x}, t\big) \\ J_{f^{eq},x}\big(\tilde{x}, t\big) \end{bmatrix}^T \begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu} \end{bmatrix} \right)\Bigg).
\end{aligned} \tag{2.58}$$

For the dual variables, by (2.2c),

$$\begin{aligned}0 \geq & -\left(\lambda_{(2)}-\lambda_{(1)}\right)^{T}\left(f^{in}\left(x_{(2)}, t\right)-f^{in}\left(x_{(1)}, t\right)\right) \\ = & -\left(\lambda_{(2)}-\lambda_{(1)}\right)^{T}\left(f^{in}\left(x_{(2)}, t\right)-f^{in}(\tilde{x}, t)-\left(f^{in}\left(x_{(1)}, t\right)-f^{in}(\tilde{x}, t)\right)\right),\end{aligned} \tag{2.59}$$

and also

$$\begin{aligned}0= & -\left(\mu_{(2)}-\mu_{(1)}\right)^{T}\left(f^{eq}\left(x_{(2)}, t\right)-f^{eq}\left(x_{(1)}, t\right)\right) \\ = & -\left(\mu_{(2)}-\mu_{(1)}\right)^{T}\left(f^{eq}\left(x_{(2)}, t\right)-f^{eq}(\tilde{x}, t)-\left(f^{eq}\left(x_{(1)}, t\right)-f^{eq}(\tilde{x}, t)\right)\right).\end{aligned} \tag{2.60}$$

Now, for each $i=1,2$, we have

$$\begin{aligned}& \left(x_{(i)}-\tilde{x}\right)^{T}\Bigg(\nabla_{x} c\left(x_{(i)}, t\right)-\nabla_{x} c(\tilde{x}, t) \\ & \quad+\begin{bmatrix}J_{f^{in}, x}\left(x_{(i)}, t\right) \\ J_{f^{eq}, x}\left(x_{(i)}, t\right)\end{bmatrix}^{T}\begin{bmatrix}\lambda_{(i)} \\ \mu_{(i)}\end{bmatrix}-\begin{bmatrix}J_{f^{in}, x}(\tilde{x}, t) \\ J_{f^{eq}, x}(\tilde{x}, t)\end{bmatrix}^{T}\begin{bmatrix}\tilde{\lambda} \\ \tilde{\mu}\end{bmatrix}\Bigg) \\ & \quad-\begin{bmatrix}\lambda_{(i)}-\tilde{\lambda} \\ \mu_{(i)}-\tilde{\mu}\end{bmatrix}^{T}\begin{bmatrix}f^{in}\left(x_{(i)}, t\right)-f^{in}(\tilde{x}, t) \\ f^{eq}\left(x_{(i)}, t\right)-f^{eq}(\tilde{x}, t)\end{bmatrix} \\ = & \left(x_{(i)}-\tilde{x}\right)^{T}\left(\nabla_{x} \mathcal{L}^{nc}\left(x_{(i)}, \lambda_{(i)}, \mu_{(i)}, t\right)-\nabla_{x} \mathcal{L}^{nc}\left(\tilde{x}, \lambda_{(i)}, \mu_{(i)}, t\right)\right) \\ & \quad+\begin{bmatrix}f^{nc}(\tilde{x}, t)+J_{f^{nc}, x}(\tilde{x}, t)\left(x_{(i)}-\tilde{x}\right)-f^{nc}\left(x_{(i)}, t\right) \\ f^{eq}(\tilde{x}, t)+J_{f^{eq}, x}(\tilde{x}, t)\left(x_{(i)}-\tilde{x}\right)-f^{eq}\left(x_{(i)}, t\right)\end{bmatrix}^{T}\begin{bmatrix}\lambda_{(i)}-\tilde{\lambda} \\ \mu_{(i)}-\tilde{\mu}\end{bmatrix} \\ & \quad+\lambda_{(i)}^{T}\left(f^{c}(\tilde{x}, t)-f^{c}\left(x_{(i)}, t\right)-J_{f^{c}}\left(x_{(i)}, t\right)^{T}\left(\tilde{x}-x_{(i)}\right)\right) \\ & \quad+\tilde{\lambda}^{T}\left(f^{c}\left(x_{(i)}, t\right)-f^{c}(\tilde{x}, t)-J_{f^{c}}(\tilde{x}, t)^{T}\left(x_{(i)}-\tilde{x}\right)\right) \\ \geq & \left(x_{(i)}-\tilde{x}\right)^{T} B^{(i)}\left(x_{(i)}-\tilde{x}\right)-\frac{M_{nc}^{(i)}\left(\delta_{(i)}\right)}{2}\left\|x_{(i)}-\tilde{x}\right\|^{2}\left\|\begin{bmatrix}\lambda_{(i)}-\tilde{\lambda} \\ \mu_{(i)}-\tilde{\mu}\end{bmatrix}\right\|,\end{aligned} \tag{2.61}$$

where

$$\begin{aligned}B^{(i)}:= & \int_{0}^{1} \nabla_{xx}^{2} \mathcal{L}^{nc}\left(x_{(i)}+\theta\left(\tilde{x}-x_{(i)}\right), \lambda_{(i)}, \mu_{(i)}, t\right) d\theta \\ & +\sum_{j=1}^{m} \lambda_{(i), j} \int_{0}^{1}(1-\theta) \nabla_{xx}^{2} f_{j}^{c}\left(x_{(i)}+\theta\left(\tilde{x}-x_{(i)}\right), t\right) d\theta.\end{aligned}$$

By summing (2.58), (2.59), and (2.60), and plugging in (2.57) and (2.61), we can see that

$$\begin{aligned}0 \geq & \sum_{i=1,2} \frac{1}{\tilde{\theta}_{i}}\left(\left(x_{(i)}-\tilde{x}\right)^{T} B^{(i)}\left(x_{(i)}-\tilde{x}\right)-\frac{M_{nc}^{(i)}\left(\delta_{(i)}\right)}{2}\left\|x_{(i)}-\tilde{x}\right\|^{2}\left\|\begin{bmatrix}\lambda_{(i)}-\tilde{\lambda} \\ \mu_{(i)}-\tilde{\mu}\end{bmatrix}\right\|\right) \\ \geq & \sum_{i=1,2} \tilde{\theta}_{i}\left(\Lambda_{m}^{(i)}\left(\delta_{(i)}\right)-\frac{\sqrt{\eta}}{2} \delta_{(i)} M_{nc}\left(\delta_{(i)}\right)\right)\left\|x_{(2)}-x_{(1)}\right\|^{2}.\end{aligned}$$

But (2.56) then implies that the right-hand side of the above inequality is positive, leading to a contradiction. □

Remark 2.7. It should be noted that, the condition (2.55) does not exclude the possibility that at time t, there exists (λ^+, μ^+) such that $(x^*(t), \lambda^+, \mu^+)$ is also a KKT point of (2.1) and $0 < \|(x^*(t), \lambda^+, \mu^+) - (x^*(t), \lambda^*, \mu^*)\|_\eta \leq \delta$, unless we also assume that the optimal dual variable associated with $x^*(t)$ is unique at time t. A typical constraint qualification condition that guarantees the uniqueness of the optimal Lagrange multiplier is the linear independence constraint qualification (LICQ) [92], which cannot be directly used in our setting but can be possibly checked if we write (2.1) in some alternative formulation. ■

Now, let us further assume that the optimal dual variable associated with $x^*(t)$ is unique for all $t \in [0, T]$ as Remark 2.7 points out. Then Theorems 2.6a and 2.6b show that, under certain conditions, the KKT points for a given time will always be isolated. Especially, when the condition (2.45) is satisfied for some δ that is not too large, there is no ambiguity in which of the KKT trajectories will be tracked by the continuous-time algorithm (2.42).

2.5 Summary

In this chapter, we conducted a comprehensive study on the regularized proximal primal-dual gradient method and its continuous-time counterpart for time-varying nonconvex optimization.

For the discrete-time algorithm, we derived sufficient conditions that guarantee bounded tracking error by investigating when the regularized proximal primal-dual gradient step has a contraction-like behavior. The tracking error bounds are directly related to the maximum temporal variability of a KKT trajectory, and also depend on pertinent algorithmic parameters such as the step size and the regularization coefficient.

We then investigated whether there exist algorithmic parameters that guarantee bounded tracking error when the sampling interval is sufficiently small. Specifically, we derived a sufficient condition for the existence of feasible parameters, which, qualitatively, suggests that the problem should be "sufficiently convex" around a KKT trajectory to overcome the nonlinearity of the nonconvex constraints.

The study on feasible parameters suggested analyzing the continuous-time limit of the discrete-time algorithm. We formulated the continuous-time limit as a system of differential inclusions, and established analytical results on the tracking performance of the system of differential inclusions. The continuous-time tracking error bound shares a similar form to the discrete-time tracking error bound. Then, we studied the existence of feasible parameters of the continuous-time counterpart, and also derived structural results on the optimal parameters that minimize the derived tracking error bound, which can serve as guidelines for choosing parameters in practice.

Finally, we derived conditions under which the KKT points for a given time will always be isolated, i.e., bifurcations or merging of KKT trajectories do not happen. These conditions are closely related to the conditions for bounded tracking error derived previously.

2.A Proofs

This section provides the proofs of the theorems that have been skipped in the text.

Proof of Proposition 2.1

We first present some lemmas and their corollaries that will be used in the proof.

Lemma 2.6. *Let g_1 and g_2 be two closed proper convex functions on $\mathbb{R}^k$, and let $g = g_1 + g_2$. Then*

$$\partial g(x) = \partial g_1(x) + \partial g_2(x), \qquad \forall x \in \mathrm{dom}(g)$$

if any one of the following conditions are satisfied:

1. ([82, Theorem 23.8]) $\mathrm{relint}\,\mathrm{dom}(g_1) \cap \mathrm{relint}\,\mathrm{dom}(g_2)$ *is nonempty;*
2. ([75, Theorem 3.16]) *there exists $x \in \mathrm{dom}(g)$ such that g_1 or g_2 is continuous at x.*

Corollary 2.1. *1. Let C_1 and C_2 be two closed convex subsets of $\mathbb{R}^k$. Then $N_{C_1 \cap C_2}(x) = N_{C_1}(x)$ if $x \in \mathrm{int}\, C_2$.*

2. Let $g : \mathbb{R}^k \to \mathbb{R} \cup \{+\infty\}$ be a closed proper convex function. Then $\partial g(x) = \partial g(x) + N_{\mathrm{dom}(g)}(x)$ for any $x \in \mathrm{dom}(g)$.

Proof of Corollary 2.1. The first corollary follows by applying Lemma 2.6 to the indicator function $I_{C_1 \cap C_2} = I_{C_1} + I_{C_2}$. The second corollary follows by noting that $g = g + I_{\mathrm{dom}(g)}$. □

Lemma 2.7 ([82, Theorem 24.4])**.** *Let $g : \mathbb{R}^k \to \mathbb{R} \cup \{+\infty\}$ be a closed proper convex function. Then the graph of $\partial g(x)$, defined by* $\mathrm{graph}(g) := \{(x, y) : x \in \mathrm{dom}(g), y \in \partial g(x)\}$, *is closed.*

Lemma 2.8 ([20, Propositions 4.7.3 and 4.6.3])**.** *Let $\bar{z}$ be a local minimum of a function $g : \mathbb{R}^k \to \mathbb{R}$ over a closed convex subset C of $\mathbb{R}^k$. Assume that g has the form $g = g_1 + g_2$ where g_1 is smooth and g_2 is convex. Then*

$$-\nabla g_1(\bar{z}) \in \partial g_2(\bar{z}) + N_C(\bar{z}).$$

Now we begin the proof of Proposition 2.1. Since $\bar{x}$ is a local minimum, we can find some $\epsilon > 0$ such that for all $x \in \mathcal{X} := \{x \in \mathrm{dom}(h) : \|x - \bar{x}\| \leq \epsilon\}$, we have $c(x)+h(x) \geq c(\bar{x})+h(\bar{x})$ as long as $f(x) \in C$. Obviously $\mathcal{X}$ is compact as $\mathrm{dom}(h)$ is closed. Let $\mathcal{D} := \{y \in C : \|y - f(\bar{x})\| \leq \epsilon\}$. Now consider the following auxiliary problem:

$$\min_{(x,s)\in\mathcal{X}\times\mathcal{D}} F_k(x, s) := c(x) + h(x) + \frac{k}{2}\|f(x) - s\|^2 + \frac{1}{2}\|x - \bar{x}\|^2 \tag{2.62}$$

for each $k \in \mathbb{N}$, and let (x_k, s_k) denote its global minimum. Since $(\bar{x}, f(\bar{x}))$ is a feasible solution to this auxiliary problem, we have

$$c(\bar{x}) + h(\bar{x}) \geq c(x_k) + h(x_k) + \frac{k}{2}\|f(x_k) - s_k\|^2 + \frac{1}{2}\|x_k - \bar{x}\|^2. \tag{2.63}$$

Since $\mathcal{X} \times \mathcal{D}$ is compact, (x_k, s_k) has a convergent subsequence. Let $(\tilde{x}, \tilde{s})$ denote the limit of an arbitrary convergent subsequence of (x_k, s_k). By (2.63) and noticing that c and f are continuous and h is closed (and thus lower semicontinuous) on $\mathcal{X}$, we get $c(\bar{x}) + h(\bar{x}) \geq c(\tilde{x}) + h(\tilde{x}) + \frac{1}{2}\|\tilde{x} - \bar{x}\|^2$ and $f(\tilde{x}) = \tilde{s}$. We then see that $(\tilde{x}, f(\tilde{x}))$ is also a feasible solution to the original problem (2.3), implying that $c(\bar{x}) + h(\bar{x}) \leq c(\tilde{x}) + h(\tilde{x})$. Therefore we have $\tilde{x} = \bar{x}$, meaning that all convergent subsequences of (x_k, s_k) will converge to the same limit, and by the compactness of $\mathcal{X} \times \mathcal{D}$, we get $(x_k, s_k) \to (\bar{x}, f(\bar{x}))$.

By applying Lemma 2.8 to (2.62), we get

$$\begin{aligned} -\nabla c(x_k) - kJ_f(x_k)^T(f(x_k) - s_k) - (x_k - \bar{x}) &\in \partial h(x_k) + N_{\mathcal{X}}(x_k), \\ k(f(x_k) - s_k) &\in N_{\mathcal{D}}(s_k). \end{aligned}$$

Since for large enough $k \in \mathbb{N}$, x_k is in the interior of $\{x \in \mathbb{R}^n : \|x - \bar{x}\| \leq \epsilon\}$ and s_k is in the interior of $\{y \in \mathbb{R}^p : \|y - f(\bar{x})\| \leq \epsilon\}$, we have $N_{\mathcal{X}}(x_k) = N_{\mathrm{dom}(h)}(x_k)$ and $N_{\mathcal{D}}(s_k) = N_C(s_k)$ by the first part of Corollary 2.1. Together with the second part of Corollary 2.1, we see that for large enough k,

$$-\nabla c(x_k) - kJ_f(x_k)^T(f(x_k) - s_k) - (x_k - \bar{x}) \in \partial h(x_k), \tag{2.64a}$$

$$k(f(x_k) - s_k) \in N_C(s_k). \tag{2.64b}$$

If there exists a subsequence of $\left(k(f(x_k) - s_k)\right)_{k\in\mathbb{N}}$ that converges to some $\bar{\lambda}$, then since $\partial h(x)$ and $N_C(s)$ have closed graphs Lemma 2.7, we get

$$-\nabla c(\bar{x}) - J_f(\bar{x})^T \bar{\lambda} \in \partial h(\bar{x}), \tag{2.65a}$$
$$\bar{\lambda} \in N_C(f(\bar{x})), \tag{2.65b}$$

and $\bar{\lambda} \in N_C(f(\bar{x}))$ is equivalent to $f(\bar{x}) \in N_{C^\circ}(\bar{\lambda})$ as can be seen by

$$\begin{aligned}
\bar{\lambda} \in N_C(f(\bar{x})) \quad &\Longleftrightarrow \quad \bar{\lambda}^T(s - f(\bar{x})) \le 0, \quad \forall s \in C \\
&\Longleftrightarrow \quad \bar{\lambda}^T s \le 0, \quad \forall s \in C \quad \text{and} \quad \bar{\lambda}^T f(\bar{x}) = 0 \\
&\Longleftrightarrow \quad \bar{\lambda} \in C^\circ \quad \text{and} \quad f(\bar{x})^T \bar{\lambda} = 0 \\
&\Longleftrightarrow \quad (\lambda - \bar{\lambda})^T f(\bar{x}) \le 0, \quad \forall \lambda \in C^\circ \\
&\Longleftrightarrow \quad f(\bar{x}) \in N_{C^\circ}(\bar{\lambda}),
\end{aligned}$$

where we used the fact that $f(\bar{x}) \in C$ and C is a closed convex cone.

Otherwise, if $\|k(f(x_k) - s_k)\| \to +\infty$ as $k \to \infty$, we define

$$l_k := \|k(f(x_k) - s_k)\|, \qquad y_k := \frac{k(f(x_k) - s_k)}{l_k}.$$

Since $\|y_k\| = 1$ for all $k \in \mathbb{N}$, by extracting a subsequence, we can without loss of generality assume that $y_k \to \tilde{y}$ for some $\tilde{y}$ as $k \to \infty$. Obviously $\tilde{y} \neq 0$, and by (2.64b) and that the graph of $N_C(s)$ is closed, we have $\tilde{y} \in N_C(f(\bar{x}))$. By (2.64a), we see that for any $x' \in \mathrm{dom}(h)$,

$$l_k^{-1}(h(x') - h(x_k)) \ge -\left(l_k^{-1}\left(\nabla c(x_k) + (x_k - \bar{x})\right) + J_f(x_k)^T y_k\right)^T (x' - x_k),$$

and by letting $k \to \infty$, we get $-(x' - \bar{x})^T J_f(\bar{x})^T \tilde{y} \le 0$ for all $x' \in \mathrm{dom}(h)$, i.e.,

$$-J_f(\bar{x})^T \tilde{y} \in N_{\mathrm{dom}(h)}(\bar{x}),$$

which contradicts the constraint qualification condition.

Next we consider the set

$$\Lambda := \left\{\bar{\lambda} \in C^\circ : \bar{\lambda} \text{ satisfies } (2.5)\right\}.$$

By (2.65), it can be easily seen that Λ is convex and closed as $\partial h(\bar{x})$ and $N_C(f(\bar{x}))$ are convex and closed. If Λ is unbounded, then there exists a sequence $(\lambda_k)_{k\in\mathbb{N}} \subset \Lambda$ such that $\|\lambda_k\| \to \infty$. By extracting a subsequence we can assume without loss of generality that $\lambda_k/\|\lambda_k\|$ converges to some $\tilde{\lambda}$ which is in C° since C° is a closed cone. By (2.65), we have

$$\frac{1}{\|\lambda_k\|}\left(h(x)-h(\bar{x})\right) \geq (x-\bar{x})^T\left(-\frac{1}{\|\lambda_k\|}\nabla c(\bar{x}) - J_f(\bar{x})^T\frac{\lambda_k}{\|\lambda_k\|}\right), \quad \forall x \in \mathrm{dom}(h)$$

and

$$\frac{\lambda_k^T f(\bar{x})}{\|\lambda_k\|} = 0$$

for all $k \in \mathbb{N}$. Now let $k \to \infty$, and we get

$$0 \geq -(x-\bar{x})^T J_f(\bar{x})^T\tilde{\lambda}, \quad \forall x \in \mathrm{dom}(h),$$

i.e., $-J_f(\bar{x})^T\tilde{\lambda} \in N_{\mathrm{dom}(h)}(\bar{x})$, and $\tilde{\lambda}^T f(\bar{x}) = 0$, which contradicts the constraint qualification condition. Therefore Λ is bounded. ■

Proof of Lemma 2.2

Denote $h := (x_2 - x_1)/\|x_2 - x_1\|$. By (1.6), for each i, we have

$$\begin{aligned}
&h'^T\left(\nabla_x f_i(x_2,t) - \nabla_x f_i(x_1,t)\right)\\
&= \|x_2 - x_1\|\int_0^1 h'^T\nabla^2_{xx} f_i(x_1 + \theta(x_2 - x_1), t)\, h\, d\theta,
\end{aligned}$$

where h' is any nonzero vector. Therefore

$$\begin{aligned}
&\left\|\left(J_{f,x}(x_2,t) - J_{f,x}(x_1,t)\right) h'\right\|\\
&= \left\|\|x_2 - x_1\|\int_0^1 D^2_{xx} f(x_1 + \theta(x_2 - x_1))(h, h')\, d\theta\right\|\\
&\leq \|x_2 - x_1\|\int_0^1 \left\|D^2_{xx} f(x_1 + \theta(x_2 - x_1))(h, h')\right\| d\theta\\
&\leq \|x_2 - x_1\| \cdot \sup_{\theta\in[0,1]}\left\|D^2_{xx} f(x_1 + \theta(x_2 - x_1), t)\right\| \|h'\|,
\end{aligned}$$

and by the arbitrariness of h', we get (2.18).

Similarly, by (1.7), for each i, we have

$$\begin{aligned}
&f_i(x_2,t) - f_i(x_1,t) - \nabla_x f_i(x_1,t)^T(x_2 - x_1)\\
&= \frac{1}{2}\int_0^1 2(1-\theta)\,(x_2 - x_1)^T\left[\nabla^2_{xx} f_i(x_1 + \theta(x_2 - x_1), t)\right](x_2 - x_1)\, d\theta\\
&= \frac{1}{2}\|x_2 - x_1\|^2\int_0^1 2(1-\theta)\, h^T\left[\nabla^2_{xx} f_i(x_1 + \theta(x_2 - x_1), t)\right] h\, d\theta,
\end{aligned}$$

and so

$$
\begin{aligned}
&f(x_2,t) - f(x_1,t) - J_{f,x}(x_1,t)(x_2 - x_1) \\
&= \frac{1}{2}\|x_2 - x_1\|^2 \int_0^1 2(1-\theta)\, D^2_{xx} f(x_1 + \theta(x_2 - x_1), t)(h,h)\, d\theta,
\end{aligned}
$$

which leads to

$$
\begin{aligned}
&\|f(x_2,t) - f(x_1,t) - J_{f,x}(x_1,t)(x_2 - x_1)\| \\
&\le \frac{1}{2}\|x_2 - x_1\|^2 \int_0^1 2(1-\theta)\left\|D^2_{xx} f(x_1 + \theta(x_2 - x_1), t)(h,h)\right\| d\theta \\
&\le \frac{1}{2}\|x_2 - x_1\|^2 \cdot \sup_{\theta\in[0,1]} \left\|D^2_{xx} f(x_1 + \theta(x_2 - x_1), t)\right\| \int_0^1 2(1-\theta)\, d\theta \\
&= \frac{1}{2}\|x_2 - x_1\|^2 \cdot \sup_{\theta\in[0,1]} \left\|D^2_{xx} f(x_1 + \theta(x_2 - x_1), t)\right\|.
\end{aligned}
$$

■

Proof of Lemma 2.3

We have

$$
\begin{aligned}
&\nabla c_\tau(\hat{x}_{\tau-1}) - \nabla c_\tau(x^*_\tau) + \begin{bmatrix} J_{f^{in}_\tau}(\hat{x}_{\tau-1}) \\ J_{f^{eq}_\tau}(\hat{x}_{\tau-1}) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_{\tau-1} \\ \hat{\mu}_{\tau-1} \end{bmatrix} - \begin{bmatrix} J_{f^{in}_\tau}(x^*_\tau) \\ J_{f^{eq}_\tau}(x^*_\tau) \end{bmatrix}^T \begin{bmatrix} \lambda^*_\tau \\ \mu^*_\tau \end{bmatrix} \\
&= \nabla_x \mathcal{L}^{nc}_\tau(\hat{x}_{\tau-1}, \lambda^*_\tau, \mu^*_\tau) - \nabla_x \mathcal{L}^{nc}_\tau(x^*_\tau, \lambda^*_\tau, \mu^*_\tau) + \begin{bmatrix} J_{f^{nc}_\tau}(\hat{x}_{\tau-1}) \\ J_{f^{eq}_\tau}(\hat{x}_{\tau-1}) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda^*_\tau \\ \hat{\mu}_{\tau-1} - \mu^*_\tau \end{bmatrix} \\
&\quad + J_{f^c_\tau}(\hat{x}_{\tau-1})^T \hat{\lambda}_{\tau-1} - J_{f^c_\tau}(x^*_\tau)^T \lambda^*_\tau \\
&= B_{\mathcal{L}^{nc}_\tau}(\hat{x}_{\tau-1} - x^*_\tau) + \begin{bmatrix} J_{f^{nc}_\tau}(\hat{x}_{\tau-1}) \\ J_{f^{eq}_\tau}(\hat{x}_{\tau-1}) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda^*_\tau \\ \hat{\mu}_{\tau-1} - \mu^*_\tau \end{bmatrix} + J_{f^c_\tau}(\hat{x}_{\tau-1})^T \hat{\lambda}_{\tau-1} - J_{f^c_\tau}(x^*_\tau)^T \lambda^*_\tau,
\end{aligned}
$$

where

$$
B_{\mathcal{L}^{nc}_\tau} := \overline{H}_{\mathcal{L}^{nc}}(\hat{x}_{\tau-1} - x^*_\tau, \tau\Delta).
$$

Then by (2.10) and (2.9a), and using the nonexpansiveness of the proximal operator, we get

$$\begin{aligned}
&\left\|\hat{x}_\tau - x_\tau^*\right\|^2 \\
&\le \Bigg\|(\hat{x}_{\tau-1} - x_\tau^*) - \alpha\Bigg(\nabla c_\tau(\hat{x}_{\tau-1}) - \nabla c_\tau(x_\tau^*) \\
&\qquad + \begin{bmatrix} J_{f_\tau^{in}}(\hat{x}_{\tau-1}) \\ J_{f_\tau^{eq}}(\hat{x}_{\tau-1}) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_{\tau-1} \\ \hat{\mu}_{\tau-1} \end{bmatrix} - \begin{bmatrix} J_{f_\tau^{in}}(x_\tau^*) \\ J_{f_\tau^{eq}}(x_\tau^*) \end{bmatrix}^T \begin{bmatrix} \lambda_\tau^* \\ \mu_\tau^* \end{bmatrix}\Bigg)\Bigg\|^2 \\
&= \Bigg\| \left(I - \alpha B_{\mathcal{L}_\tau^{nc}}\right)(\hat{x}_{\tau-1} - x_\tau^*) \\
&\qquad - \alpha\Bigg(\begin{bmatrix} J_{f_\tau^{nc}}(\hat{x}_{\tau-1}) \\ J_{f_\tau^{eq}}(\hat{x}_{\tau-1}) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix} + J_{f_\tau^c}(\hat{x}_{\tau-1})^T \hat{\lambda}_{\tau-1} - J_{f_\tau^c}(x_\tau^*)^T \lambda_\tau^* \Bigg)\Bigg\|^2. \quad (2.66)
\end{aligned}$$

We have,

$$\begin{aligned}
&\left\| \begin{bmatrix} J_{f_\tau^{nc}}(\hat{x}_{\tau-1}) \\ J_{f_\tau^{eq}}(\hat{x}_{\tau-1}) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix} + J_{f_\tau^c}(\hat{x}_{\tau-1})^T \hat{\lambda}_{\tau-1} - J_{f_\tau^c}(x_\tau^*)^T \lambda_\tau^* \right\| \\
&= \left\| \begin{bmatrix} J_{f_\tau^{in}}(\hat{x}_{\tau-1}) \\ J_{f_\tau^{eq}}(\hat{x}_{\tau-1}) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix} + \left(J_{f_\tau^c}(\hat{x}_{\tau-1}) - J_{f_\tau^c}(x_\tau^*)\right)^T \lambda_\tau^* \right\| \\
&\le \left\| \begin{bmatrix} J_{f_\tau^{in}}(\hat{x}_{\tau-1}) \\ J_{f_\tau^{eq}}(\hat{x}_{\tau-1}) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix} \right\| + \left\|J_{f_\tau^c}(\hat{x}_{\tau-1}) - J_{f_\tau^c}(x_\tau^*)\right\| \sup_{t\in[0,T]} \|\lambda^*(t)\| \\
&\le L_f(\delta) \left\| \begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix} \right\| + M_c(\delta)\left\|\hat{x}_{\tau-1} - x_\tau^*\right\| \sup_{t\in[0,T]} \|\lambda^*(t)\|, \quad (2.67)
\end{aligned}$$

where we used (2.18) and the definitions of $M_c(\delta)$ and $L_f(\delta)$.

For the dual variables, by (2.10b), (2.10c), (2.9b) and (2.2d), and using the nonexpansiveness of projection onto convex sets,

$$\begin{aligned}
&\left\| \begin{bmatrix} \hat{\lambda}_\tau - \lambda_\tau^* \\ \hat{\mu}_\tau - \mu_\tau^* \end{bmatrix} \right\|^2 \\
&= \left\| (1 - \eta\alpha\epsilon) \begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix} - \eta\alpha\epsilon \begin{bmatrix} \lambda_\tau^* - \lambda_{\text{prior}} \\ \mu_\tau^* - \mu_{\text{prior}} \end{bmatrix} + \eta\alpha \begin{bmatrix} f_\tau^{in}(\hat{x}_{\tau-1}) - f_\tau^{in}(x_\tau^*) \\ f_\tau^{eq}(\hat{x}_{\tau-1}) - f_\tau^{eq}(x_\tau^*) \end{bmatrix} \right\|^2 \\
&= \left\| (1 - \eta\alpha\epsilon) \begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix} - \eta\alpha\epsilon \begin{bmatrix} \lambda_\tau^* - \lambda_{\text{prior}} \\ \mu_\tau^* - \mu_{\text{prior}} \end{bmatrix} \right\|^2 + \eta^2\alpha^2 \left\| \begin{bmatrix} f_\tau^{in}(\hat{x}_{\tau-1}) - f_\tau^{in}(x_\tau^*) \\ f_\tau^{eq}(\hat{x}_{\tau-1}) - f_\tau^{eq}(x_\tau^*) \end{bmatrix} \right\|^2 \\
&\quad + 2\eta\alpha \left((1 - \eta\alpha\epsilon) \begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix} - \eta\alpha\epsilon \begin{bmatrix} \lambda_\tau^* - \lambda_{\text{prior}} \\ \mu_\tau^* - \mu_{\text{prior}} \end{bmatrix} \right)^T \begin{bmatrix} f_\tau^{in}(\hat{x}_{\tau-1}) - f_\tau^{in}(x_\tau^*) \\ f_\tau^{eq}(\hat{x}_{\tau-1}) - f_\tau^{eq}(x_\tau^*) \end{bmatrix}.
\end{aligned}$$

Noting that $\|\hat{x}_{\tau-1} - x_\tau^*\| \le \|\hat{z}_{\tau-1} - z_\tau^*\|_\eta \le \delta$, by (1.6) and the definitions of M_d, $L_f(\delta)$, we get

$$\left\|\begin{bmatrix}\hat{\lambda}_\tau - \lambda_\tau^* \\ \hat{\mu}_\tau - \mu_\tau^*\end{bmatrix}\right\|^2 \le \left((1-\eta\alpha\epsilon)\left\|\begin{bmatrix}\hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^*\end{bmatrix}\right\| + \eta\alpha\epsilon M_d\right)^2 + \eta^2\alpha^2 L_f^2(\delta)\|\hat{x}_{\tau-1} - x_\tau^*\|^2$$
$$+ 2\eta\alpha(1-\eta\alpha\epsilon)\begin{bmatrix}\hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^*\end{bmatrix}^T \begin{bmatrix} f_\tau^{in}(\hat{x}_{\tau-1}) - f_\tau^{in}(x_\tau^*) \\ f_\tau^{eq}(\hat{x}_{\tau-1}) - f_\tau^{eq}(x_\tau^*)\end{bmatrix}$$
$$+ 2\eta^2\alpha^2\epsilon M_d L_f(\delta)\|\hat{x}_{\tau-1} - x_\tau^*\|. \tag{2.68}$$

By the convexity of the components of f_τ^c and noting that $\hat{\lambda}_{\tau-1} \in \mathbb{R}_+^m$, we have

$$\hat{\lambda}_{\tau-1}^T\left(f_\tau^c(\hat{x}_{\tau-1}) + J_{f_\tau^c}(\hat{x}_{\tau-1})(x_\tau^* - \hat{x}_{\tau-1}) - f_\tau^c(x_\tau^*)\right) \le 0.$$

In addition, by (1.7), we have

$$\lambda_\tau^{*T}\left(f_\tau^c(x_\tau^*) + J_{f_\tau^c}(x_\tau^*)(\hat{x}_{\tau-1} - x_\tau^*) - f_\tau^c(\hat{x}_{\tau-1})\right)$$
$$= -\frac{1}{2}(\hat{x}_{\tau-1} - x_\tau^*)^T\left(\sum_{i=1}^m \lambda_{\tau,i}^* B_{f_{\tau,i}^c}\right)(\hat{x}_{\tau-1} - x_\tau^*),$$

where we denote $B_{f_{\tau,i}^c} := \overline{H}_{f_i^c}(\hat{x}_{\tau-1} - x_\tau^*, \tau\Delta)$. Therefore,

$$\begin{bmatrix}\hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^*\end{bmatrix}^T \begin{bmatrix} f_\tau^{in}(\hat{x}_{\tau-1}) - f_\tau^{in}(x_\tau^*) \\ f_\tau^{eq}(\hat{x}_{\tau-1}) - f_\tau^{eq}(x_\tau^*)\end{bmatrix}$$
$$- (\hat{x}_{\tau-1} - x_\tau^*)^T\left(\begin{bmatrix} J_{f_\tau^{nc}}(\hat{x}_{\tau-1}) \\ J_{f_\tau^{eq}}(\hat{x}_{\tau-1})\end{bmatrix}^T \begin{bmatrix}\hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^*\end{bmatrix} + J_{f_\tau^c}(\hat{x}_{\tau-1})^T\hat{\lambda}_{\tau-1} - J_{f_\tau^c}(x_\tau^*)^T\lambda_\tau^*\right)$$
$$= \begin{bmatrix}\hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^*\end{bmatrix}^T \begin{bmatrix} f_\tau^{nc}(\hat{x}_{\tau-1}) + J_{f_\tau^{nc}}(\hat{x}_{\tau-1})(x_\tau^* - \hat{x}_{\tau-1}) - f_\tau^{nc}(x_\tau^*) \\ f_\tau^{eq}(\hat{x}_{\tau-1}) + J_{f_\tau^{eq}}(\hat{x}_{\tau-1})(x_\tau^* - \hat{x}_{\tau-1}) - f_\tau^{eq}(x_\tau^*)\end{bmatrix}$$
$$+ \hat{\lambda}_{t-1}^T\left(f_\tau^c(\hat{x}_{\tau-1}) + J_{f_\tau^c}(\hat{x}_{\tau-1})(x_\tau^* - \hat{x}_{\tau-1}) - f_\tau^c(x_\tau^*)\right)$$
$$+ \lambda_\tau^{*T}\left(f_\tau^c(x_\tau^*) + J_{f_\tau^c}(x_\tau^*)(\hat{x}_{\tau-1} - x_\tau^*) - f_\tau^c(\hat{x}_{\tau-1})\right)$$
$$\le \frac{M_{nc}(\delta)}{2}\|\hat{x}_{\tau-1} - x_\tau^*\|^2\left\|\begin{bmatrix}\hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^*\end{bmatrix}\right\| - \frac{1}{2}(\hat{x}_{\tau-1} - x_\tau^*)^T\left(\sum_{i=1}^m \lambda_{\tau,i}^* B_{f_{\tau,i}^c}\right)(\hat{x}_{\tau-1} - x_\tau^*)$$
$$\le \frac{\sqrt{\eta}}{4}\delta M_{nc}(\delta)\|\hat{z}_{\tau-1} - z_\tau^*\|_\eta^2 - \frac{1}{2}(\hat{x}_{\tau-1} - x_\tau^*)^T\left(\sum_{i=1}^m \lambda_{\tau,i}^* B_{f_{\tau,i}^c}\right)(\hat{x}_{\tau-1} - x_\tau^*), \tag{2.69}$$

where we used (2.19) and the definition of $M_{nc}(\delta)$ in the second step, and used

$$\|\hat{x}_{\tau-1} - x_\tau^*\|\left\|\begin{bmatrix}\hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^*\end{bmatrix}\right\| \le \frac{1}{2}\left(\sqrt{\eta}\|\hat{x}_{\tau-1} - x_\tau^*\|^2 + \frac{1}{\sqrt{\eta}}\left\|\begin{bmatrix}\hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^*\end{bmatrix}\right\|^2\right)$$
$$= \frac{\sqrt{\eta}}{2}\|\hat{z}_{\tau-1} - z_\tau^*\|_\eta^2$$

in the last step.

Now we take the sum of (2.66) and (2.68) and use (2.67) to bound $\left\|\hat{z}_\tau - z_\tau^*\right\|_\eta^2$ by

$$
\begin{aligned}
&\left\|\hat{z}_\tau - z_\tau^*\right\|_\eta^2 = \|\hat{x}_\tau - x_\tau^*\|^2 + \eta^{-1}\left\|\begin{bmatrix}\hat{\lambda}_\tau - \lambda_\tau^* \\ \hat{\mu}_\tau - \mu_\tau^*\end{bmatrix}\right\|^2 \\
&\leq (\hat{x}_{\tau-1} - x_\tau^*)^T \left(I - \alpha B_{\mathcal{L}_\tau^{nc}}\right)^2 (\hat{x}_{\tau-1} - x_\tau^*) \\
&\quad + \alpha^2 \left(L_f(\delta)\left\|\begin{bmatrix}\hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^*\end{bmatrix}\right\| + M_c(\delta)\left\|\hat{x}_{\tau-1} - x_\tau^*\right\| \sup_{t\in[0,T]} \|\lambda^*(t)\|\right)^2 \\
&\quad - 2\alpha(\hat{x}_{\tau-1} - x_\tau^*)^T \left(I - \alpha B_{\mathcal{L}_\tau^{nc}}\right)\left(\begin{bmatrix}J_{f_\tau^{nc}}(\hat{x}_{\tau-1}) \\ J_{f_\tau^{eq}}(\hat{x}_{\tau-1})\end{bmatrix}^T \begin{bmatrix}\hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^*\end{bmatrix}\right. \\
&\qquad\qquad\qquad\qquad\qquad\qquad \left. + J_{f_\tau^c}(\hat{x}_{\tau-1})^T \hat{\lambda}_{\tau-1} - J_{f_\tau^c}(x_\tau^*)^T \lambda_\tau^*\right) \\
&\quad + \eta^{-1}\left((1 - \eta\alpha\epsilon)\left\|\begin{bmatrix}\hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^*\end{bmatrix}\right\| + \eta\alpha\epsilon M_d\right)^2 + \eta\alpha^2 L_f^2(\delta)\left\|\hat{x}_{\tau-1} - x_\tau^*\right\|^2 \\
&\quad + 2\alpha(1 - \eta\alpha\epsilon)\begin{bmatrix}\hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^*\end{bmatrix}^T \begin{bmatrix}f_\tau^{in}(\hat{x}_{\tau-1}) - f_\tau^{in}(x_\tau^*) \\ f_\tau^{eq}(\hat{x}_{\tau-1}) - f_\tau^{eq}(x_\tau^*)\end{bmatrix} \\
&\quad + 2\eta\alpha^2\epsilon M_d L_f(\delta)\left\|\hat{x}_{\tau-1} - x_\tau^*\right\|. \qquad (2.70)
\end{aligned}
$$

Notice that

$$
\begin{aligned}
&\alpha^2 \left(L_f(\delta)\left\|\begin{bmatrix}\hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^*\end{bmatrix}\right\| + M_c(\delta)\left\|\hat{x}_{\tau-1} - x_\tau^*\right\| \sup_{t\in[0,T]} \|\lambda^*(t)\|\right)^2 \\
&\qquad + \eta\alpha^2 L_f^2(\delta)\left\|\hat{x}_{\tau-1} - x_\tau^*\right\|^2 \\
&\leq \alpha^2\left(\sqrt{\eta} L_f(\delta) + M_c(\delta)\sup_{t\in[0,T]} \|\lambda^*(t)\|\right)^2 \left\|\hat{z}_{\tau-1} - z_\tau^*\right\|_\eta^2 \\
&= \alpha^2 D^2(\delta, \eta)\left\|\hat{z}_{\tau-1} - z_\tau^*\right\|_\eta^2. \qquad (2.71)
\end{aligned}
$$

Moreover,

$$
\begin{aligned}
&(1-\eta\alpha\epsilon)\begin{bmatrix}\hat{\lambda}_{\tau-1}-\lambda_\tau^* \\ \hat{\mu}_{\tau-1}-\mu_\tau^*\end{bmatrix}^T \begin{bmatrix} f_\tau^{in}(\hat{x}_{\tau-1})-f_\tau^{in}(x_\tau^*) \\ f_\tau^{eq}(\hat{x}_{\tau-1})-f_\tau^{eq}(x_\tau^*)\end{bmatrix} \\
&\quad -(\hat{x}_{\tau-1}-x_\tau^*)^T\left(I-\alpha B_{\mathcal{L}_\tau^{nc}}\right)\left(\begin{bmatrix}J_{f_\tau^{nc}}(\hat{x}_{\tau-1}) \\ J_{f_\tau^{eq}}(\hat{x}_{\tau-1})\end{bmatrix}^T\begin{bmatrix}\hat{\lambda}_{\tau-1}-\lambda_\tau^* \\ \hat{\mu}_{\tau-1}-\mu_\tau^*\end{bmatrix}\right. \\
&\qquad\qquad\qquad\qquad \left. + J_{f_\tau^c}(\hat{x}_{\tau-1})^T\hat{\lambda}_{\tau-1}-J_{f_\tau^c}(x_\tau^*)^T\lambda_\tau^*\right) \\
&=(1-\eta\alpha\epsilon)\left(\begin{bmatrix}\hat{\lambda}_{\tau-1}-\lambda_\tau^* \\ \hat{\mu}_{\tau-1}-\mu_\tau^*\end{bmatrix}^T \begin{bmatrix} f_\tau^{in}(\hat{x}_{\tau-1})-f_\tau^{in}(x_\tau^*) \\ f_\tau^{eq}(\hat{x}_{\tau-1})-f_\tau^{eq}(x_\tau^*)\end{bmatrix}\right. \\
&\quad \left. -(\hat{x}_{\tau-1}-x_\tau^*)^T\left(\begin{bmatrix}J_{f_\tau^{nc}}(\hat{x}_{\tau-1}) \\ J_{f_\tau^{eq}}(\hat{x}_{\tau-1})\end{bmatrix}^T\begin{bmatrix}\hat{\lambda}_{\tau-1}-\lambda_\tau^* \\ \hat{\mu}_{\tau-1}-\mu_\tau^*\end{bmatrix} + J_{f_\tau^c}(\hat{x}_{\tau-1})^T\hat{\lambda}_{\tau-1}-J_{f_\tau^c}(x_\tau^*)^T\lambda_\tau^*\right)\right) \\
&\quad -\alpha(\hat{x}_{\tau-1}-x_\tau^*)^T\left(\eta\epsilon I-B_{\mathcal{L}_\tau^{nc}}\right)\left(\begin{bmatrix}J_{f_\tau^{nc}}(\hat{x}_{\tau-1}) \\ J_{f_\tau^{eq}}(\hat{x}_{\tau-1})\end{bmatrix}^T\begin{bmatrix}\hat{\lambda}_{\tau-1}-\lambda_\tau^* \\ \hat{\mu}_{\tau-1}-\mu_\tau^*\end{bmatrix}\right. \\
&\qquad\qquad\qquad\qquad \left. + J_{f_\tau^c}(\hat{x}_{\tau-1})^T\hat{\lambda}_{\tau-1}-J_{f_\tau^c}(x_\tau^*)^T\lambda_\tau^*\right) \\
&\le (1-\eta\alpha\epsilon)\left(\frac{\sqrt{\eta}}{4}\delta M_{nc}(\delta)\left\|\hat{z}_{\tau-1}-z_\tau^*\right\|_\eta^2-\frac{1}{2}(\hat{x}_{\tau-1}-x_\tau^*)^T\left(\sum_{i=1}^m\lambda_{\tau,i}^*B_{f_{\tau,i}^c}\right)(\hat{x}_{\tau-1}-x_\tau^*)\right) \\
&\quad +\alpha\left\|\eta\epsilon I-B_{\mathcal{L}_\tau^{nc}}\right\|\left\|\hat{x}_{\tau-1}-x_\tau^*\right\| \\
&\qquad \times\left(L_f(\delta)\left\|\begin{bmatrix}\hat{\lambda}_{\tau-1}-\lambda_\tau^* \\ \hat{\mu}_{\tau-1}-\mu_\tau^*\end{bmatrix}\right\|+M_c(\delta)\left\|\hat{x}_{\tau-1}-x_\tau^*\right\|\sup_{t\in[0,T]}\left\|\lambda^*(t)\right\|\right) \\
&\le (1-\eta\alpha\epsilon)\left(\frac{\sqrt{\eta}}{4}\delta M_{nc}(\delta)\left\|\hat{z}_{\tau-1}-z_\tau^*\right\|_\eta^2-\frac{1}{2}(\hat{x}_{\tau-1}-x_\tau^*)^T\left(\sum_{i=1}^m\lambda_{\tau,i}^*B_{f_{\tau,i}^c}\right)(\hat{x}_{\tau-1}-x_\tau^*)\right) \\
&\quad +\alpha\left\|\eta\epsilon I-B_{\mathcal{L}_\tau^{nc}}\right\|D(\delta,\eta)\left\|\hat{z}_{\tau-1}-z_\tau^*\right\|_\eta^2. \qquad (2.72)
\end{aligned}
$$

Therefore by plugging (2.71) and (2.72) into (2.70), we get

$$
\begin{aligned}
&\left\|\hat{z}_\tau - z_\tau^*\right\|_\eta^2 \\
&\leq (\hat{x}_{\tau-1} - x_\tau^*)^T \left[\left(I - \alpha B_{\mathcal{L}_\tau^{nc}}\right)^2 - \alpha(1-\eta\alpha\epsilon) \sum_{i=1}^m \lambda_{\tau,i}^* B_{f_{\tau,i}^c} \right] (\hat{x}_{\tau-1} - x_\tau^*) \\
&\quad + (1-\eta\alpha\epsilon)^2 \eta^{-1} \left\| \begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix} \right\|^2 + \alpha^2 D^2(\delta,\eta) \left\|\hat{z}_{\tau-1} - z_\tau^*\right\|_\eta^2 \\
&\quad + \alpha(1-\eta\alpha\epsilon)\frac{\sqrt{\eta}}{2}\delta M_{nc}(\delta) \left\|\hat{z}_{\tau-1} - z_\tau^*\right\|^2 + 2\alpha^2 \left\|\eta\epsilon I - B_{\mathcal{L}_\tau^{nc}}\right\| D(\delta,\eta) \left\|\hat{z}_{\tau-1} - z_\tau^*\right\|_\eta^2 \\
&\quad + 2\sqrt{\eta}\alpha\epsilon M_d \left(\frac{1-\eta\alpha\epsilon}{\sqrt{\eta}} \left\| \begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix} \right\| + \sqrt{\eta}\alpha L_f(\delta) \left\|\hat{x}_{\tau-1} - x_\tau^*\right\| \right) \\
&\quad + \eta\alpha^2\epsilon^2 M_d^2.
\end{aligned}
$$

It's not hard to see that

$$
\begin{aligned}
&\left(\frac{1-\eta\alpha\epsilon}{\sqrt{\eta}} \left\| \begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix} \right\| + \sqrt{\eta}\alpha L_f(\delta) \left\|\hat{x}_{\tau-1} - x_\tau^*\right\| \right) \\
&\leq \max\left\{1-\eta\alpha\epsilon,\ \sqrt{\eta}\alpha L_f(\delta)\right\} \left\|\hat{z}_{\tau-1} - z_\tau^*\right\|_\eta,
\end{aligned}
$$

and by the definition of $\rho(\delta,\alpha,\eta,\epsilon)$ and $\kappa(\delta,\alpha,\eta,\epsilon)$, we get

$$
\begin{aligned}
&\left\|\hat{z}_\tau - z_\tau^*\right\|_\eta^2 \\
&\leq \rho^2(\delta,\alpha,\eta,\epsilon) \left\|\hat{z}_{\tau-1} - z_\tau^*\right\|_\eta^2 + \eta\alpha^2\epsilon^2 M_d^2 \\
&\quad + 2\sqrt{\eta}\alpha\epsilon M_d \cdot \max\left\{1-\eta\alpha\epsilon, \sqrt{\eta}\alpha L_f(\delta)\right\} \left\|\hat{z}_{\tau-1} - z_\tau^*\right\|_\eta \\
&\leq \left(\rho(\delta,\alpha,\eta,\epsilon) \left\|\hat{z}_{\tau-1} - z_\tau^*\right\|_\eta + \kappa(\delta,\alpha,\eta,\epsilon)\sqrt{\eta}\alpha\epsilon M_d\right)^2,
\end{aligned}
$$

which is just (2.22).

Now let δ, η and ϵ be fixed. With the help of Lemma 2.4, it can be shown that, if we temporarily allow α to take arbitrary values in $\mathbb{R}$, then the function $\alpha \mapsto \rho^{(\mathrm{P})}(\delta,\alpha,\eta,\epsilon)$ is a continuous function over $\alpha \in \mathbb{R}$, and so

$$
\lim_{\alpha\to 0^+} \rho^{(\mathrm{P})}(\delta,\alpha,\alpha\eta,\epsilon) = \rho^{(\mathrm{P})}(\delta,0,\eta,\epsilon) = 1.
$$

Then by the definition of $\rho(\delta,\alpha,\eta,\epsilon)$, it's straightforward to get $\rho(\delta,\alpha,\eta,\epsilon) \to 1$ when $\alpha \to 0^+$, which further leads to

$$
\lim_{\alpha\to 0^+} \kappa(\delta,\alpha,\eta,\epsilon) = 1.
$$

We also have

$$\frac{\max\left\{1-\eta\alpha\epsilon, \sqrt{\eta}\alpha L_f(\delta)\right\}}{\rho(\delta,\alpha,\eta,\epsilon)} \leq \frac{1-\eta\alpha\epsilon+\alpha D(\delta,\eta)}{\rho(\delta,\alpha,\eta,\epsilon)}$$
$$\leq \frac{\sqrt{2\left((1-\alpha\eta\epsilon)^2+\alpha^2 D^2(\delta,\eta)\right)}}{\rho(\delta,\alpha,\eta,\epsilon)} \leq \sqrt{2},$$

which implies that $\kappa(\delta,\alpha,\eta,\epsilon) \leq \sqrt{2}$. ■

Proof of Lemma 2.4

Part 1. Let $a \in \mathbb{R}$ be arbitrary, and consider the set

$$\begin{aligned} g^{-1}[(a,+\infty)] &= \{y \in Y : g(y) > a\} \\ &= \{y \in Y : \exists x \in X \text{ such that } f(x,y) > a\} \\ &= \bigcup_{x\in X}\{y \in Y : f(x,y) > a\}. \end{aligned}$$

The continuity of f implies that $\{y \in Y : f(x,y) > a\}$ is open for each $x \in X$, and so $g^{-1}[(a,+\infty)]$ is open. By the arbitrariness of $a \in \mathbb{R}$, we see that g is lower semicontinuous.

Now let $y^0 \in Y$ be arbitrary, and let $(y_n)_{n\in\mathbb{N}}$ be any sequence in Y such that $y_n \to y^0$ and $\lim_{n\to\infty} g(y_n)$ exists. Since X is compact and f is continuous, we can see that for any n, there is some $x_n \in X$ such that $g(y_n) = f(x_n, y_n)$. By the compactness of X, we can find a subsequence $(x_{k_n})_{n\in\mathbb{N}}$ such that $x_{k_n} \to x^0$ for some $x^0 \in X$ as $n \to \infty$. Then

$$\lim_{n\to\infty} g(y_n) = \lim_{n\to\infty} f(x_n, y_n) = \lim_{n\to\infty} f(x_{k_n}, y_{k_n}) = f(x^0, y^0) \leq g(y^0),$$

where the third equality follows from the continuity of f, and the last inequality follows from the definition of g. By the arbitrariness of the sequence $(y_n)_{n\in\mathbb{N}}$, we get

$$\limsup_{y\to y^0} g(y) \leq g(y^0),$$

where $y^0 \in Y$ is arbitrary. Now we can conclude that f is upper semicontinuous, and thus continuous on Y.

Part 2. Now suppose $f : R\mathcal{B}_n \times V \to \mathbb{R}$ is continuous. Define the auxiliary function $\tilde{f} : R\mathcal{B}_n \times (0,R) \times V \to \mathbb{R}$ by

$$\tilde{f}(u,r,v) = f\left(\mathcal{P}_{r\mathcal{B}_n}(u), v\right).$$

The function $(u, r) \mapsto \mathcal{P}_{r\mathcal{B}_n}(u)$ is continuous (in fact Lipschitz), as for any (u_1, r_1) and (u_2, r_2) in $R\mathcal{B}_n \times (0, R)$, we have

$$\begin{aligned}
&\left\|\mathcal{P}_{r_1\mathcal{B}_n}(u_1) - \mathcal{P}_{r_2\mathcal{B}_n}(u_2)\right\| \\
\leq\ &\left\|\mathcal{P}_{r_1\mathcal{B}_n}(u_1) - \mathcal{P}_{r_1\mathcal{B}_n}(u_2)\right\| + \left\|\mathcal{P}_{r_1\mathcal{B}_n}(u_2) - \mathcal{P}_{r_2\mathcal{B}_n}(u_2)\right\| \\
\leq\ &\|u_1 - u_2\| + |r_1 - r_2|.
\end{aligned}$$

Therefore $\tilde{f}$ is also a continuous function. Moreover,

$$g(r, v) = \sup_{u:\|u\|\leq r} f(u, v) = \sup_{u\in R\mathcal{B}_n} \tilde{f}(u, r, v).$$

By the compactness of $R\mathcal{B}_n$ and the first part of Lemma 2.4, we conclude that g is continuous. ■

Proof of Lemma 2.5

The proof is directly based on the following lemma.

Lemma 2.9 ([94]). *Let I be a closed interval with zero as left endpoint. Let $u(t)$ be a continuous nonnegative function that satisfies the integral inequality*

$$u(t) \leq u_0 + \int_0^t w(s)u^p(s)\, ds,$$

where $w(t)$ is a continuous nonnegative function on I. For $0 \leq p < 1$ we have

$$u(t) \leq \left(u_0^{1-p} + (1-p)\int_0^t w(s)\, ds\right)^{\frac{1}{1-\beta}},$$

and for $p = 1$, we have

$$u(t) \leq u_0 \exp \int_0^t w(s)\, ds.$$

Let us define $u(t) = e^{2\beta t} v^2(t)$. Then

$$\begin{aligned}
\frac{d}{dt}u(t) &= 2\beta e^{2\beta t} v^2(t) + e^{2\beta t}\frac{d}{dt}\left(v^2(t)\right) \\
&\leq 2\beta e^{2\beta t} v^2(t) + 2e^{2\beta t}(\alpha v(t) - \beta v^2(t)) \\
&= 2\alpha e^{2\beta t} v(t) = 2\alpha e^{\beta t}\sqrt{u(t)}
\end{aligned}$$

for almost all $t \in [0, T]$. Therefore by Lemma 2.9,

$$u(t) \leq \left(\sqrt{u(0)} + \alpha \int_0^t e^{\beta s}\, ds\right)^2 = \left(\sqrt{u(0)} + \frac{\alpha}{\beta}(e^{\beta t} - 1)\right)^2,$$

and by the definition of $u(t)$, we get the desired result. ■

Proof of Theorem 2.3

The following definitions and lemmas from set-valued analysis are needed.

Definition 2.3. Let X and Y be topological spaces. We say that $F : X \to 2^Y$ is a *set-valued map* from X to Y. The *domain* of F is defined by $\mathrm{dom}(F) := \{x \in X : F(x) \neq \emptyset\}$. The *graph* of F is defined by $\mathrm{graph}(F) := \{(x, y) \in X \times Y : y \in F(x)\}$

We say that the set-valued map F is *closed* if its graph is a closed subset of $X \times Y$.

We say that the set-valued map F is *upper semicontinuous* at $x \in \mathrm{dom}(F)$ if for any open subset $U \subseteq Y$ such that $F(x) \subseteq U$, there exists an open subset $V \subseteq X$ containing x such that for all $x' \in V$, we have $F(x') \subseteq U$.

Lemma 2.10 ([7, Proposition 1.4.9])**.** *Let X and Y be two topological spaces, and let F and G be two set-valued maps from X to Y. Assume that F is closed, that $G(x)$ is compact and that G is upper semicontinuous at $x \in \mathrm{dom}(F \cap G)$. Then $F \cap G$ is upper semicontinuous at x.*

Lemma 2.11 ([7, Theorem 7.2.2])**.** *Let X be a topological vector space, Y a Banach space and F be a nontrivial set-valued map from X to Y. We assume that F is upper semicontinuous on its domain.*

Let us consider measurable functions $x_m(\cdot)$ and $y_m(\cdot)$ from Ω to X and Y respectively, satisfying: for almost all $\omega \in \Omega$ and for all neighborhoods U of 0 in the product space $X \times Y$, there exists $M := M(\omega, U)$ such that

$$\forall m > M,\ (x_m(\omega), y_m(\omega)) \in \mathrm{graph}(F) + U.$$

If we assume that

1. *$x_m(\cdot)$ converges almost everywhere to a function $x(\cdot)$,*
2. *$y_m(\cdot) \in L^1(\Omega; Y)$ and converges weakly in $L^1(\Omega; Y)$ to a function $y \in L^1(\Omega; Y)$,*

then for almost all $\omega \in \Omega$ such that $x(\omega) \in \mathrm{dom}(F)$, $y(\omega) \in \mathrm{cl}\,\mathrm{conv}\, F(x(\omega))$.

The following lemma provides conditions for the closedness of the partial subdifferential.

Lemma 2.12. *Suppose we are given $g : \mathbb{R}^p \times [0, T] \to \mathbb{R} \cup \{+\infty\}$ satisfying*

1. *$\mathrm{graph}_t(g)$ is κ-Lipschitz in t, where $\mathrm{graph}_t(g) := \{(z, g(z, t)) : z \in \mathrm{dom}_t(g)\}$;*
2. *for each fixed $t \in [0, T]$, $g(\cdot, t)$ is convex and proper, and is continuous when restricted to its domain.*

Then the set

$$\{(z,t,y) \in \mathbb{R}^p \times [0,T] \times \mathbb{R}^p : z \in \mathrm{dom}_t(g),\, y \in \partial_z g(z,t)\}$$

is closed; in other words, the set-valued map $(z,t) \mapsto \partial_z g(z,t)$ *is closed.*

Proof of Lemma 2.12. Denote

$$A = \{(z,t,y) \in \mathbb{R}^p \times [0,T] \times \mathbb{R}^p : z \in \mathrm{dom}_t(g),\, y \in \partial_z g(z,t)\}\,.$$

Let (z_k, t_k, y_k), $k \in \mathbb{N}$, be a sequence in A that converges to some (z,t,y). We then have

$$\begin{aligned}
\inf_{u \in \mathrm{dom}_t(g)} \|z-u\| &\leq \|z-z_k\| + \inf_{u \in \mathrm{dom}_t(g)} \|z_k - u\| \\
&\leq \|z-z_k\| + \inf_{u \in \mathrm{dom}_t(g)} \|(z_k, g(z_k,t_k)) - (u, g(u,t))\| \\
&\leq \|z-z_k\| + d_H\left(\mathrm{graph}_t(g), \mathrm{graph}_{t_k}(g)\right) \\
&\leq \|z-z_k\| + \kappa|t-t_k|,
\end{aligned}$$

where $k \in \mathbb{N}$ is arbitrary. By letting $k \to \infty$ we get $\inf_{u \in \mathrm{dom}_t(g)} \|z-u\| = 0$, and since $\mathrm{dom}_t(g)$ is closed, we have $z \in \mathrm{dom}_t(g)$.

Next, for each $k \in \mathbb{N}$, let

$$u_k \in \operatorname*{arg\,min}_{u \in \mathrm{dom}_t(g)} \|(u, g(u,t)) - (z_k, g(z_k,t))\|.$$

We have $\|(u_k, g(u_k,t)) - (z_k, g(z_k,t_k))\| \leq \kappa|t-t_k|$ since $\mathrm{graph}_t(g)$ is κ-Lipschitz with respect to t. Therefore

$$\begin{aligned}
\|u_k - z\| &\leq \|u_k - z_k\| + \|z_k - z\| \\
&\leq \|(u_k, g(u_k,t)) - (z_k, g(z_k,t_k))\| + \|z_k - z\| \\
&\leq \kappa|t-t_k| + \|z_k - z\|,
\end{aligned}$$

which implies that u_k converges to z since (z_k,t_k) converges to (z,t). Then by taking the limit of

$$\begin{aligned}
\|g(z_k,t_k) - g(z,t)\| &\leq \|g(z_k,t_k) - g(u_k,t)\| + \|g(u_k,t) - g(z,t)\| \\
&\leq \kappa|t-t_k| + \|g(u_k,t) - g(z,t)\|,
\end{aligned}$$

and using the continuity of $g(\cdot,t)$, we get $g(z_k,t_k) \to g(z,t)$.

Now let $w \in \mathrm{dom}_t(g)$ be arbitrary, and let

$$w_k \in \operatorname*{arg\,min}_{u \in \mathrm{dom}_{t_k}(g)} \|(u, g(u, t_k)) - (w, g(w, t))\|.$$

We have $\|(w_k, g(w_k, t_k)) - (w, g(w, t))\| \leq \kappa |t - t_k|$ as $\mathrm{graph}_t(g)$ is κ-Lipschitz with respect to t. Then

$$\begin{aligned}
&\begin{bmatrix} y_k \\ -1 \end{bmatrix}^T \begin{bmatrix} w - z_k \\ g(w, t) - g(z_k, t_k) \end{bmatrix} \\
= &\begin{bmatrix} y_k \\ -1 \end{bmatrix}^T \left(\begin{bmatrix} w - w_k \\ g(w, t) - g(w_k, t_k) \end{bmatrix} + \begin{bmatrix} w_k - z_k \\ g(w_k, t_k) - g(z_k, t_k) \end{bmatrix} \right) \\
\leq &\begin{bmatrix} y_k \\ -1 \end{bmatrix}^T \begin{bmatrix} w - w_k \\ g(w, t) - g(w_k, t_k) \end{bmatrix} \leq (\|y_k\| + 1) \cdot \kappa |t - t_k|,
\end{aligned}$$

where we used

$$y_k \in \partial_z g(z_k, t_k) \iff y_k^T (u - z_k) \leq g(u, t_k) - g(z_k, t_k), \quad \forall u \in \mathrm{dom}_{t_k}(g).$$

By letting $k \to \infty$ and noting that $z_k \to z$, $y_k \to y$ and $g(z_k, t_k) \to g(z, t)$, we get

$$\begin{bmatrix} y \\ -1 \end{bmatrix}^T \begin{bmatrix} w - z \\ g(w, t) - g(z, t) \end{bmatrix} \leq 0,$$

or $y^T (w - z) \leq g(w, t) - g(z, t)$. By the arbitrariness of $w \in \mathrm{dom}_t(g)$, we see that $y \in \partial_z g(z, t)$, and therefore $(z, t, y) \in A$. □

The following lemma constitutes the core step of the proof of Theorem 2.3. It can be viewed as a generalization of [28, Theorem 4.1′] where the normal cone is replaced by the subdifferential, and the proof uses essentially the same techniques as in [2, 28].

Lemma 2.13. *Suppose* $\Phi : \mathbb{R}^p \times [0, T] \to \mathbb{R}^p$ *and* $g : \mathbb{R}^p \times [0, T] \to \mathbb{R} \cup \{+\infty\}$*, and that*

1. $\mathrm{graph}_t(g)$ *is* κ_1*-Lipschitz in* t*, and for each fixed* $t \in [0, T]$*,* $g(\cdot, t)$ *is a proper convex function,*
2. *there exists* $\ell > 0$ *such that*
$$\sup_{z_1, z_2 \in \mathrm{dom}_t(g)} \frac{|g(z_2, t) - g(z_1, t)|}{\|z_2 - z_1\|} \leq \ell$$
for every $t \in [0, T]$*. In other words,* $g(z, t)$ *is uniformly Lipschitz continuous with respect to* z*,*

3. Φ *is continuous when restricted to the set* $\bigcup_{t\in[0,T]} \mathrm{dom}_t(g) \times [0,T]$*, and there exists some* $\kappa_2 > 0$ *such that*

$$\|\Phi(z,t)\| \leq \kappa_2(1+\|z\|), \qquad \forall (z,t) \in \bigcup_{t\in[0,T]} \mathrm{dom}_t(g) \times [0,T].$$

Let $\hat{z}_0 \in \mathrm{dom}_0(g)$ *be arbitrary, and for each* $K \in \mathbb{N}$*, Define* $\hat{z}_\tau^{(K)}$*,* $\tau \in \{0,1,2,\ldots,K\}$ *by*

$$\begin{aligned}
\hat{z}_0^{(K)} &= \hat{z}_0, \\
\hat{z}_\tau^{(K)} &= \mathrm{prox}_{\Delta_K g(\cdot,\tau\Delta_K)} \left[\hat{z}_{\tau-1}^{(K)} + \Delta_K \Phi\left(\hat{z}_{\tau-1}^{(K)}, \tau\Delta_K\right)\right],
\end{aligned}$$

where $\Delta_K := T/K$*, and for* $t \in [0,T]$*, define*

$$\hat{z}^{(K)}(t) = \frac{\tau\Delta_K - t}{\Delta_K}\hat{z}_{\tau-1}^{(K)} + \frac{t-(\tau-1)\Delta_K}{\Delta_K}\hat{z}_\tau^{(K)} \tag{2.73}$$

if $t \in [(\tau-1)\Delta_K, \tau\Delta_K]$*. Then, if we keep* T *constant and let* $K \to \infty$*, the sequence* $\left(\hat{z}^{(K)}\right)_{K\in\mathbb{N}}$ *defined in* (2.73) *has a convergent subsequence, and any convergent subsequence converges uniformly to a Lipschitz continuous* $\hat{z}$ *that satisfies*

$$\begin{aligned}
\hat{z}(0) &= \hat{z}_0, \\
-\frac{d}{dt}\hat{z}(t) + \Phi(\hat{z}(t),t) &\in \partial_z g(\hat{z}(t),t), \quad \forall t \in [0,T]\, a.e.
\end{aligned} \tag{2.74}$$

Proof of Lemma 2.13. Let $A = \bigcup_{t\in[0,T]} \mathrm{dom}_t(g)$, $g_\tau^K(\cdot) = g(\cdot,\tau\Delta_K)$. For each $\tau \geq 1$, let

$$\begin{aligned}
u_\tau^{(K)} &= \mathrm{prox}_{\Delta_K g_\tau^K}\left[\hat{z}_{\tau-1}^{(K)}\right], \\
v_\tau^{(K)} &\in \underset{v\in\mathrm{dom}\left(g_\tau^K\right)}{\arg\min} \left\|(v, g_\tau^K(v)) - \left(\hat{z}_{\tau-1}^{(K)}, g_{\tau-1}^K\left(\hat{z}_{\tau-1}^{(K)}\right)\right)\right\|.
\end{aligned}$$

We have

$$\Delta_K g_\tau^K\left(u_\tau^{(K)}\right) + \frac{1}{2}\left\|u_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\|^2 \leq \Delta_K g_\tau^K\left(v_\tau^{(K)}\right) + \frac{1}{2}\left\|v_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\|^2$$

by definition, which leads to

$$\begin{aligned}
\left\|u_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\|^2 &\leq \left\|v_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\|^2 + 2\Delta_K\left(g_\tau^K\left(v_\tau^{(K)}\right) - g_\tau^K\left(u_\tau^{(K)}\right)\right) \\
&\leq \left\|v_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\|^2 + 2\Delta_K\ell\left\|v_\tau^{(K)} - u_\tau^{(K)}\right\| \\
&\leq \left\|v_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\|^2 + 2\Delta_K\ell\left(\left\|v_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\| + \left\|u_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\|\right),
\end{aligned}$$

and since $\left\|v_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\| \leq \kappa_1\Delta_K$ by the κ_1-Lipschitz continuity of $\mathrm{graph}_t(g)$ in t, we get

$$\left\|u_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\|^2 - 2\Delta_K\ell\left\|u_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\| \leq \Delta_K^2\left(\kappa_1^2 + 2\kappa_1\ell\right),$$

which implies that

$$\left\|u_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\| \leq \Delta_K(\kappa_1 + 2\ell).$$

Then

$$\begin{aligned}
\left\|\hat{z}_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\| \leq & \left\|\text{prox}_{\Delta_K g_\tau^K}\left[\hat{z}_{\tau-1}^{(K)} + \Delta_K \Phi\left(\hat{z}_{\tau-1}^{(K)}, \tau\Delta_K\right)\right] - \text{prox}_{\Delta_K g_\tau^K}\left[\hat{z}_{\tau-1}^{(K)}\right]\right\| \\
& + \left\|\text{prox}_{\Delta_K g_\tau^K}\left[\hat{z}_{\tau-1}^{(K)}\right] - \hat{z}_{\tau-1}^{(K)}\right\| \\
\leq & \Delta_K \left\|\Phi\left(\hat{z}_{\tau-1}^{(K)}, \tau\Delta_K\right)\right\| + \left\|u_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\| \\
\leq & \Delta_K(\kappa_1 + \kappa_2 + 2\ell) + \Delta_K\kappa_2\left\|\hat{z}_{\tau-1}^{(K)}\right\|, \qquad (2.75)
\end{aligned}$$

and so

$$\left\|\hat{z}_\tau^{(K)}\right\| \leq \Delta_K(\kappa_1 + \kappa_2 + 2\ell) + (1 + \Delta_K\kappa_2)\left\|\hat{z}_{\tau-1}^{(K)}\right\|$$

for any $\tau = 1, \ldots, K$. By induction we can see that

$$\left\|\hat{z}_\tau^{(K)}\right\| \leq (1 + \Delta_K\kappa_2)^\tau \left(\|\hat{z}_0\| + \frac{\kappa_1 + \kappa_2 + 2\ell}{\kappa_2}\right) - \frac{\kappa_1 + \kappa_2 + 2\ell}{\kappa_2}$$

holds for all $\tau = 0, \ldots, K$, and since $\Delta_K = T/K$, we get

$$\begin{aligned}
\left\|\hat{z}_\tau^{(K)}\right\| &\leq \left(1 + \frac{T\kappa_2}{K}\right)^K \left(\|\hat{z}_0\| + \frac{(\kappa_1 + \kappa_2 + 2\ell)}{\kappa_2}\right) - \frac{\kappa_1 + \kappa_2 + 2\ell}{\kappa_2} \\
&\leq e^{T\kappa_2}\left(\|\hat{z}_0\| + \frac{(\kappa_1 + \kappa_2 + 2\ell)}{\kappa_2}\right) - \frac{\kappa_1 + \kappa_2 + 2\ell}{\kappa_2} =: \kappa_3
\end{aligned}$$

for any $\tau = 0, 1, \ldots, K$. By plugging it back to (2.75), we have

$$\left\|\hat{z}_\tau^{(K)} - \hat{z}_{\tau-1}^{(K)}\right\| \leq \Delta_K(\kappa_1 + \kappa_2 + \kappa_2\kappa_3 + 2\ell),$$

and consequently

$$\left\|\frac{d}{dt}\hat{z}^{(K)}(t)\right\| = \Delta_K^{-1}\left\|\hat{z}_{\lceil t/\Delta_K\rceil}^{(K)} - \hat{z}_{\lfloor t/\Delta_K\rfloor}^{(K)}\right\| \leq \kappa_1 + \kappa_2 + \kappa_2\kappa_3 + 2\ell =: \tilde{\ell}$$

for almost every $t \in [0, T]$.

Let $D\hat{z}_i^{(K)} \in L^\infty([0, T])$ denote the weak derivative of the i'th entry of $\hat{z}^{(K)}$ for each $i = 1, \ldots, p$. Then the sequence $\left(D\hat{z}_i^{(K)}\right)_{K\in\mathbb{N}}$ lies in the ball

$$B = \left\{f \in L^\infty([0, T]) : \operatorname*{ess\,sup}_{t\in[0,T]} |f(t)| \leq \tilde{\ell}\right\}.$$

The Banach–Alaoglu theorem [83] indicates that B is weak* sequentially compact, and so $\left(D\hat{z}_i^{(K)}\right)_{K\in\mathbb{N}}$ has a convergent subsequence with respect to the weak* topology.

We extract an arbitrary convergent subsequence and still denote it by $\big(D\hat{z}_i^{(K)}\big)_{K\in\mathbb{N}}$. Then $D\hat{z}_i^{(K)} \xrightarrow{w^*} q_i$ for some $q_i \in B$, or in other words,

$$\int_0^T u(t)q_i(t)\,dt = \lim_{K\to\infty}\int_0^T u(t)D\hat{z}_i^{(K)}(t)\,dt$$

for all $u \in L^1([0,T])$. Consequently $\hat{z}^{(K)}$ converges uniformly to $\hat{z}$ given by

$$\hat{z}(t) = \hat{z}_0 + \int_0^t q(s)\,ds,$$

where $q : [0,T] \to \mathbb{R}^p$ is the vector-valued function with entries $g_1, \ldots, g_p$.

Next we prove that any convergent subsequence of $\big(\hat{z}^{(K)}\big)_{K\in\mathbb{N}}$ converges to a limit that satisfies the differential inclusions (2.74). We still use $\hat{z}$ to denote the limit of an arbitrary convergent subsequence of $\big(\hat{z}^{(K)}\big)_{K\in\mathbb{N}}$, and without loss of generality we assume $\hat{z}^{(K)} \to \hat{z}$ by extracting the subsequence. Since $\big(\hat{z}^{(K)}\big)_{K\in\mathbb{N}}$ is equi-Lipschitz, the convergence $\hat{z}^{(K)} \to \hat{z}$ is uniform, $\hat{z}$ is $\tilde{\ell}$-Lipschitz, and $D\hat{z}^{(K)} \xrightarrow{w^*} D\hat{z}$. Define

$$\delta^{(K)}(t) = \left\lfloor \frac{t}{\Delta_K} \right\rfloor \Delta_K, \qquad \theta^{(K)}(t) = \left\lceil \frac{t}{\Delta_K} \right\rceil \Delta_K.$$

Then we have for almost all $t \in [0,T]$,

$$\begin{aligned}
&-\frac{d}{dt}\hat{z}^{(K)}(t) + \Phi\Big(\hat{z}^{(K)}\big(\delta^{(K)}(t)\big), \theta^{(K)}(t)\Big)\\
&= \frac{1}{\Delta_K}\left(\hat{z}^{(K)}\big(\delta^{(K)}(t)\big) + \Delta_K\Phi\Big(\hat{z}^{(K)}\big(\delta^{(K)}(t)\big), \theta^{(K)}(t)\Big) - \hat{z}^{(K)}\big(\theta^{(K)}(t)\big)\right)\\
&\in \partial_z g\Big(\hat{z}^{(K)}\big(\theta^{(K)}(t)\big), \theta^{(K)}(t)\Big),
\end{aligned}$$

where the second fact in Lemma 2.1 is used in the last step. In addition, for almost all $t \in [0,T]$,

$$\left\|-\frac{d}{dt}\hat{z}^{(K)}(t) + \Phi\Big(\hat{z}^{(K)}\big(\delta^{(K)}(t)\big), \theta^{(K)}(t)\Big)\right\| \le \tilde{\ell} + \kappa_2(1+\kappa_3),$$

and so for almost all $t \in [0,T]$,

$$\begin{aligned}
&-\frac{d}{dt}\hat{z}^{(K)}(t) + \Phi\Big(\hat{z}^{(K)}\big(\delta^{(K)}(t)\big), \theta^{(K)}(t)\Big)\\
&\in \partial_z g\Big(\hat{z}^{(K)}\big(\theta^{(K)}(t)\big), \theta^{(K)}(t)\Big) \cap \big(\tilde{\ell} + \kappa_2(1+\kappa_3)\big)\mathcal{B}_p.
\end{aligned}$$

Now we define the set-valued map

$$F(z,t) = \partial_z g(z,t) \cap \big(\tilde{\ell} + \kappa_2(1+\kappa_3)\big)\mathcal{B}_p, \qquad (z,t) \in \mathbb{R}^p \times [0,T].$$

By Lemma 2.12 and Lemma 2.10, the set-valued map F is upper semicontinuous. We then have

$$-\frac{d}{dt}\hat{z}^{(K)}(t) + \Phi\Big(\hat{z}^{(K)}\big(\delta^{(K)}(t)\big), \theta^{(K)}(t)\Big) \in F\Big(\hat{z}^{(K)}\big(\theta^{(K)}(t)\big), \theta^{(K)}(t)\Big).$$

Noticing that

$$\lim_{K\to\infty} \delta^{(K)}(t) = \lim_{K\to\infty} \theta^{(K)}(t) = t,$$

$$\lim_{K\to\infty} \Phi\Big(\hat{z}^{(K)}(\delta^{(K)}(t)), \theta^{(K)}(t)\Big) = \Phi(\hat{z}(t), t),$$

by Lemma 2.11, we can conclude that, for almost all $t \in [0, T]$,

$$-\frac{d}{dt}\hat{z}(t) + \Phi(\hat{z}(t), t) \in F(\hat{z}(t), t),$$

which implies (2.74). □

Now we are ready to finish the proof of Theorem 2.3. For each $t \in [0, T]$ and $z = (x, \lambda, \mu) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{m'}$, we define

$$\Phi(z, t) := \beta \begin{bmatrix} -\nabla_x c(x,t) - J_{f^{in},x}(x,t)^T \lambda - J_{f^{eq},x}(x,t)^T \mu \\ \eta\left(f^{in}(x,t) - \epsilon(\lambda - \lambda_{\text{prior}})\right) \\ \eta\left(f^{eq}(x,t) - \epsilon(\mu - \mu_{\text{prior}})\right) \end{bmatrix} \tag{2.76}$$

and

$$g(z, t) := \beta h(x, t) + I_{\mathbb{R}^m_+}(\lambda).$$

Since g is a separable sum, we have [74]

$$\begin{aligned} \text{prox}_{\Delta_K g(\cdot,t)}(x, \lambda, \mu) &= \Big(\text{prox}_{\Delta_K \beta h(\cdot,t)}(x),\ \text{prox}_{\Delta_K I_{\mathbb{R}^m_+}}(\lambda), \mu\Big) \\ &= \Big(\text{prox}_{\Delta_K \beta h(\cdot,t)}(x),\ \mathcal{P}_{\mathbb{R}^m_+}(\lambda), \mu\Big) \end{aligned}$$

for each $t \in [0, T]$ and $x \in \text{dom}_t(h)$, $\lambda \in \mathbb{R}^m_+$, $\mu \in \mathbb{R}^{m'}$. The iterations (2.37) can then be formulated as

$$\begin{aligned} \hat{z}_0^{(K)} &= \hat{z}_0, \\ \hat{z}_\tau^{(K)} &= \text{prox}_{\Delta_K g(\cdot,\tau\Delta_K)}\left[\hat{z}_{\tau-1}^{(K)} + \Delta_K \Phi\Big(\hat{z}_{\tau-1}^{(K)}, \tau\Delta_K\Big)\right]. \end{aligned}$$

We check the conditions of Lemma 2.13 as follows:

1. $g(z,t)$ is $\max\{1,\beta\}\kappa_1$-Lipschtz with respect to t, as

$$\begin{aligned}
&\inf_{z\in \mathrm{dom}_{t_2}(g)} \|(z, g(z,t_2)) - (z_1, g(z_1,t_1)\| \\
\leq\ & \inf_{\substack{x\in \mathrm{dom}_{t_2}(h),\\ \lambda\in\mathbb{R}_+^m, \mu\in\mathbb{R}^{m'}}} (\|(x, \beta h(x,t_2)) - (x_1, \beta h(x_1,t_1))\| + \|(\lambda,\mu) - (\lambda_1,\mu_1)\|) \\
\leq\ & \inf_{x\in \mathrm{dom}_{t_2}(h)} \max\{1,\beta\}\|(x, h(x,t_2)) - (x_1, h(x_1,t_1))\| \\
\leq\ & \max\{1,\beta\}\kappa_1|t_2 - t_1|
\end{aligned}$$

for any $t_1, t_2 \in [0,T]$ and $z_1 = (x_1,\lambda_1,\mu_1) \in \mathrm{dom}_{t_1}(g)$.

2. By (2.39), we have

$$\begin{aligned}
&\sup_{z_1,z_2\in\mathrm{dom}_t(g)} \frac{|g(z_2,t) - g(z_1,t)|}{\|z_2 - z_1\|} \\
=& \sup_{x_1,x_2\in\mathrm{dom}_t(h)} \beta\frac{|h(x_2,t) - h(x_1,t)|}{\|x_2 - x_1\|} \leq \beta\ell
\end{aligned}$$

for all $t \in [0,T]$.

3. Φ is obviously continuous on $\bigcup_{t\in[0,T]} \mathrm{dom}_t(g) \times [0,T]$. Moreover,

$$\begin{aligned}
\|\Phi(z,t)\| \leq \beta &\left(\|\nabla_x c(x,t)\| + \left\|\begin{bmatrix} J_{f^{in},x}(x,t) \\ J_{f^{eq},x}(x,t)\end{bmatrix}\right\| \left\|\begin{bmatrix}\lambda \\ \mu\end{bmatrix}\right\|\right) \\
&+ \eta\beta \left\|\begin{bmatrix} f^{in}(x,t) \\ f^{eq}(x,t)\end{bmatrix}\right\| + \eta\beta\epsilon \left\|\begin{bmatrix}\lambda \\ \mu\end{bmatrix}\right\| + \eta\beta\epsilon \left\|\begin{bmatrix}\lambda_{\mathrm{prior}} \\ \mu_{\mathrm{prior}}\end{bmatrix}\right\|.
\end{aligned}$$

Let $x_{\mathrm{aux}} \in \bigcup_{t\in[0,T]} \mathrm{dom}_t(h)$ be arbitrary, and

$$\begin{aligned}
\kappa_3 &:= \sup\left\{\left\|\begin{bmatrix} J_{f^{in},x}(x,t) \\ J_{f^{eq},x}(x,t)\end{bmatrix}\right\| : (x,t) \in \bigcup_{t\in[0,T]} \mathrm{dom}_t(h) \times [0,T]\right\}, \\
\kappa_4 &:= \sup_{t\in[0,T]} \left\|\begin{bmatrix} f^{in}(x_{\mathrm{aux}},t) \\ f^{eq}(x_{\mathrm{aux}},t)\end{bmatrix}\right\|,
\end{aligned}$$

both of which are finite. Then

$$\left\|\begin{bmatrix} f^{in}(x,t) \\ f^{eq}(x,t)\end{bmatrix}\right\| \leq \kappa_4 + \kappa_3(\|x\| + \|x_{\mathrm{aux}}\|).$$

By (2.40) and noticing that $\|x\| \leq \|z\|$ and $\|(\lambda,\mu)\| \leq \|z\|$, we get

$$\begin{aligned}
\|\Phi(z,t)\| &\leq \beta(\kappa_2(1 + \|z\|) + \kappa_3\|z\|) \\
&\quad + \eta\beta\left(\kappa_4 + \kappa_3(\|z\| + \|x_{\mathrm{aux}}\|)\right) + \eta\beta\epsilon\|z\| + \eta\beta\epsilon\left\|(\lambda_{\mathrm{prior}}, \mu_{\mathrm{prior}})\right\| \\
&\leq \kappa_5(1 + \|z\|),
\end{aligned}$$

where κ_5 satisfies

$$\kappa_5 \geq \beta\kappa_2 + \eta\beta(\kappa_4 + \kappa_3\|x_{\text{aux}}\|) + \eta\beta\epsilon\|(\lambda_{\text{prior}}, \mu_{\text{prior}})\|,$$
$$\kappa_5 \geq \beta(\kappa_2 + \kappa_3) + \eta\beta\kappa_3 + \eta\beta\epsilon.$$

By Lemma 2.13, the sequence of trajectories defined by (2.73) [and consequently (2.38)] then has convergent subsequences each of which converges to some Lipschitz continuous solution to (2.74). Let $\hat{z}(t) = (\hat{x}(t), \hat{\lambda}(t))$ denote one such solution. By (2.74) and the definition of Φ, we have

$$\left(-\frac{d}{dt}\begin{bmatrix}\hat{\lambda}(t)\\ \hat{\mu}(t)\end{bmatrix} + \eta\beta\begin{bmatrix}f^{in}(\hat{x}(t),t)\\ f^{eq}(\hat{x}(t),t)\end{bmatrix} - \eta\beta\epsilon\begin{bmatrix}\hat{\lambda}(t)-\lambda_{\text{prior}}\\ \hat{\mu}(t)-\mu_{\text{prior}}\end{bmatrix}\right)^T\begin{bmatrix}\lambda_{\text{prior}}-\hat{\lambda}(t)\\ \mu_{\text{prior}}-\hat{\mu}(t)\end{bmatrix} \leq 0,$$

which implies that

$$\begin{aligned}
&\frac{1}{2}\frac{d}{dt}\left\|\begin{bmatrix}\hat{\lambda}(t)-\lambda_{\text{prior}}\\ \hat{\mu}(t)-\mu_{\text{prior}}\end{bmatrix}\right\|^2\\
&\leq \eta\beta\begin{bmatrix}f^{in}(\hat{x}(t),t)\\ f^{eq}(\hat{x}(t),t)\end{bmatrix}^T\begin{bmatrix}\hat{\lambda}(t)-\lambda_{\text{prior}}\\ \hat{\mu}(t)-\mu_{\text{prior}}\end{bmatrix} - \eta\beta\epsilon\left\|\begin{bmatrix}\hat{\lambda}(t)-\lambda_{\text{prior}}\\ \hat{\mu}(t)-\mu_{\text{prior}}\end{bmatrix}\right\|^2\\
&\leq \eta\beta\kappa_2\left\|\begin{bmatrix}\hat{\lambda}(t)-\lambda_{\text{prior}}\\ \hat{\mu}(t)-\mu_{\text{prior}}\end{bmatrix}\right\| - \eta\beta\epsilon\left\|\begin{bmatrix}\hat{\lambda}(t)-\lambda_{\text{prior}}\\ \hat{\mu}(t)-\mu_{\text{prior}}\end{bmatrix}\right\|^2.
\end{aligned}$$

By Lemma 2.5,

$$\left\|\begin{bmatrix}\hat{\lambda}(t)-\lambda_{\text{prior}}\\ \hat{\mu}(t)-\mu_{\text{prior}}\end{bmatrix}\right\| \leq e^{-\eta\beta\epsilon t}\left\|\begin{bmatrix}\hat{\lambda}_0-\lambda_{\text{prior}}\\ \hat{\mu}_0-\mu_{\text{prior}}\end{bmatrix}\right\| + \frac{\kappa_2}{\epsilon}\left(1-e^{-\eta\beta\epsilon t}\right),$$

and then by triangle inequality we see that $\left\|(\hat{\lambda}(t), \hat{\mu}(t))\right\|$ is always strictly less than

$$R_d := \left\|\begin{bmatrix}\hat{\lambda}_0-\lambda_{\text{prior}}\\ \hat{\mu}_0-\mu_{\text{prior}}\end{bmatrix}\right\| + \left\|\begin{bmatrix}\lambda_{\text{prior}}\\ \mu_{\text{prior}}\end{bmatrix}\right\| + \frac{\kappa_2}{\epsilon}.$$

Now suppose $\hat{z}_{(1)} = \left(\hat{x}_{(1)}, \hat{\lambda}_{(1)}, \hat{\mu}_{(1)}\right)$ and $\hat{z}_{(2)} = \left(\hat{x}_{(2)}, \hat{\lambda}_{(2)}, \hat{\mu}_{(2)}\right)$ are two solutions to the differential inclusions (2.74). Then for any $t \in [0, T]$,

$$
\begin{aligned}
&\left\|\Phi\big(\hat{z}_{(2)}(t), t\big) - \Phi\big(\hat{z}_{(1)}(t), t\big)\right\| \\
&\leq \beta \left\|\nabla_x c\big(\hat{x}_{(2)}(t), t\big) - \nabla_x c\big(\hat{x}_{(1)}(t), t\big)\right\| \\
&\quad + \beta \left\| \begin{bmatrix} J_{f^{in},x}\big(\hat{x}_{(2)}(t), t\big) - J_{f^{in},x}\big(\hat{x}_{(1)}(t), t\big) \\ J_{f^{eq},x}\big(\hat{x}_{(2)}(t), t\big) - J_{f^{eq},x}\big(\hat{x}_{(1)}(t), t\big) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_{(2)}(t) \\ \hat{\mu}_{(2)}(t) \end{bmatrix} \right\| \\
&\quad + \beta \left\| \begin{bmatrix} J_{f^{in},x}\big(\hat{x}_{(1)}(t), t\big) \\ J_{f^{eq},x}\big(\hat{x}_{(1)}(t), t\big) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_{(2)}(t) - \hat{\lambda}_{(1)}(t) \\ \hat{\mu}_{(2)}(t) - \hat{\mu}_{(1)}(t) \end{bmatrix} \right\| \\
&\quad + \eta\beta \left\| \begin{bmatrix} f^{in}\big(\hat{x}_{(2)}(t), t\big) - f^{in}\big(\hat{x}_{(1)}(t), t\big) \\ f^{eq}\big(\hat{x}_{(2)}(t), t\big) - f^{eq}\big(\hat{x}_{(1)}(t), t\big) \end{bmatrix} \right\| + \eta\beta\epsilon \left\| \begin{bmatrix} \hat{\lambda}_{(2)}(t) - \hat{\lambda}_{(1)}(t) \\ \hat{\mu}_{(2)}(t) - \hat{\mu}_{(1)}(t) \end{bmatrix} \right\| \\
&\leq \beta l(t) \left\|\hat{x}_{(2)}(t) - \hat{x}_{(1)}(t)\right\| + \beta l(t) \left\|\hat{x}_{(2)}(t) - \hat{x}_{(1)}(t)\right\| \cdot R_d \\
&\quad + \beta(\kappa_3 + \eta\epsilon) \left\| \begin{bmatrix} \hat{\lambda}_{(2)}(t) - \hat{\lambda}_{(1)}(t) \\ \hat{\mu}_{(2)}(t) - \hat{\mu}_{(1)}(t) \end{bmatrix} \right\| + \eta\beta\kappa_3 \left\|\hat{x}_{(2)}(t) - \hat{x}_{(1)}(t)\right\| \\
&\leq \beta\left((1 + R_d) l(t) + (1 + \eta)\kappa_3 + \eta\epsilon\right) \left\|\hat{z}_{(2)}(t) - \hat{z}_{(1)}(t)\right\|,
\end{aligned}
$$

where we used (2.41). Denote $\tilde{l}(t) = \beta\left((1 + R_d) l(t) + (1 + \eta)\kappa_3 + \eta\epsilon\right)$. Obviously $\tilde{l}(t)$ is nonnegative and integrable. Now by (2.74),

$$
-\frac{d}{dt}\hat{z}_{(i)}(t) + \Phi\big(\hat{z}_{(i)}(t), t\big) \in \partial_z g\big(\hat{z}_{(i)}(t), t\big), \quad \forall t \in [0, T] \; a.e.
$$

with $\hat{z}_{(i)}(t) = \hat{z}_0$ for $i = 1, 2$. We then have

$$
\begin{aligned}
g\big(\hat{z}_{(2)}(t), t\big) - g\big(\hat{z}_{(1)}(t), t\big) &\geq \left(-\frac{d}{dt}\hat{z}_{(1)}(t) + \Phi\big(\hat{z}_{(1)}(t), t\big)\right)^T \big(\hat{z}_{(2)}(t) - \hat{z}_{(1)}(t)\big), \\
g\big(\hat{z}_{(1)}(t), t\big) - g\big(\hat{z}_{(2)}(t), t\big) &\geq \left(-\frac{d}{dt}\hat{z}_{(2)}(t) + \Phi\big(\hat{z}_{(2)}(t), t\big)\right)^T \big(\hat{z}_{(1)}(t) - \hat{z}_{(2)}(t)\big)
\end{aligned}
$$

for almost all $t \in [0, T]$. By taking their sum, we get

$$
\left(\frac{d}{dt}\big(\hat{z}_{(2)}(t) - \hat{z}_{(1)}(t)\big) - \Big(\Phi\big(\hat{z}_{(2)}(t), t\big) - \Phi\big(\hat{z}_{(1)}(t), t\big)\Big)\right)^T \big(\hat{z}_{(2)}(t) - \hat{z}_{(1)}(t)\big) \leq 0,
$$

or

$$
\begin{aligned}
\frac{1}{2}\frac{d}{dt}\left\|\hat{z}_{(2)}(t) - \hat{z}_{(1)}(t)\right\|^2 &= \left(\frac{d}{dt}\big(\hat{z}_{(2)}(t) - \hat{z}_{(1)}(t)\big)\right)^T \big(\hat{z}_{(2)}(t) - \hat{z}_{(1)}(t)\big) \\
&\leq \Big(\Phi\big(\hat{z}_{(2)}(t), t\big) - \Phi\big(\hat{z}_{(1)}(t), t\big)\Big)^T \big(\hat{z}_{(2)}(t) - \hat{z}_{(1)}(t)\big) \\
&\leq \hat{l}(t) \left\|\hat{z}_{(2)}(t) - \hat{z}_{(1)}(t)\right\|^2.
\end{aligned}
$$

By Lemma 2.9, $\left\|\hat{z}_{(2)}(t) - \hat{z}_{(1)}(t)\right\|^2 = 0$ for every $t \in [0, T]$, i.e., the two solutions are identical.

The uniqueness of the absolute continuous solution to (2.74) also implies that any convergent subsequence of $\left(\hat{z}^{(K)}\right)_{k\in\mathbb{N}}$ converges uniformly to the same limit $\hat{z}$, and consequently any convergent subsequence of $\left(D\hat{z}_i^{(K)}\right)_{k\in\mathbb{N}}$ converges to the same limit $D\hat{z}_i$ with respect to the weak* topology for each $i = 1, \ldots, p$. In the proof of Lemma 2.13, we see that $\left(D\hat{z}_i^{(K)}\right)_{k\in\mathbb{N}}$ and $D\hat{z}_i$ lie in the set

$$B = \left\{ f \in L^\infty([0,T]) : \operatorname*{ess\,sup}_{t\in[0,T]} |f(t)| \leq \tilde{\ell} \right\}$$

for some $\tilde{\ell} > 0$ for each i, and the Banach–Alaoglu theorem [83] ensures that B is a compact and metrizable with respect to the weak* topology, we can then conclude that $\left(D\hat{z}_i^{(K)}\right)_{k\in\mathbb{N}}$ converges to $D\hat{z}_i$ with respect to the weak* topology for every $i = 1, \ldots, p$, and $\left(\hat{z}^{(K)}\right)_{k\in\mathbb{N}}$ converges uniformly to $\hat{z}$. ■

Proof of Theorem 2.5

We first prove that σ_η is a continuous function of η over $\eta \in \mathbb{R}_{++}$. Let $t_1, t_2 \in [0, T]$ with $t_1 \neq t_2$ be given, and define

$$\begin{aligned}\overline{v}(x; t_1, t_2) &:= \frac{\|z^*(t_2) - z^*(t_1)\|_{x^{-2}}}{|t_2 - t_1|} \\ &= \frac{\left(\|x^*(t_2) - x^*(t_1)\|^2 + x^2\left(\|\lambda^*(t_2) - \lambda^*(t_1)\|^2 + \|\mu^*(t_2) - \mu^*(t_1)\|^2\right)\right)^{1/2}}{|t_2 - t_1|}\end{aligned}$$

for $x \in \mathbb{R}_{++}$. Obviously $\overline{v}(x; t_1, t_2)$ is a convex function of x over $x \in \mathbb{R}_{++}$. Then since $\sigma_{x^{-2}}$ is the supremum of $v(x; t_1, t_2)$ over $\{(t_1, t_2) \in [0, T]^2 : t_1 \neq t_2\}$, $\sigma_{x^{-2}}$ is also a convex function of x over $x \in \mathbb{R}_{++}$. As the domain of $\sigma_{x^{-2}}$ is $\mathbb{R}_{++}$ which is open, we can conclude that $\sigma_{x^{-2}}$ is a continuous function of x over $x \in \mathbb{R}_{++}$, and consequently σ_η is a continuous function of η over $\eta \in \mathbb{R}_{++}$.

Part 1. The proof uses the same approach as in proving Theorem 2.2. Let $R > \bar{\delta}$ be arbitrary, and define

$$f_R(\delta, \beta, \eta, \epsilon) = \delta\gamma(\delta, \eta, \epsilon) - \sqrt{\eta}\epsilon M_d - \beta^{-1}\sigma_\eta.$$

We consider two cases.

1. $M_d \neq 0$: Let $\delta_0 = \bar{\delta}$ and

$$\eta_0 = \left(\frac{2\Lambda_m(\delta_0)}{\delta_0 M_{nc}(\delta_0)}\right)^2, \qquad \epsilon_0 = \frac{\Lambda_m(\delta_0)}{\eta_0}.$$

We then have

$$f_R(\delta_0, \beta, \eta_0, \epsilon_0) = \frac{\bar{\delta}}{2}\left(\Lambda_m(\bar{\delta}) - M_d M_{nc}(\bar{\delta})\right) - \beta^{-1}\sigma_{\eta_0}$$

Since $\Lambda_m(\bar{\delta}) > M_d M_{nc}(\bar{\delta})$, we can find sufficiently large $\beta_0 > 0$ so that $f_R(\delta_0, \beta_0, \eta_0, \epsilon_0) > 0$, implying that $\mathscr{S}_{\mathrm{fp}}$ is nonempty.

2. $M_d = 0$: Let $\eta_0 > 0$ be arbitrary, and let $\epsilon_0 = \eta_0^{-1}\Lambda_m(\delta_0)$. Then

$$\gamma(\delta, \eta_0, \epsilon_0) = \Lambda_m(\delta) - \frac{\sqrt{\eta_0}}{4}\delta M_{nc}(\delta).$$

By the monotonicity of $\Lambda_m(\delta)$ and $M_{nc}(\delta)$,

$$\lim_{\delta\to 0^+} \gamma(\delta, \eta_0, \epsilon_0) = \lim_{\delta\to 0^+} \Lambda_m(\delta) \geq \Lambda_m\left(\bar{\delta}\right) > 0.$$

Therefore there exists some $\delta_0 \in \left(0, \bar{\delta}\right]$ such that $\gamma(\delta_0, \eta_0, \epsilon_0) > 0$, and we have

$$f_R(\delta_0, \beta, \eta_0, \epsilon_0) = \delta_0\gamma(\delta_0, \eta_0, \epsilon_0) - \beta^{-1}\sigma_{\eta_0}.$$

Therefore we can find sufficiently large $\beta_0 > 0$ such that $f_R(\delta_0, \beta_0, \eta_0, \epsilon_0)$ is positive, and consequently $\mathscr{S}_{\mathrm{fp}}$ is non-empty.

Finally, by Lemma 2.4 and the continuity of $\eta \mapsto \sigma_\eta$, it can be seen that $f_R(\delta, \beta, \eta, \epsilon)$ is a continuous function over $(\delta, \beta, \eta, \epsilon) \in (0, R) \times \mathbb{R}^3_{++}$. Therefore the set

$$\mathscr{S}_{\mathrm{fp}} \cap \left((0, R) \times \mathbb{R}^3_{++}\right) = \{(\delta, \beta, \eta, \epsilon) \in (0, R) \times \mathbb{R}^3_{++} : f_R(\delta, \beta, \eta, \epsilon) > 0\}$$

is an open subset of $\mathbb{R}^4_{++}$, and consequently

$$\mathscr{S}_{\mathrm{fp}} = \bigcup_{R > \bar{\delta}} \mathscr{S}_{\mathrm{fp}} \cap \left((0, R) \times \mathbb{R}^3_{++}\right)$$

is an open subset of $\mathbb{R}^4_{++}$.

Part 2. Denote

$$\begin{aligned} g_0(\eta, \epsilon) &:= \frac{\beta^{-1}\sigma_\eta + \sqrt{\eta}\epsilon M_d}{\gamma(\delta, \eta, \epsilon)}, \\ g_1(\eta, \epsilon) &:= \delta\, \gamma(\delta, \eta, \epsilon) - \sqrt{\eta}\epsilon M_d - \beta^{-1}\sigma_\eta, \end{aligned}$$

where $(\eta, \epsilon) \in \mathbb{R}^2_{++}$. Obviously $g_0(\eta, \epsilon) = \delta$ if $g_1(\eta, \epsilon) = 0$, and $g_0(\eta, \epsilon) < \delta$ if $g_1(\eta, \epsilon) > 0$. It can also be seen that $g_1(\eta, \epsilon)$ is a continuous function over $(\eta, \epsilon) \in \mathbb{R}^2_{++}$.

Now let $M > 1$ be arbitrary such that

$$\frac{2\Lambda_m(\delta)}{M} \le \min\left\{1, \frac{M_d}{\delta^2}\right\},$$

and let $(\eta, \epsilon) \in \mathbb{R}^2_{++}$ be arbitrary such that $\eta + \epsilon \ge M$. Consider the following two cases:

1. $\eta\epsilon \ge \Lambda_m(\delta)$. Then

$$\begin{aligned} g_1(\eta, \epsilon) &= \delta\Lambda_m(\delta) - \sqrt{\left(\frac{\delta^2 M_{nc}(\delta)}{4}\sqrt{\eta} + \sqrt{\eta}\epsilon M_d\right)^2} - \beta^{-1}\sigma_\eta \\ &\le \delta\Lambda_m(\delta) - \sqrt{\frac{\delta^4 M_{nc}^2(\delta)}{16}\eta + M_d^2\eta\epsilon^2} - \beta^{-1}\sigma_\eta \\ &\le \delta\Lambda_m(\delta) - \sqrt{\frac{\delta^4 M_{nc}^2(\delta)}{16}\eta + M_d^2\Lambda_m(\delta)\epsilon} - \beta^{-1}\sigma_\eta \\ &\le \delta\Lambda_m(\delta) - \min\left\{\frac{\delta^2 M_{nc}(\delta)}{4}, M_d\sqrt{\Lambda_m(\delta)}\right\}\sqrt{M} - \beta^{-1}\sigma_\eta. \end{aligned}$$

2. $\eta\epsilon < \Lambda_m(\delta)$. In this case, since

$$\epsilon + \frac{\Lambda_m(\delta)}{\epsilon} > \epsilon + \eta \ge M,$$

we must have $\epsilon \ge M/2$ or $\epsilon^{-1}\Lambda_m(\delta) \ge M/2$. In the former case, we have $\eta < \epsilon^{-1}\Lambda_m(\delta) \le 2\Lambda_m(\delta)/M$, and by the choice of M we have $\eta \le 1$ and $\delta\sqrt{\eta} \le M_d$. Thus

$$\begin{aligned} g_1(\eta, \epsilon) &= \delta\eta\epsilon - \frac{\sqrt{\eta}}{4}\delta^2 M_{nc}(\delta) - \sqrt{\eta}\epsilon M_d - \beta^{-1}\sigma_\eta \\ &\le \sqrt{\eta}\epsilon(\delta\sqrt{\eta} - M_d) - \beta^{-1}\sigma_\eta \le -\beta^{-1}\sigma_1. \end{aligned}$$

For the latter case, we have $\eta \ge M - \epsilon \ge M - 2\Lambda_m(\delta)/M \ge M - 1$, and therefore

$$\begin{aligned} g_1(\eta, \epsilon) &= \delta\eta\epsilon - \frac{\sqrt{\eta}}{4}\delta^2 M_{nc}(\delta) - \sqrt{\eta}\epsilon M_d - \beta^{-1}\sigma_\eta \\ &\le \delta\Lambda_m(\delta) - \frac{\sqrt{M-1}}{4}\delta^2 M_{nc}(\delta). \end{aligned}$$

Summarizing these results, we get by the arbitrariness of M that

$$\limsup_{\eta+\epsilon\to+\infty} g_1(\eta, \epsilon) < 0.$$

Therefore the set $g_1^{-1}\left[[0, +\infty)\right]$ is a bounded subset of $\mathbb{R}^2_{++}$.

Now let $(\tilde{\eta}, \tilde{\epsilon})$ be any boundary point of the set $g_1^{-1}\left[[0, +\infty)\right]$ in $\mathbb{R}^2$, and let $(\eta_k, \epsilon_k)_{k\in\mathbb{N}}$ be a sequence in $g_1^{-1}\left[[0, +\infty)\right]$ that converges to $(\tilde{\eta}, \tilde{\epsilon})$. Obviously

$$\liminf_{k\to\infty} g_1(\eta_k, \epsilon_k) \geq 0,$$

and by the continuity of g_1 on $\mathbb{R}^2_{++}$, we further have $\lim_{k\to\infty} g_1(\eta_k, \epsilon_k) = g_1(\tilde{\eta}, \tilde{\epsilon}) = 0$ if $(\tilde{\eta}, \tilde{\epsilon}) \in \mathbb{R}^2_{++}$. Since

$$\limsup_{\substack{(\eta,\epsilon)\to(0,\tilde{\epsilon})\\ (\eta,\epsilon)\in\mathbb{R}^2_{++}}} g_1(\eta, \epsilon) = \limsup_{\eta\to 0^+} \left(-\beta^{-1}\sigma_\eta\right) < 0,$$

we see that $\tilde{\eta} \neq 0$, and then since

$$\lim_{\substack{(\eta,\epsilon)\to(\tilde{\eta},0)\\ (\eta,\epsilon)\in\mathbb{R}^2_{++}}} g_1(\eta, \epsilon) = -\frac{\sqrt{\tilde{\eta}}}{4}\delta^2 M_{nc}(\delta) - \beta^{-1}\sigma_{\tilde{\eta}} < 0,$$

we see that $\tilde{\epsilon} \neq 0$. Therefore the boundary point of $g_1^{-1}[[0, +\infty)]$ in $\mathbb{R}^2$ are all in $\mathbb{R}^2_{++}$. By the continuity of g_1, we can conclude that $g_1^{-1}[[0, +\infty)]$ is a closed subset of $\mathbb{R}^2$. Together with the boundedness shown above, we have shown that $g_1^{-1}[[0, +\infty)]$ is compact.

By the continuity of $g_0(\eta, \epsilon)$ over $(\eta, \epsilon) \in g_1^{-1}[[0, +\infty)]$ and the compactness of $g_1^{-1}[[0, +\infty)]$, the minimum of $g_0(\eta, \epsilon)$ over $(\eta, \epsilon) \in g_1^{-1}[[0, +\infty)]$ is achieved by some $(\eta^*, \epsilon^*) \in g_1^{-1}[[0, +\infty)]$. By assumption, the set

$$\mathscr{A}_{\text{fp}}(\delta, \beta) = g_1^{-1}[(0, +\infty)] \subseteq g_1^{-1}[[0, +\infty)]$$

is nonempty, and for $(\eta, \epsilon) \in g_1^{-1}[(0, +\infty)]$ we have $g_0(\eta, \epsilon) < \delta$, while for $(\eta, \epsilon) \in g_1^{-1}[\{0\}]$ we have $g_0(\eta, \epsilon) = \delta$. Therefore (η^*, ϵ^*) must be in the set $g_1^{-1}[(0, +\infty)]$.

Next we show that $\epsilon^* = \Lambda_m(\delta^*)/\eta^*$. It's not hard to check that the function $\epsilon \mapsto g_1(\eta^*, \epsilon)$ is continuous over $\epsilon \in \mathbb{R}_{++}$, is monotonic when $\epsilon \leq \Lambda_m(\delta)/\eta^*$, and is decreasing when $\epsilon \geq \Lambda_m(\delta)/\eta^*$. Thus the set

$$A_{\eta^*} := \{\epsilon > 0 : g_1(\eta^*, \epsilon) > 0\}$$

is an open interval in $\mathbb{R}_{++}$. We have

$$\frac{d}{d\epsilon} g_0(\eta^*, \epsilon) = -\frac{\eta^*\left(\delta M_d M_{nc}(\delta)/4 + \beta^{-1}\sigma_{\eta^*}\right)}{\left(\eta^*\epsilon - \sqrt{\eta^*}\delta M_{nc}(\delta)/4\right)^2} < 0$$

when $\epsilon \in A_{\eta^*} \cap (0, \Lambda_m(\delta)/\eta^*)$, and

$$\frac{d}{d\epsilon} g_0(\eta^*, \epsilon) = \frac{\sqrt{\eta^*} M_d}{\Lambda_m(\delta) - \sqrt{\eta^*}\delta M_{nc}(\delta)/4} > 0$$

when $\epsilon \in A_{\eta^*} \cap (\Lambda_m(\delta)/\eta^*, +\infty)$. Since the minimum of $g_0(\eta^*, \epsilon)$ over $\epsilon \in A_{\eta^*}$ is achieved, we must have $\epsilon^* = \Lambda_m(\delta)/\eta^*$. This conclusion has a further implication: If we define the function

$$\begin{aligned}\tilde{g}_0(x) &= g_0\left(x^{-2}, x^2\Lambda_m(\delta)\right) = \frac{\beta^{-1}\sigma_{x^{-2}} + x\Lambda_m(\delta)M_d}{\Lambda_m(\delta) - x^{-1}\delta M_{nc}(\delta)/4},\\ \tilde{g}_1(x) &= g_1\left(x^{-2}, x^2\Lambda_m(\delta)\right)\end{aligned}$$

for $x \in (x_l, +\infty)$, where

$$x_l := \frac{\delta M_{nc}(\delta)}{4\Lambda_m(\delta)},$$

then

$$\sqrt{\frac{\epsilon^*}{\Lambda_m(\delta)}} = \frac{1}{\sqrt{\eta^*}} \in \operatorname*{arg\,min}_{x\in\tilde{g}_1^{-1}[(0,+\infty)]} \tilde{g}_0(x).$$

Finally, we prove that $\tilde{g}_0(x)$ has a unique minimizer over $(x_l, +\infty)$. Since $\sigma_{x^{-2}}$ is a convex and nondecreasing function of x, it is absolutely continuous and admits a weak derivative $u(x)$ which is nonnegative and nondecreasing in x. Then we see that $\tilde{g}_0(x)$ is also absolutely continuous, whose weak derivative is given by

$$D\tilde{g}_0(x) = \frac{\beta^{-1}u(x) + \Lambda_m(\delta)M_d - \frac{\beta^{-1}x^{-1}\sigma_{x^{-2}}+\Lambda_m(\delta)M_d}{x(\Lambda_m(\delta)-x^{-1}\delta M_{nc}(\delta)/4)}\frac{\delta M_{nc}(\delta)}{4}}{\Lambda_m(\delta) - x^{-1}\delta M_{nc}(\delta)/4}.$$

We can see that $x^{-1}\sigma_{x^{-2}}$ is equal to

$$\sup_{\substack{t_1,t_2\in[0,T],\\ t_1\neq t_2}} \frac{\left(x^{-2}\left\|x^*(t_2) - x^*(t_1)\right\|^2 + \left(\|\lambda^*(t_2) - \lambda^*(t_1)\|^2 + \|\mu^*(t_2) - \mu^*(t_1)\|^2\right)\right)^{1/2}}{|t_2 - t_1|},$$

showing that $x^{-1}\sigma_{x^{-2}}$ is nonincreasing in x. Then we can easily verify that

$$\frac{\beta^{-1}x^{-1}\sigma_{x^{-2}} + \Lambda_m(\delta)M_d}{x(\Lambda_m(\delta) - x^{-1}\delta M_{nc}(\delta)/4)}$$

is a strictly decreasing function of x, as the numerator is nonincreasing in x and the denominator is positive and strictly increasing in x for $x > \delta M_{nc}(\delta)/(4\Lambda_m(\delta))$. Moreover, we have

$$\begin{aligned}&\lim_{x\to+\infty}\left(\beta^{-1}u(x) + \Lambda_m(\delta)M_d - \frac{\beta^{-1}x^{-1}\sigma_{x^{-2}} + \Lambda_m(\delta)M_d}{x(\Lambda_m(\delta) - x^{-1}\delta M_{nc}(\delta)/4)}\frac{\delta M_{nc}(\delta)}{4}\right)\\ &= \beta^{-1}\lim_{x\to+\infty} u(x) + \Lambda_m(\delta)M_d > 0,\end{aligned}$$

while

$$\lim_{x \to x_l^+} \left(\beta^{-1} u(x) + \Lambda_m(\delta) M_d - \frac{\beta^{-1} x^{-1} \sigma_{x^{-2}} + \Lambda_m(\delta) M_d}{x(\Lambda_m(\delta) - x^{-1} \delta M_{nc}(\delta)/4)} \frac{\delta M_{nc}(\delta)}{4} \right)$$
$$= -\infty.$$

Therefore there exists a unique $x^* \in (x_l, +\infty)$ such that $D\tilde{g}_0(x) < 0$ for $x \in (x_l, x^*)$ and $D\tilde{g}_0(x) > 0$ for $x \in (x^*, +\infty)$. We then see that $\tilde{g}_0(x)$ is a unimodal function over $x \in (x_l, +\infty)$ with the unique minimizer x^* . Therefore

$$\{x^*\} = \arg\min_{x \in (x_l, +\infty)} \tilde{g}_0(x).$$

Since $\tilde{g}_1^{-1}[(0, +\infty)] \subseteq (x_l, +\infty)$, and both $\tilde{g}_1^{-1}[(0, +\infty)]$ and $(x_l, +\infty)$ are open subsets of $\mathbb{R}_{++}$, we have

$$\arg\min_{x \in \tilde{g}_1^{-1}[(0,+\infty)]} \tilde{g}_0(x) \subseteq \arg\min_{x \in (x_l, +\infty)} \tilde{g}_0(x),$$

from which we can conclude that $x^* = 1/\sqrt{\eta^*} = \sqrt{\epsilon^*/\Lambda_m(\delta)}$.

The unimodality of $b_{\delta,\beta}(\epsilon)$ can be obtained by noting that $b_{\delta,\beta}(\epsilon) = \tilde{g}_0\big(\sqrt{\epsilon/\Lambda_m(\delta)}\big)$ and that $\epsilon \mapsto \sqrt{\epsilon/\Lambda_m(\delta)}$ is a strictly increasing function. ■

Chapter 3

SECOND-ORDER ALGORITHMS FOR TIME-VARYING OPTIMIZATION

3.1 Problem Formulation

In this chapter, we consider the following time-varying optimization problem:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & c_\tau(x) + h_\tau(x) \\ \text{s.t.} \quad & f_\tau^{in}(x) \le 0, \\ & f_\tau^{eq}(x) = 0. \end{aligned} \tag{3.1}$$

Here τ labels the discrete time index which ranges in the set $\mathcal{T} = \{1, 2, \dots, K\}$ for some $K \in \mathbb{N}$. The maps $c_\tau : \mathbb{R}^n \to \mathbb{R}$, $f_\tau^{in} : \mathbb{R}^n \to \mathbb{R}^m$ and $f_\tau^{eq} : \mathbb{R}^n \to \mathbb{R}^{m'}$ are twice continuously differentiable, and $h_\tau : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a closed proper convex function with a closed domain for each $\tau \in \mathcal{T}$. We assume that (3.1) always has a solution for every $\tau \in \mathcal{T}$.

By "second-order" algorithms, we mean that for each $\tau \in \mathcal{T}$, not only the current function value and gradient information is used, but also the exact or approximate curvature information is employed when generating approximate solutions to (3.1), which is similar to Newton and quasi-Newton methods in static optimization. In other words, the algorithms discussed in this chapter can be viewed as generalizations of Newton or quasi-Newton methods in the time-varying setting.

Unlike in Chapter 2, we are not going to consider continuous-time limit of second-order algorithms, and we directly consider a discrete-time setting where we already get a sequence of sampled problems. As we shall see, this is partially due to the fact that in practice, second-order methods are usually employed in situations where sampling intervals are large compared to where first-order methods are used. For the analysis of the asymptotic behavior of second-order methods in the continuous-time limit, we refer to [39, 97].

3.2 Approximate Newton Method: A Special Case

We first consider the special case where there are no explicit equality and inequality constraints in (3.1), i.e.,

$$\min_{x\in\mathbb{R}^n} \quad c_\tau(x) + h_\tau(x) \tag{3.2}$$

for each $\tau \in \mathcal{T}$. By Lemma 2.8, its optimality condition is given by

$$-\nabla c_\tau\big(x_\tau^*\big) \in \partial h_\tau\big(x_\tau^*\big), \tag{3.3}$$

where x_τ^* is any local optimal solution to (3.2). Just as in Chapter 2, in the following discussion, we use x_τ^* to denote an arbitrarily chosen local optimal solution in case there are multiple local optimal solutions to (3.2), and we will mainly focus on this particular sequence $\big(x_\tau^*\big)_{\tau\in\mathcal{T}}$.

The approximate Newton method for (3.2) is given by the iteration

$$\hat{x}_\tau = \operatorname*{arg\,min}_{x\in\mathbb{R}^n} \; \nabla c_\tau(\hat{x}_{\tau-1})^T(x - \hat{x}_{\tau-1}) + \frac{1}{2}(x - \hat{x}_{\tau-1})^T B_\tau(x - \hat{x}_{\tau-1}) + h_\tau(x) \tag{3.4}$$

or equivalently in the form of a generalized equation

$$-\nabla c_\tau(\hat{x}_{\tau-1}) - B_\tau(\hat{x}_\tau - \hat{x}_{\tau-1}) \in \partial h_\tau(\hat{x}_\tau) \tag{3.5}$$

for each $\tau \in \mathcal{T}$, where $B_\tau \in \mathbb{R}^{n\times n}$ is a positive definite matrix. The rationale behind (3.4) is that we approximate the smooth part of the objective function c_τ by a convex quadratic function for each τ, and when h_τ is simple or has special structures, we expect that the resulting convex program can be solved much more efficiently than finding the batch solution. We do not put restrictions on the method of producing the matrix B_τ in (3.2) apart from the positive definiteness of B_τ. Therefore (3.2) gives a class of algorithms. For example, we can set B_τ to be a scalar matrix, and in this case (3.2) is a special case of the first-order proximal gradient method proposed in Chapter 2. We can also set $B_\tau = \nabla^2 c_\tau(\hat{x}_{\tau-1})$, or use any quasi-Newton method to calculate B_τ, and in this case we obtain a generalization of the (quasi-)Newton method to the time-varying situation, which can be seen more clearly by noting that (3.4) is equivalent to

$$\hat{x}_\tau = \hat{x}_{\tau-1} - B_\tau^{-1}\nabla c_\tau(\hat{x}_{\tau-1})$$

when $h_\tau = 0$.

Since we use a quadratic approximation of c_τ in deriving (3.4), we can tell from intuition that the quality of this quadratic approximation plays a significant role in the tracking performance. We now proceed to capture this intuition mathematically.

Tracking Performance

We define the *tracking error* by

$$\big\|\hat{x}_\tau - x_\tau^*\big\|_W$$

for each $\tau \in \mathcal{T}$, where we denote $\|x\|_W := x^T W x$ for any $x \in \mathbb{R}^n$ and any positive definite $W \in \mathbb{R}^{n\times n}$. Tracking performance will then be evaluated quantitatively by the tracking error.

We then define

$$\sigma_W := \sup_{\tau\in\mathcal{T}\setminus\{0\}} \left\|x_\tau^* - x_{\tau-1}^*\right\|_W,$$

which upper bounds the distance between consecutive optimal solutions. It characterizes how fast the optimal solution x_τ^* drifts as time proceeds.

Lemma 3.1. *Let $\tau \in \mathcal{T}$ be arbitrary. If $\hat{z}_\tau$ is generated by* (3.4)*, then*

$$\left\|\hat{x}_\tau - x_\tau^*\right\|_{B_\tau} \le \rho_\tau \left\|\hat{x}_{\tau-1} - x_\tau^*\right\|_{B_\tau}, \tag{3.6}$$

where

$$\rho_\tau := \frac{\left\|B_\tau^{-1}\left(\nabla c_\tau(\hat{x}_{\tau-1}) - \nabla c_\tau(x_\tau^*)\right) - \left(\hat{x}_{\tau-1} - x_\tau^*\right)\right\|_{B_\tau}}{\|\hat{x}_{\tau-1} - x_\tau^*\|_{B_\tau}}.$$

Proof. The generalized equation (3.5) implies that

$$h_\tau(x_\tau^*) - h_\tau(\hat{x}_\tau) \ge \left(x_\tau^* - \hat{x}_\tau\right)^T \left(-\nabla c_\tau(\hat{x}_{\tau-1}) - B_\tau\left(\hat{x}_\tau - \hat{x}_{\tau-1}\right)\right).$$

On the other hand, by the optimality condition (3.3), we have

$$h_\tau(\hat{x}_\tau) - h_\tau(x_\tau^*) \ge -\left(\hat{x}_\tau - x_\tau^*\right)^T \nabla c_\tau(x_\tau^*).$$

We then have

$$\left(\hat{x}_\tau - x_\tau^*\right)^T \left(\nabla c_\tau(\hat{x}_{\tau-1}) - \nabla c_\tau(x_t^*) + B_\tau\left(\hat{x}_\tau - \hat{x}_{\tau-1}\right)\right) \le 0,$$

which leads to

$$\begin{aligned}
&\left\|\hat{x}_\tau - x_\tau^*\right\|_{B_\tau}^2 = \left(\hat{x}_\tau - x_\tau^*\right)^T B_\tau \left(\hat{x}_\tau - x_\tau^*\right) \\
\le\ & \left(\hat{x}_\tau - x_\tau^*\right)^T B_\tau \left(B_\tau^{-1}\left(\nabla c_\tau(x_t^*) - \nabla c_\tau(\hat{x}_{\tau-1})\right) - \left(x_\tau^* - \hat{x}_{\tau-1}\right)\right) \\
\le\ & \left\|\hat{x}_\tau - x_\tau^*\right\|_{B_\tau} \cdot \left\|B_\tau^{-1}\left(\nabla c_\tau(x_t^*) - \nabla c_\tau(\hat{x}_{\tau-1})\right) - \left(x_\tau^* - \hat{x}_{\tau-1}\right)\right\|_{B_\tau} \\
=\ & \left\|\hat{x}_\tau - x_\tau^*\right\|_{B_\tau} \cdot \rho_\tau \left\|\hat{x}_{\tau-1} - x_\tau^*\right\|_{B_\tau}^2.
\end{aligned}$$

The inequality (3.6) now follows directly. □

Lemma 3.1 directly leads to the following theorem on the tracking error bound for (3.4).

Theorem 3.1. *Let $W \in \mathbb{R}^{n\times n}$ be a positive definite matrix, and*

$$\lambda_M := \sup_{\tau\in\mathcal{T}} \inf\{\lambda \in \mathbb{R} : \lambda W \geq B_\tau\},$$
$$\lambda_m := \inf_{\tau\in\mathcal{T}} \sup\{\lambda \in \mathbb{R} : \lambda W \leq B_\tau\}.$$

If

$$\rho := \sup_{\tau\in\mathcal{T}} \frac{\left\|B_\tau^{-1}\left(\nabla c_\tau(\hat{x}_{\tau-1}) - \nabla c_\tau(x_\tau^*)\right) - (\hat{x}_{\tau-1} - x_\tau^*)\right\|_{B_\tau}}{\|\hat{x}_{\tau-1} - x_\tau^*\|_{B_\tau}} < \sqrt{\frac{\lambda_m}{\lambda_M}}, \tag{3.7}$$

then

$$\left\|\hat{x}_\tau - x_\tau^*\right\|_W \leq \frac{\rho\sigma_W}{\sqrt{\lambda_m/\lambda_M} - \rho} + \left(\rho\sqrt{\frac{\lambda_M}{\lambda_m}}\right)^\tau \left(\left\|\hat{x}_0 - x_1^*\right\|_W - \frac{\sigma_W\sqrt{\lambda_m/\lambda_M}}{\sqrt{\lambda_m/\lambda_M} - \rho}\right) \tag{3.8}$$

for any $\tau \in \mathcal{T}$.

Proof. The definition of ρ implies that $\rho_\tau \leq \rho$ for all $\tau \in \mathcal{T}$. Therefore by Lemma 3.1, for $\tau = 1$, we have

$$\left\|\hat{x}_1 - x_1^*\right\|_W \leq \sqrt{\lambda_m^{-1}}\left\|\hat{x}_1 - x_1^*\right\|_{B_1} \leq \rho\sqrt{\lambda_m^{-1}}\left\|\hat{x}_0 - x_1^*\right\|_{B_1} \leq \rho\sqrt{\frac{\lambda_M}{\lambda_m}}\left\|\hat{x}_1 - x_0^*\right\|_W,$$

while if (3.8) holds for some τ, then

$$\begin{aligned}
&\left\|\hat{x}_{\tau+1} - x_{\tau+1}^*\right\|_W \\
&\leq \sqrt{\lambda_m^{-1}}\left\|\hat{x}_{\tau+1} - x_{\tau+1}^*\right\|_{B_\tau} \leq \rho\sqrt{\lambda_m^{-1}}\left\|\hat{x}_\tau - x_{\tau+1}^*\right\|_{B_\tau} \\
&\leq \rho\sqrt{\frac{\lambda_M}{\lambda_m}}\left(\left\|\hat{x}_\tau - x_\tau^*\right\|_W + \left\|x_\tau^* - x_{\tau+1}^*\right\|_W\right) \\
&\leq \rho\sqrt{\frac{\lambda_M}{\lambda_m}}\left(\frac{\rho\sigma_W}{\sqrt{\lambda_m/\lambda_M} - \rho} + \left(\rho\sqrt{\frac{\lambda_M}{\lambda_m}}\right)^\tau\left(\left\|\hat{x}_0 - x_1^*\right\|_W - \frac{\sigma_W\sqrt{\lambda_m/\lambda_M}}{\sqrt{\lambda_m/\lambda_M} - \rho}\right) + \sigma_W\right) \\
&= \frac{\rho\sigma_W}{\sqrt{\lambda_m/\lambda_M} - \rho} + \left(\rho\sqrt{\frac{\lambda_M}{\lambda_m}}\right)^{\tau+1}\left(\left\|\hat{x}_0 - x_1^*\right\|_W - \frac{\sigma_W\sqrt{\lambda_m/\lambda_M}}{\sqrt{\lambda_m/\lambda_M} - \rho}\right).
\end{aligned}$$

The bound (3.8) then follows from mathematical induction. □

The condition (3.7) and the tracking error bound (3.8) suggest that ρ should be sufficiently small so that good tracking performance can be achieved. However, in practice, it is difficult to evaluate the quantity ρ unless we carry out the whole iterations (3.4), which makes Theorem 3.1 not very useful for estimating the tracking error *a priori*. Rather, Theorem 3.1 justifies our intuition that the quality of the quadratic approximation in (3.4), which is now quantitatively assessed by ρ, plays a central role in the tracking performance. Furthermore, in order to make ρ as small as possible, we need

$$\nabla c_\tau\left(\hat{x}_{\tau-1}\right) - \nabla c_\tau\left(x_\tau^*\right) \approx B_\tau\left(\hat{x}_{\tau-1} - x_\tau^*\right).$$

for each τ; in other words, the matrices B_τ in (3.4) should approximate the (averaged) Hessian of c_τ along the direction $\hat{x}_{\tau-1} - x_\tau^*$. This implication partially substantiates our motivation of introducing second-order methods: although (3.4) is in general harder to compute than the proximal gradient method, the tracking error can be potentially reduced by employing a more sophisticated B_τ.

A Centralized Algorithm Based on L-BFGS-B

As Theorem 3.1 points out, the approach of producing the matrices B_τ has a major effect on the tracking performance of the approximate Newton method (3.4). However, the Hessian of c_τ is in general a dense $n \times n$ matrix, which makes the scalability of (3.4) questionable as n increases. Another issue is that (3.4) involves optimization of a potentially nonsmooth function, and in the special case where h_τ is the indicator function of some convex set $\mathcal{X}_\tau$, we obtain a constrained optimization problem which cannot be solved by directly applying algorithms for unconstrained optimization.

In this subsection, we provide a specific algorithm based on L-BFGS-B to deal with these issues, when the nonsmooth convex function h_τ is the indicator function of a rectangular set in $\mathbb{R}^n$ with a nonempty interior:

$$h_\tau = I_{\mathcal{X}_\tau}, \qquad \mathcal{X}_\tau = \left\{x \in \mathbb{R}^n : \underline{x}_\tau \le x \le \overline{x}_\tau\right\}.$$

The L-BFGS-B algorithm [26, 69] is a quasi-Newton method that solves nonlinear programs with box constraints. It employs two key techniques: the limited-memory BFGS method to produce the approximate Hessian, and the gradient projection method[1] to handle the box constraints. By applying these two key techniques to the time-varying optimization setting, we propose the following procedure for each $\tau \in \mathcal{T}$:

1. Produce B_τ by the limited-memory BFGS method.

[1] Not to be confused with the projected gradient method.

2. Use the gradient projection method to get an approximate solution to the quadratic program

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & q_\tau(x) := g_\tau^T(x-\hat{x}_{\tau-1}) + \frac{1}{2}(x-\hat{x}_{\tau-1})^T B_\tau (x-\hat{x}_{\tau-1}) \\ \text{s.t.} \quad & \underline{x}_\tau \leq x \leq \overline{x}_\tau, \end{aligned} \tag{3.9}$$

where $g_\tau = \nabla c_\tau(\hat{x}_{\tau-1})$.

Note that the gradient projection method which will be presented shortly will only produce an approximate solution to the quadratic program (3.9) in general. Nevertheless, simulation shows that such an approximate solution usually suffices.

We now give a succinct introduction to these two key techniques under the time-varying optimization setting.

Limited-memory BFGS In limited-memory BFGS, the approximate Hessian B_τ takes the form

$$B_\tau = \vartheta_\tau I - K_\tau M_\tau K_\tau^T, \tag{3.10}$$

where $K_\tau \in \mathbb{R}^{n\times 2d}$ and $M_\tau \in \mathbb{R}^{2d\times 2d}$ for some $d \in \mathbb{N}$. In practice d is typically between 3 and 20 and is usually much smaller than n. It can be seen that B_τ is a small rank correction of a scalar matrix.

To be specific, we denote

$$s_\tau = \hat{x}_\tau - \hat{x}_{\tau-1}, \qquad y_\tau = \nabla c_\tau(\hat{x}_\tau) - \nabla c_\tau(\hat{x}_{\tau-1}),$$

and

$$Y_\tau = \begin{bmatrix} y_{\tau-d} & \cdots & y_{\tau-1} \end{bmatrix}, \qquad S_\tau = \begin{bmatrix} s_{\tau-d} & \cdots & s_{\tau-1} \end{bmatrix},$$

The matrices K_τ and M_τ are then given by

$$K_\tau = \begin{bmatrix} Y_\tau & \vartheta_\tau S_\tau \end{bmatrix}, \qquad M_\tau = \begin{bmatrix} -D_\tau & L_\tau^T \\ L_\tau & \vartheta_\tau S_\tau^T S_\tau \end{bmatrix}^{-1},$$

where $D_\tau \in \mathbb{R}^{d\times d}$ and $L_\tau \in \mathbb{R}^{d\times d}$ are given by

$$\begin{aligned} D_\tau &= \operatorname{diag}\left(s_{\tau-d}^T y_{\tau-d}, \ldots, s_{\tau-1}^T y_{\tau-1}\right), \\ (L_\tau)_{ij} &= \begin{cases} s_{\tau-d-1+i}^T y_{\tau-d-1+j}, & \text{if } i > j, \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

The scalar ϑ_τ is given by

$$\vartheta_\tau = \frac{y_{\tau-1}^T y_{\tau-1}}{y_{\tau-1}^T s_{\tau-1}}.$$

For $1 < \tau \leq d$, the matrices Y_τ, S_τ, D_τ and L_τ will only be constructed from $s_1, \ldots, s_{\tau-1}$ and $y_1, \ldots, y_{\tau-1}$, and consequently $Y_\tau \in \mathbb{R}^{n\times(\tau-1)}$, $S_\tau \in \mathbb{R}^{n\times(\tau-1)}$, $K_\tau \in \mathbb{R}^{n\times 2(\tau-1)}$ and $M_\tau \in \mathbb{R}^{2(\tau-1)\times 2(\tau-1)}$. For $\tau = 1$, we let $B_\tau = \vartheta_0 I$ where $\vartheta_0 > 0$ is an initial estimate. For the rationale behind the limited-memory BFGS method and minor modifications to guarantee the positive definiteness of B_τ, we refer to [26, 27, 72].

The gradient projection method The gradient projection method estimates the free variables of the box constraints in (3.9) and then performs a subspace minimization to provide an approximate solution to (3.9). The estimation of the free variables is via the generalized Cauchy point. Let

$$\tilde{x}_\tau(t) = \mathcal{P}_{\mathcal{X}_\tau}\left[w_{\tau-1} - t\nabla q_\tau(w_{\tau-1})\right], \qquad w_{\tau-1} := \mathcal{P}_{\mathcal{X}_\tau}(\hat{x}_{\tau-1}),$$

and let t_τ^c be the smallest minimizer of the piecewise quadratic function $q_\tau(\tilde{x}_\tau(t))$ over $t > 0$. The generalized Cauchy point is then given by $x_\tau^c := \tilde{x}_\tau\left(t_\tau^c\right)$. Let

$$\mathcal{I} = \left\{i \in \{1, \ldots, n\} : x_{\tau,i}^c = \overline{x}_{\tau,i} \text{ or } x_{\tau,i}^c = \underline{x}_{\tau,i}\right\}.$$

We then solve

$$\begin{aligned} u_\tau = \operatorname*{arg\,min}_{u\in\mathbb{R}^n} \quad & q_\tau(x_\tau^c + u) \\ \text{s.t.} \quad & u_{\tau,\mathcal{I}} = 0. \end{aligned} \tag{3.11}$$

It can be immediately recognized that (3.11) is essentially an unconstrained quadratic program. After obtaining u_τ, we compute[2]

$$\tilde{u}_\tau = \mathcal{P}_{\mathcal{X}_\tau}\left[x_\tau^c + u_\tau\right] - w_{\tau-1}, \tag{3.12}$$

and generate $\hat{x}_\tau$ by

$$\hat{x}_\tau = w_{\tau-1} + \alpha_\tau \tilde{u}_\tau,$$

where α_τ is equal to 1 or is determined by some backtracking strategy.

Because of the compact representation of the limited-memory BFGS method (3.10), the search for t_τ^c and the minimization (3.11) can be computed in a very efficient manner that has time complexity $O(d^2 n + d^3 + n \log n)$. More details of the gradient projection method can be found in [72, Section 16.7] and also in [26, 69].

[2] In the case where the $\tilde{u}_\tau$ generated by (3.12) does not satisfy $\nabla q_\tau(w_{\tau-1})^T \tilde{u}_\tau < 0$, we set $\tilde{u}_\tau = x_\tau^c + u_\tau - w_{\tau-1}$, which is a descent direction of q_τ as shown in [26]. A backtracking is then needed to ensure that $\hat{x}_\tau$ is feasible.

3.3 Approximate Newton Method: The General Case

Now we consider the general case given by

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c_\tau(x) + h_\tau(x) \\ \text{s.t.} \quad & f_\tau^{in}(x) \leq 0, \\ & f_\tau^{eq}(x) = 0, \end{aligned} \tag{3.1}$$

where there are explicit equality and inequality constraints. By Proposition 2.1, under certain constraint qualification conditions, the KKT conditions are given by

$$-\nabla c_\tau\left(x_\tau^*\right) - \begin{bmatrix} J_{f_\tau^{in}}\left(x_\tau^*\right) \\ J_{f_\tau^{eq}}\left(x_\tau^*\right) \end{bmatrix}^T \begin{bmatrix} \lambda_\tau^* \\ \mu_\tau^* \end{bmatrix} \in \partial h_\tau\left(x_\tau^*\right), \tag{3.13a}$$

$$f_\tau^{in}\left(x_\tau^*\right) \in N_{\mathbb{R}_+^m}\left(\lambda_\tau^*\right), \tag{3.13b}$$

$$f_\tau^{eq}\left(x_\tau^*\right) = 0, \tag{3.13c}$$

where $\left(x_\tau^*, \lambda_\tau^*, \mu_\tau^*\right) \in \text{dom}(h_\tau) \times \mathbb{R}_+^m \times \mathbb{R}^{m'}$ is a KKT point of (3.1). Just as in Chapter 2, when there are multiple solutions to the KKT conditions, we arbitrarily select one of them, denote it by $z_\tau^* = \left(x_\tau^*, \lambda_\tau^*, \mu_\tau^*\right)$, and mainly focus on this trajectory in most part of our study.

We shall introduce two approaches that handle the inequality and equality constraints differently.

The Penalty Approach

The penalty approach introduces the penalty functions

$$\phi_\tau(x) := \sum_{i=1}^{m} \left[f_{\tau,i}^{in}(x)\right]_+^\kappa + \sum_{j=1}^{m'} \left|f_{\tau,j}^{eq}(x)\right|^\kappa .$$

Here $[\cdot]_+ := \max\{0, \cdot\}$ denotes the positive part of a scalar, and $\kappa \in (2, +\infty)$ is a fixed constant. Then instead of (3.1), we consider the following penalized version

$$\min_{x \in \mathbb{R}^n} \quad F_\tau^\epsilon(x) + h_\tau(x), \tag{3.14}$$

where

$$F_\tau^\epsilon(x) := c_\tau(x) + \frac{1}{\kappa\epsilon}\phi_\tau(x),$$

and $\epsilon > 0$ is the penalty parameter. The penalized function $F_\tau^\epsilon(x)$ is twice continuously differentiable as we assume $\kappa > 2$. Then, we can just apply the approximate

Newton method (3.4), with c_τ being replaced by F_τ^ϵ. Specifically,

$$\begin{aligned}\hat{x}_\tau = \arg\min_{x\in\mathbb{R}^n} \quad & \left(\nabla c_\tau\left(\hat{x}_{\tau-1}\right) + \begin{bmatrix} J_{f_\tau^{in}}\left(\hat{x}_{\tau-1}\right) \\ J_{f_\tau^{eq}}\left(\hat{x}_{\tau-1}\right)\end{bmatrix}^T \begin{bmatrix}\hat{\lambda}_{\tau-1} \\ \hat{\mu}_{\tau-1}\end{bmatrix}\right)^T (x - \hat{x}_{\tau-1}) \\ & + \frac{1}{2}(x - \hat{x}_{\tau-1})^T B_\tau (x - \hat{x}_{\tau-1}) + h_\tau(x),\end{aligned} \tag{3.15}$$

where

$$\begin{aligned}\hat{\lambda}_{\tau-1,i} &= \epsilon^{-1}\left[f_{\tau,i}^{in}(\hat{x}_{\tau-1})\right]_+^{\kappa-1}, & i = 1,\ldots,m, \\ \hat{\mu}_{\tau-1,j} &= \epsilon^{-1}\operatorname{sgn}\left(f_{\tau,i}^{eq}(\hat{x}_{\tau-1})\right)\left|f_{\tau,i}^{eq}(\hat{x}_{\tau-1})\right|^{\kappa-1}, & j = 1,\ldots,m',\end{aligned}$$

and B_τ is a positive definite matrix that tries to approximate the Hessian of F_τ^ϵ.

On the other hand, we should note that (3.15) tracks a solution to the penalized problem (3.14), which in general is different from the solutions to the original problem (3.1). As a result, if we calculate the tracking error with respect to x_τ^*, there will be some additional term in the tracking error bound due to this difference, which can be regarded as a side effect of the penalty approach. This is similar to the situation in Chapter 2 where the regularization on the dual variables introduces additional error. In fact, the similarity is not a coincidence, as we shall show that the penalized problem is closely related to a min-max problem with regularization on the dual variables. Let us consider

$$\min_{x\in\mathbb{R}^n}\ \max_{\lambda\in\mathbb{R}_+^m,\mu\in\mathbb{R}^{m'}} \mathcal{L}_\tau^\epsilon(x,\lambda,\mu) \tag{3.16}$$

with

$$\begin{aligned}\mathcal{L}_\tau^\epsilon(x,\lambda,\mu) := {} & c_\tau(x) + h_\tau(x) + \lambda^T f_\tau^{in}(x) + \mu^T f_\tau^{eq}(x) \\ & - \frac{\epsilon^{\beta-1}}{\beta}\left(\sum_{i=1}^m |\lambda_i|^\beta + \sum_{j=1}^{m'} |\mu_j|^\beta\right),\end{aligned} \tag{3.17}$$

where $\beta \in (1,2)$ is determined by $\kappa^{-1} + \beta^{-1} = 1$. The term

$$-\frac{\epsilon^{\beta-1}}{\beta}\left(\sum_{i=1}^m |\lambda_i|^\beta + \sum_{j=1}^{m'} |\mu_j|^\beta\right)$$

represents the regularization added on the dual variables. It's not hard to see that $\mathcal{L}_\tau^\epsilon(x,\lambda,\mu)$ is additively separable in the dual variables, and the maximization of $\mathcal{L}_\tau^\epsilon(x,\lambda,\mu)$ over $\lambda \in \mathbb{R}_+^m, \mu \in \mathbb{R}^{m'}$ can be explicitly solved. Let $\tilde{\lambda}_\tau(x)$ and $\tilde{\mu}_\tau(x)$ denote the optimal dual variables that maximize $\mathcal{L}_\tau^\epsilon(x,\lambda,\mu)$ for a given fixed x.

Then

$$\tilde{\lambda}_{\tau,i}(x) = \epsilon^{-1} \left[f^{in}_{\tau,i}(x)\right]_+^{\frac{1}{\beta-1}},$$
$$\tilde{\mu}_{\tau,j}(x) = \epsilon^{-1} \operatorname{sgn}\left(f^{eq}_{\tau,j}(x)\right) \left|f^{eq}_{\tau,j}(x)\right|^{\frac{1}{\beta-1}},$$

and

$$\begin{aligned}
&\max_{\lambda\in\mathbb{R}^m_+,\mu\in\mathbb{R}^{m'}} \mathcal{L}^\epsilon_\tau(x,\lambda,\mu) = \mathcal{L}^\epsilon_\tau\big(x, \tilde{\lambda}_\tau(x), \tilde{\mu}_\tau(x)\big) \\
&= c_\tau(x) + \epsilon^{-1}\sum_{i=1}^{m} \left[f^{in}_{\tau,i}(x)\right]_+^{\frac{1}{\beta-1}} f^{in}_{\tau,i}(x) + \epsilon^{-1}\sum_{j=1}^{m'} \operatorname{sgn}\left(f^{eq}_{\tau,j}(x)\right)\left|f^{eq}_{\tau,j}(x)\right|^{\frac{1}{\beta-1}} f^{eq}_{\tau,j}(x) \\
&\quad - \frac{\epsilon^{\beta-1}}{\beta}\cdot\epsilon^{-\beta}\left(\sum_{i=1}^{m}\left[f^{in}_{\tau,i}(x)\right]_+^{\frac{\beta}{\beta-1}} + \sum_{j=1}^{m'}\left|f^{eq}_{\tau,j}(x)\right|^{\frac{\beta}{\beta-1}}\right) + h_\tau(x) \\
&= c_\tau(x) + \frac{1}{\kappa\epsilon}\left(\sum_{i=1}^{m}\left[f^{in}_{\tau,i}(x)\right]_+^{\kappa} + \sum_{j=1}^{m'}\left|f^{eq}_{\tau,j}(x)\right|^{\kappa}\right) + h_\tau(x) = F^\epsilon_\tau(x) + h_\tau(x),
\end{aligned}$$

where we used the fact that $\kappa = \beta/(\beta-1)$. Thus we see that the penalized problem (3.14) is essentially equivalent to the min-max problem (3.16) in the sense that $\tilde{x}_\tau$ is a local optimal solution to (3.14) if and only if $\big(\tilde{x}_\tau, \tilde{\lambda}(\tilde{x}_\tau), \tilde{\mu}(\tilde{x}_\tau)\big)$ is a local solution to the min-max problem (3.16) with regularization on the dual variables.

Intuitively, as the penalty parameter ϵ tends to zero, the solution to the penalized problem (3.14) should converge in certain sense to an optimal solution to the original problem (3.1). We shall not provide a comprehensive study of how the difference of the two solutions change as ϵ decreases. The following two theorems bound this difference for two special cases, whose proofs are given in Appendix 3.A.

Theorem 3.2. *Consider the following convex program*

$$\begin{aligned}
\min_{x\in\mathbb{R}^n} \quad & c(x) + h(x) \\
s.t. \quad & f^{in}(x) \le 0, \\
& a_j^T x = b_j, \quad j = 1,\dots,m',
\end{aligned} \tag{3.18}$$

where $c : \mathbb{R}^n \to \mathbb{R}$ *and* $f^{in} : \mathbb{R}^n \to \mathbb{R}$ *are twice continuously differentiable,* $h : \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$, $a_j \in \mathbb{R}^n$ *for each* $j = 1,\dots,m'$, $b \in \mathbb{R}^{p'}$. *The functions* $c(x)$, $h(x)$ *and each component of* $f^{in}(x)$ *are all convex over* $x \in \mathbb{R}^n$. *Let* X^* *denote the set of optimizers to* (3.18) *which we assume to be nonempty.*

Assume that the constraint qualification condition (2.4) *holds for* (3.18) *at any optimal solution. Furthermore, we assume that there exists some* $\nu > 0$ *such that*

$$(x - \mathcal{P}_{X^*}(x))^T \left(\nabla c(x) - \nabla c\big(\mathcal{P}_{X^*}(x)\big)\right) \ge \nu \left\|x - \mathcal{P}_{X^*}(x)\right\|^2, \quad \forall x \in \operatorname{dom}(h). \tag{3.19}$$

Let $\tilde{x}_\epsilon$ denote a solution to the penalized problem

$$\min_{x\in\mathbb{R}^n} \quad c(x) + h(x) + \frac{1}{\kappa\epsilon}\left(\sum_{i=1}^{m}\left[f_i^{in}(x)\right]_+^{\kappa} + \sum_{j=1}^{m'}\left|a_j^T x - b_j\right|^{\kappa}\right), \tag{3.20}$$

where $\epsilon > 0$ and $\kappa > 1$. Then

$$\begin{aligned} &\nu\,\|\tilde{x}_\epsilon - \mathcal{P}_{X^*}(\tilde{x}_\epsilon)\|^2 + \frac{(\kappa-1)}{\kappa\epsilon}\left(\sum_{i=1}^{m}\left[f_i^{in}(\tilde{x}_\epsilon)\right]_+^{\kappa} + \sum_{j=1}^{m'}\left|a_j^T \tilde{x}_\epsilon - b_j\right|^{\kappa}\right) \\ \le\; &\frac{(\kappa-1)\,\epsilon^{\frac{1}{\kappa-1}}}{\kappa}\left(\sum_{i=1}^{m}\left|\lambda_i^*\right|^{\frac{\kappa}{\kappa-1}} + \sum_{j=1}^{m'}\left|\mu_j^*\right|^{\frac{\kappa}{\kappa-1}}\right), \end{aligned} \tag{3.21}$$

where (λ^*, μ^*) is any optimal dual variable associated with $\mathcal{P}_{X^*}(\tilde{x}_\epsilon)$.

Theorem 3.3. *Let $c : \mathbb{R}^n \to \mathbb{R}$, $f^{in} : \mathbb{R}^n \to \mathbb{R}^m$, $f^{eq} : \mathbb{R}^n \to \mathbb{R}^{m'}$, $g^{in} : \mathbb{R}^n \to \mathbb{R}^p$ and $g^{eq} : \mathbb{R}^n \to \mathbb{R}^{p'}$ be twice continuously differentiable. Suppose x^* is a local optimal solution to the nonlinear program*

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & c(x) \\ \text{s.t.} \quad & f^{in}(x) \le 0, \qquad f^{eq}(x) = 0, \\ & g^{in}(x) \le 0, \qquad g^{eq}(x) = 0. \end{aligned} \tag{3.22}$$

Suppose the linear independence constraint qualification holds for (3.22) at x^*, and let the optimal dual variables be $\lambda^* \in \mathbb{R}_+^m$, $\mu^* \in \mathbb{R}^{m'}$, $\zeta^* \in \mathbb{R}_+^p$ and $\upsilon^* \in \mathbb{R}^{p'}$. We further assume that strict complementary slackness holds for λ^* and ζ^*, that all entries of μ^* are nonzero, and that $\nabla^2\mathcal{L}(x^*, \lambda^*, \mu^*, \zeta^*, \upsilon^*)$ is positive definite, where $\mathcal{L}$ denotes the Lagrangian

$$\mathcal{L}(x, \lambda, \mu, \zeta, \upsilon) = c(x) + \lambda^T f^{in}(x) + \mu^T f^{eq}(x) + \zeta^T g^{in}(x) + \upsilon^T g^{eq}(x).$$

Then there exist $\bar{\epsilon} > 0$, $M > 0$ and a unique continuous function $\tilde{x} : [0, \bar{\epsilon}) \to \mathbb{R}^n$ such that $\tilde{x}(0) = x^*$ and $\tilde{x}(\epsilon)$ is a local solution to the penalized problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & c(x) + \frac{1}{\kappa\epsilon}\left(\sum_{i=1}^{m}\left[f_i^{in}(x)\right]_+^{\kappa} + \sum_{j=1}^{m'}\left|f_j^{eq}(x)\right|^{\kappa}\right) \\ \text{s.t.} \quad & g^{in}(x) \le 0, \qquad g^{eq}(x) = 0 \end{aligned} \tag{3.23}$$

for any $\epsilon \in (0, \bar{\epsilon})$, where $\kappa > 2$ is a fixed constant. Moreover,

$$\|\tilde{x}(\epsilon) - x^*\|_{\ell_\kappa} \le \epsilon^{\frac{1}{\kappa-1}} M \left(\sum_{i=1}^{m}\left|\lambda_i^*\right|^{\frac{\kappa}{\kappa-1}} + \sum_{i=1}^{m}\left|\mu_i^*\right|^{\frac{\kappa}{\kappa-1}}\right)^{\frac{1}{\kappa}}. \tag{3.24}$$

Remark 3.1. Theorem 3.2 considers convex programs, and the bound (3.21) is "global" in the sense that it holds for any local optimal solution to the penalized problem (3.20). Theorem 3.3 considers nonlinear programs for which linear independence constraint qualification and strict complementary slackness hold at some optimal point, and the result is "local" in the sense that the bound (3.24) only applies to penalized problems with $\epsilon < \bar{\epsilon}$ where $\bar{\epsilon}$ is guaranteed to exist but not explicitly specified. ■

There is a trade-off when choosing the penalty parameter ϵ: A smaller ϵ in general leads to more accurate tracking as the penalized problem (3.14) approximates the original problem (3.1) better, but it also makes the penalized objective function F_τ^ϵ more ill-conditioned, which could also increase the difficulty in producing the approximate Hessian B_τ. The parameter κ should not be too large as well, and in practice we usually choose $2 < \kappa < 3$.

The Saddle Point Approach

In the penalty approach, we require $\kappa > 2$ so that the function F_τ^ϵ is twice continuously differentiable. If we set $\kappa = 2$, the penalty approach may still work in practice, but it becomes unclear whether the quasi-Newton method is able to handle the discontinuity of the Hessian. In the following, we propose an alternative approach that is similar to the penalty approach for $\kappa = 2$ but avoids the discontinuity issue by employing a saddle point formulation.

It has been shown that for $\kappa = 2$, the penalized problem (3.14) is equivalent to the min-max problem

$$\min_{x\in\mathbb{R}^n} \max_{\lambda\in\mathbb{R}^m_+,\mu\in\mathbb{R}^{m'}} \quad c_\tau(x) + h_\tau(x) + \lambda^T f_\tau^{in}(x) + \mu^T f_\tau^{eq}(x) - \frac{\epsilon}{2}\left(\|\lambda\|^2 + \|\mu\|^2\right). \tag{3.25}$$

In the saddle point approach, for each $\tau \in \mathcal{T}$, we let $\hat{z}_\tau = \left(\hat{x}_\tau, \hat{\lambda}_\tau, \hat{\mu}_\tau\right)$ to be the solution to the following saddle point problem:

$$\begin{aligned}\min_{x\in\mathbb{R}^n} \max_{\lambda\in\mathbb{R}^m_+,\mu\in\mathbb{R}^{m'}} \quad & \nabla c_\tau(\hat{x}_{\tau-1})^T(x-\hat{x}_{\tau-1}) + \frac{1}{2}(x-\hat{x}_{\tau-1})^T B_\tau (x-\hat{x}_{\tau-1}) \\ & + \begin{bmatrix}\lambda \\ \mu\end{bmatrix}^T \begin{bmatrix} f_\tau^{in}(\hat{x}_{\tau-1}) + J_{f_\tau^{in}}(\hat{x}_{\tau-1})(x-\hat{x}_{\tau-1}) \\ f_\tau^{eq}(\hat{x}_{\tau-1}) + J_{f_\tau^{eq}}(\hat{x}_{\tau-1})(x-\hat{x}_{\tau-1})\end{bmatrix} \\ & - \frac{\epsilon}{2}\left(\left\|\lambda - \lambda_\tau^{\text{reg}}\right\|^2 + \left\|\mu - \mu_\tau^{\text{reg}}\right\|^2\right) + h_\tau(x),\end{aligned} \tag{3.26}$$

where

$$\lambda_\tau^{\text{reg}} = (1-\chi)\hat{\lambda}_{\tau-1} + \chi\lambda_{\text{prior}},$$
$$\mu_\tau^{\text{reg}} = (1-\chi)\hat{\mu}_{\tau-1} + \chi\mu_{\text{prior}}.$$

In (3.26), B_τ is still a positive definite matrix that is updated at each time slot, $\epsilon > 0$ is the regularization parameter, λ_{prior} and μ_{prior} are employed as prior estimation of the optimal dual variables, and $\chi \in (0,1]$ is a constant.

By comparing (3.26) with (3.25), we can see that, apart from a more sophisticated regularization that drives the dual variables towards $\left(\lambda_\tau^{\text{reg}}, \mu_\tau^{\text{reg}}\right)$, the saddle point approach still employs the "quadratic approximation trick" as in (3.4). It is also closely related to sequential quadratic programming for static optimziation problems. But since the min-max formulation is kept, what we obtain is a saddle point problem whose objective function is convex in x and strongly concave in (λ, μ). The KKT conditions on $\hat{z}_\tau$ are given by

$$-\nabla c_\tau(\hat{x}_{\tau-1}) - B_\tau(\hat{x}_\tau - \hat{x}_{\tau-1}) - \begin{bmatrix} J_{f_\tau^{in}}(\hat{x}_{\tau-1}) \\ J_{f_\tau^{eq}}(\hat{x}_{\tau-1}) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_\tau \\ \hat{\mu}_\tau \end{bmatrix} \in \partial h_\tau(\hat{x}_\tau), \tag{3.27a}$$
$$f_\tau^{in}(\hat{x}_{\tau-1}) + J_{f_\tau^{in}}(\hat{x}_{\tau-1})(\hat{x}_\tau - \hat{x}_{\tau-1}) - \epsilon\left(\hat{\lambda}_\tau - \lambda_\tau^{\text{reg}}\right) \in N_{\mathbb{R}_+^m}\left(\hat{\lambda}_\tau\right), \tag{3.27b}$$
$$f_\tau^{eq}(\hat{x}_{\tau-1}) + J_{f_\tau^{eq}}(\hat{x}_{\tau-1})(\hat{x}_\tau - \hat{x}_{\tau-1}) - \epsilon\left(\hat{\mu}_\tau - \mu_\tau^{\text{reg}}\right) = 0. \tag{3.27c}$$

These conditions can be written in the form of a generalized linear equation

$$\begin{bmatrix} B_\tau & J_{f_\tau^{in}}(\hat{x}_{\tau-1})^T & J_{f_\tau^{eq}}(\hat{x}_{\tau-1})^T \\ -J_{f_\tau^{in}}(\hat{x}_{\tau-1}) & \epsilon I & 0 \\ -J_{f_\tau^{eq}}(\hat{x}_{\tau-1}) & 0 & \epsilon I \end{bmatrix} \begin{bmatrix} \tilde{x}_\tau \\ \hat{\lambda}_\tau \\ \hat{\mu}_\tau \end{bmatrix} \in -\partial\tilde{h}_\tau\left(\begin{bmatrix} \tilde{x}_\tau \\ \hat{\lambda}_\tau \\ \hat{\mu}_\tau \end{bmatrix}\right) + \begin{bmatrix} -\nabla c_\tau(\hat{x}_{\tau-1}) \\ f_\tau^{in}(\hat{x}_{\tau-1}) + \epsilon\lambda_\tau^{\text{reg}} \\ f_\tau^{eq}(\hat{x}_{\tau-1}) + \epsilon\mu_\tau^{\text{reg}} \end{bmatrix},$$

where we denote $\tilde{x}_\tau = \hat{x}_\tau - \hat{x}_{\tau-1}$ and

$$\tilde{h}_\tau(x, \lambda, \mu) = h_\tau(x + \hat{x}_{\tau-1}) + I_{\mathbb{R}_+^m}(\lambda).$$

When h_τ is simple or has special structures, we expect that this generalized linear equation can be solved much more efficiently than finding a batch solution.

Now let us study the tracking performance of the saddle point approach. For any positive definite matrix $W \in \mathbb{R}^{n\times n}$ and positive scalar ϵ, we define the norm

$$\|z\|_{W,\epsilon} := \left(\|x\|_W^2 + \epsilon\|\lambda\|^2 + \epsilon\|\mu\|^2\right)^{1/2}$$

for any $z = (x, \lambda, \mu) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{m'}$. We use

$$\sigma_{W,\epsilon} := \sup_{t\in\mathcal{T}\setminus\{1\}} \left\|z_\tau^* - z_{\tau-1}^*\right\|_{W,\epsilon}$$

to denote the maximum distance between each consecutive optimal primal-dual pair with respect to the norm $\|\cdot\|_{W,\epsilon}$. We denote

$$\mathcal{L}_\tau^s := c_\tau(x) + \lambda^T f_\tau^{in}(x) + \mu^T f_\tau^{eq}(x),$$

which can be thought of as the Lagrangian of the "smooth" part of the problem (3.1).

We also define the following quantities:

$$\begin{aligned} M_d &:= \sup_{t\in\mathcal{T}} \left\| \begin{bmatrix} \lambda_\tau^* - \lambda_{\text{prior}} \\ \mu_\tau^* - \mu_{\text{prior}} \end{bmatrix} \right\|, \\ M_f(\delta, W) &:= \sup_{\tau\in\mathcal{T}} \sup_{u:\|u\|_W\leq\delta} \frac{1}{\|u\|_W^2} \left\| \begin{bmatrix} f_\tau^{in}\left(x_\tau^* + u\right) - J_{f_\tau^{in}}\left(x_\tau^* + u\right)u - f_\tau^{in}\left(x_\tau^*\right) \\ f_\tau^{eq}\left(x_\tau^* + u\right) - J_{f_\tau^{eq}}\left(x_\tau^* + u\right)u - f_\tau^{eq}\left(x_\tau^*\right) \end{bmatrix} \right\|. \end{aligned}$$

The quantity $M_f(\delta, W)$ measures the nonlinearity of the constraint functions f_τ^{in} and f_τ^{eq} within the neighborhood around x_τ^* of radius $\delta > 0$, which is similar to the quantity $M_{nc}(\delta)$ in the regularized proximal primal-dual gradient method in Chapter 2; W is a positive definite matrix that provides some flexibility in choosing the norm of the primal variable.

Theorem 3.4. *Let $W \in \mathbb{R}^{n\times n}$ be a positive definite matrix, and*

$$\lambda_M := \sup_{\tau\in\mathcal{T}} \inf\{\lambda \in \mathbb{R} : \lambda W \succeq B_\tau\},$$
$$\lambda_m := \inf_{\tau\in\mathcal{T}} \sup\{\lambda \in \mathbb{R} : \lambda W \preceq B_\tau\},$$

and define

$$\rho^{(\mathrm{P})} := \sup_{t\in\mathcal{T}} \frac{\left\|\left(\hat{x}_{\tau-1} - x_\tau^*\right) - B_\tau^{-1}\left(\nabla_x \mathcal{L}_\tau^s\left(\hat{x}_{\tau-1}, \lambda_\tau^*, \mu_\tau^*\right) - \nabla_x \mathcal{L}_\tau^s\left(x_\tau^*, \lambda_\tau^*, \mu_\tau^*\right)\right)\right\|_{B_\tau}}{\|\hat{x}_{\tau-1} - x_\tau^*\|_{B_\tau}},$$

$$\rho := \left(\max\left\{ \rho^{(\mathrm{P})} \sqrt{\frac{\lambda_M}{\lambda_m}}, 1 - \chi \right\}^2 + (1-\chi)\frac{\delta M_f(\delta, \lambda_m W)}{2\sqrt{\epsilon}} + \frac{\delta^2 M_f^2(\delta, \lambda_m W)}{4\epsilon} \right)^{1/2}.$$

If

$$\sigma_{\lambda_m W,\epsilon} \leq (1-\rho)\delta - \sqrt{2\epsilon}\chi M_d, \tag{3.28}$$

and the initial point $\hat{z}_0$ satisfies $\left\|\hat{z}_0 - z_1^*\right\|_{\lambda_m W,\epsilon} \leq \delta$, then

$$\begin{aligned} \left\|\hat{z}_\tau - z_\tau^*\right\|_{\lambda_m W,\epsilon} \leq{}& \frac{\rho\sigma_{\lambda_m W,\epsilon} + \sqrt{2\epsilon}\chi M_d}{1-\rho} \\ &+ \rho^\tau \left(\left\|\hat{z}_0 - z_1^*\right\|_{\lambda_m W,\epsilon} - \frac{\sigma_{\lambda_m W,\epsilon} + \sqrt{2\epsilon}\chi M_d}{1-\rho} \right) \end{aligned} \tag{3.29}$$

for all $\tau \in \mathcal{T}$.

Proof. Let $\tau \in \mathcal{T}$ be arbitrary, and suppose that $\left\|\hat{x}_{\tau-1} - x^*_\tau\right\|_{\lambda_m W} \leq \delta$. By (3.27a), we have

$$\begin{aligned} &h_\tau\left(x^*_\tau\right) - h_\tau\left(\hat{x}_\tau\right) \\ &\geq \ -\left(x^*_\tau - \hat{x}_\tau\right)^T \left(\nabla c_\tau\left(\hat{x}_{\tau-1}\right) + B_\tau(\hat{x}_\tau - \hat{x}_{\tau-1}) + \begin{bmatrix} J_{f^{in}_\tau}(\hat{x}_{\tau-1}) \\ J_{f^{eq}_\tau}(\hat{x}_{\tau-1}) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_\tau \\ \hat{\mu}_\tau \end{bmatrix}\right), \end{aligned}$$

while (3.13b) implies that

$$\begin{aligned} &h_\tau\left(\hat{x}_\tau\right) - h_\tau\left(x^*_\tau\right) \\ &\geq \ -\left(\hat{x}_\tau - x^*_\tau\right)^T \left(\nabla c_\tau\left(x^*_\tau\right) + \begin{bmatrix} J_{f^{in}_\tau}\left(x^*_{\tau-1}\right) \\ J_{f^{eq}_\tau}\left(x^*_{\tau-1}\right) \end{bmatrix}^T \begin{bmatrix} \lambda^*_\tau \\ \mu^*_\tau \end{bmatrix}\right). \end{aligned}$$

Therefore

$$\begin{aligned} 0 \geq \left(\hat{x}_\tau - x^*_\tau\right)^T \Bigg(&\nabla c_\tau\left(\hat{x}_{\tau-1}\right) + B_\tau(\hat{x}_\tau - \hat{x}_{\tau-1}) - \nabla c_\tau\left(x^*_\tau\right) \\ &+ \begin{bmatrix} J_{f^{in}_\tau}\left(\hat{x}_{\tau-1}\right) \\ J_{f^{eq}_\tau}\left(\hat{x}_{\tau-1}\right) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_\tau \\ \hat{\mu}_\tau \end{bmatrix} - \begin{bmatrix} J_{f^{in}_\tau}\left(x^*_{\tau-1}\right) \\ J_{f^{eq}_\tau}\left(x^*_{\tau-1}\right) \end{bmatrix}^T \begin{bmatrix} \lambda^*_\tau \\ \mu^*_\tau \end{bmatrix}\Bigg), \end{aligned}$$

and we get

$$\begin{aligned} &\left\|\hat{x}_\tau - x^*_\tau\right\|^2_{B_\tau} \\ &\leq \ -\left(\hat{x}_\tau - x^*_\tau\right)^T \Bigg(\nabla c_\tau\left(\hat{x}_{\tau-1}\right) - \nabla c_\tau\left(x^*_\tau\right) - B_\tau\left(\hat{x}_{\tau-1} - x^*_\tau\right) \\ &\qquad + \begin{bmatrix} J_{f^{in}_\tau}\left(\hat{x}_{\tau-1}\right) \\ J_{f^{eq}_\tau}\left(\hat{x}_{\tau-1}\right) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_\tau \\ \hat{\mu}_\tau \end{bmatrix} - \begin{bmatrix} J_{f^{in}_\tau}\left(x^*_{\tau-1}\right) \\ J_{f^{eq}_\tau}\left(x^*_{\tau-1}\right) \end{bmatrix}^T \begin{bmatrix} \lambda^*_\tau \\ \mu^*_\tau \end{bmatrix}\Bigg) \\ &= \ -\left(\hat{x}_\tau - x^*_\tau\right)^T \left(\nabla_x \mathcal{L}^s_\tau\left(\hat{x}_{\tau-1}, \lambda^*_\tau, \mu^*_\tau\right) - \nabla_x \mathcal{L}^s_\tau\left(x^*_\tau, \lambda^*_\tau, \mu^*_\tau\right) - B_\tau\left(\hat{x}_{\tau-1} - x^*_\tau\right)\right) \\ &\quad -\left(\hat{x}_\tau - x^*_\tau\right)^T \begin{bmatrix} J_{f^{in}_\tau}\left(\hat{x}_{\tau-1}\right) \\ J_{f^{eq}_\tau}\left(\hat{x}_{\tau-1}\right) \end{bmatrix}^T \begin{bmatrix} \hat{\lambda}_\tau - \lambda^*_\tau \\ \hat{\mu}_\tau - \mu^*_\tau \end{bmatrix}. \end{aligned} \tag{3.30}$$

Now, by (3.27b), we can get

$$\left(\lambda^*_\tau - \hat{\lambda}_\tau\right)^T \left(f^{in}_\tau(\hat{x}_{\tau-1}) + J_{f^{in}_\tau}(\hat{x}_{\tau-1})(\hat{x}_\tau - \hat{x}_{\tau-1}) - \epsilon\left(\hat{\lambda}_\tau - \lambda^{\text{reg}}_\tau\right)\right) \leq 0,$$

and by (3.13c),

$$\left(\hat{\lambda}_\tau - \lambda^*_\tau\right)^T f^{in}_\tau\left(x^*_\tau\right) \leq 0.$$

Thus

$$\left(\hat{\lambda}_\tau - \lambda^*_\tau\right)^T \left(f^{in}_\tau(\hat{x}_{\tau-1}) + J_{f^{in}_\tau}(\hat{x}_{\tau-1})(\hat{x}_\tau - \hat{x}_{\tau-1}) - \epsilon\left(\hat{\lambda}_\tau - \lambda^{\text{reg}}_\tau\right) - f^{in}_\tau\left(x^*_\tau\right)\right) \geq 0,$$

or

$$
\begin{aligned}
\epsilon \left\|\hat{\lambda}_\tau - \lambda_\tau^*\right\|^2 \leq\ & \left(\hat{\lambda}_\tau - \lambda_\tau^*\right)^T J_{f_\tau^{in}}\left(\hat{x}_{\tau-1}\right)\left(\hat{x}_\tau - x_\tau^*\right) \\
& + \left(\hat{\lambda}_\tau - \lambda_\tau^*\right)^T \left(f_\tau^{in}\left(\hat{x}_{\tau-1}\right) + J_{f_\tau^{in}}\left(\hat{x}_{\tau-1}\right)\left(x_\tau^* - \hat{x}_{\tau-1}\right) - f_\tau^{in}\left(x_\tau^*\right)\right) \\
& + \epsilon \left(\hat{\lambda}_\tau - \lambda_\tau^*\right)^T \left(\lambda_\tau^{\text{reg}} - \lambda_\tau^*\right).
\end{aligned} \tag{3.31}
$$

Similarly, by (3.27b) and (3.13c), it can be shown that

$$
\begin{aligned}
\epsilon \left\|\hat{\mu}_\tau - \mu_\tau^*\right\|^2 =\ & \left(\hat{\mu}_\tau - \mu_\tau^*\right)^T J_{f_\tau^{eq}}\left(\hat{x}_{\tau-1}\right)\left(\hat{x}_\tau - x_\tau^*\right) \\
& + \left(\hat{\mu}_\tau - \mu_\tau^*\right)^T \left(f_\tau^{eq}\left(\hat{x}_{\tau-1}\right) + J_{f_\tau^{eq}}\left(\hat{x}_{\tau-1}\right)\left(x_\tau^* - \hat{x}_{\tau-1}\right) - f_\tau^{eq}\left(x_\tau^*\right)\right) \\
& + \epsilon \left(\hat{\mu}_\tau - \mu_\tau^*\right)^T \left(\mu_\tau^{\text{reg}} - \mu_\tau^*\right).
\end{aligned} \tag{3.32}
$$

By summing (3.30), (3.31), and (3.32), we can see that

$$
\begin{aligned}
& \left\|\hat{x}_\tau - x_\tau^*\right\|_{B_\tau}^2 + \epsilon \left\|\hat{\lambda}_\tau - \lambda_\tau^*\right\|^2 + \epsilon \left\|\hat{\mu}_\tau - \mu_\tau^*\right\|^2 \\
\leq\ & -\left(\hat{x}_\tau - x_\tau^*\right)^T \left(\nabla_x \mathcal{L}_\tau^s\left(\hat{x}_{\tau-1}, \lambda_\tau^*, \mu_\tau^*\right) - \nabla_x \mathcal{L}_\tau^s\left(x_\tau^*, \lambda_\tau^*, \mu_\tau^*\right) - B_\tau\left(\hat{x}_{\tau-1} - x_\tau^*\right)\right) \\
& + \begin{bmatrix} \hat{\lambda}_\tau - \lambda_\tau^* \\ \hat{\mu}_\tau - \mu_\tau^* \end{bmatrix}^T \begin{bmatrix} f_\tau^{in}\left(\hat{x}_{\tau-1}\right) + J_{f_\tau^{in}}\left(\hat{x}_{\tau-1}\right)\left(x_\tau^* - \hat{x}_{\tau-1}\right) - f_\tau^{in}\left(x_\tau^*\right) \\ f_\tau^{eq}\left(\hat{x}_{\tau-1}\right) + J_{f_\tau^{eq}}\left(\hat{x}_{\tau-1}\right)\left(x_\tau^* - \hat{x}_{\tau-1}\right) - f_\tau^{eq}\left(x_\tau^*\right) \end{bmatrix} \\
& + \epsilon \begin{bmatrix} \hat{\lambda}_\tau - \lambda_\tau^* \\ \hat{\mu}_\tau - \mu_\tau^* \end{bmatrix}^T \begin{bmatrix} \lambda_\tau^{\text{reg}} - \lambda_\tau^* \\ \mu_\tau^{\text{reg}} - \mu_\tau^* \end{bmatrix} \\
=\ & \begin{bmatrix} \hat{x}_\tau - x_\tau^* \\ \hat{\lambda}_\tau - \lambda_\tau^* \\ \hat{\mu}_\tau - \mu_\tau^* \end{bmatrix}^T \begin{bmatrix} B_\tau & & \\ & \epsilon I_m & \\ & & \epsilon I_{m'} \end{bmatrix} \begin{bmatrix} d_{\tau,x} \\ d_{\tau,\lambda} \\ d_{\tau,\mu} \end{bmatrix},
\end{aligned}
$$

where we denote

$$
\begin{bmatrix} d_{\tau,x} \\ d_{\tau,\lambda} \\ d_{\tau,\mu}, \end{bmatrix} = \begin{bmatrix} \left(\hat{x}_{\tau-1} - x_\tau^*\right) - B_\tau^{-1}\left(\nabla_x \mathcal{L}_\tau^s\left(\hat{x}_{\tau-1}, \lambda_\tau^*, \mu_\tau^*\right) - \nabla_x \mathcal{L}_\tau^s\left(x_\tau^*, \lambda_\tau^*, \mu_\tau^*\right)\right) \\ \epsilon^{-1}\left(f_\tau^{in}\left(\hat{x}_{\tau-1}\right) + J_{f_\tau^{in}}\left(\hat{x}_{\tau-1}\right)\left(x_\tau^* - \hat{x}_{\tau-1}\right) - f_\tau^{in}\left(x_\tau^*\right)\right) + \lambda_\tau^{\text{reg}} - \lambda_\tau^* \\ \epsilon^{-1}\left(f_\tau^{eq}\left(\hat{x}_{\tau-1}\right) + J_{f_\tau^{eq}}\left(\hat{x}_{\tau-1}\right)\left(x_\tau^* - \hat{x}_{\tau-1}\right) - f_\tau^{eq}\left(x_\tau^*\right)\right) + \mu_\tau^{\text{reg}} - \mu_\tau^* \end{bmatrix}.
$$

By the Cauchy–Schwarz inequality, we see that

$$
\begin{aligned}
& \left\|\hat{x}_\tau - x_\tau^*\right\|_{B_\tau}^2 + \epsilon \left\|\hat{\lambda}_\tau - \lambda_\tau^*\right\|^2 + \epsilon \left\|\hat{\mu}_\tau - \mu_\tau^*\right\|^2 \\
\leq\ & \left(\left\|\hat{x}_\tau - x_\tau^*\right\|_{B_\tau}^2 + \epsilon \left\|\hat{\lambda}_\tau - \lambda_\tau^*\right\|^2 + \epsilon \left\|\hat{\mu}_\tau - \mu_\tau^*\right\|^2\right)^{1/2} \\
& \times \left(\left\|d_{\tau,x}\right\|_{B_\tau}^2 + \epsilon \left\|d_{\tau,\lambda}\right\|^2 + \epsilon \left\|d_{\tau,\mu}\right\|^2\right)^{1/2},
\end{aligned}
$$

and thus

$$\left\|\hat{x}_\tau - x_\tau^*\right\|_{B_\tau}^2 + \epsilon \left\|\hat{\lambda}_\tau - \lambda_\tau^*\right\|^2 + \epsilon \left\|\hat{\mu}_\tau - \mu_\tau^*\right\|^2$$
$$\leq \|d_{\tau,x}\|_{B_\tau}^2 + \epsilon \left\|d_{\tau,\lambda}\right\|^2 + \epsilon \left\|d_{\tau,\mu}\right\|^2 .$$

By the definition of $\rho^{(\mathrm{P})}$, we have

$$\|d_{\tau,x}\|_{B_\tau} \leq \rho^{(\mathrm{P})} \left\|\hat{x}_{\tau-1} - x_\tau^*\right\|_{B_\tau} \leq \rho^{(\mathrm{P})} \sqrt{\frac{\lambda_M}{\lambda_m}} \left\|\hat{x}_{\tau-1} - x_\tau^*\right\|_{\lambda_m W},$$

and by the definition of $M_f(\delta, W)$ and M_d, we have

$$\begin{aligned}
&\epsilon \left(\left\|d_{\tau,\lambda}\right\|^2 + \left\|d_{\tau,\mu}\right\|^2\right) \\
&\leq \epsilon \left(\epsilon^{-1} \frac{M_f(\delta, \lambda_m W)}{2} \left\|\hat{x}_{\tau-1} - x_\tau^*\right\|_{\lambda_m W}^2 + (1-\chi) \left\|\begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix}\right\| + \chi \left\|\begin{bmatrix} \lambda_{\mathrm{prior}} - \lambda_\tau^* \\ \mu_{\mathrm{prior}} - \mu_\tau^* \end{bmatrix}\right\|\right)^2 \\
&\leq \frac{\delta^2 M_f^2(\delta, \lambda_m W)}{4\epsilon} \left\|\hat{x}_{\tau-1} - x_\tau^*\right\|_W^2 + \epsilon(1-\chi)^2 \left\|\begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix}\right\|^2 + \epsilon \chi^2 M_d^2 \\
&\quad + (1-\chi)\delta M_f(\delta, \lambda_m W) \left\|\hat{x}_{\tau-1} - x_\tau^*\right\|_{\lambda_m W} \left\|\begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix}\right\| \\
&\quad + \chi M_d \delta M_f(\delta, \lambda_m W) \left\|\hat{x}_{\tau-1} - x_\tau^*\right\|_{\lambda_m W} + 2\epsilon M_d \chi (1-\chi) \left\|\begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix}\right\|.
\end{aligned}$$

We also note that

$$\left\|\hat{x}_{\tau-1} - x_\tau^*\right\|_{\lambda_m W} \left\|\begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix}\right\| \leq \frac{1}{2\sqrt{\epsilon}} \left(\left\|\hat{x}_{\tau-1} - x_\tau^*\right\|_{\lambda_m W}^2 + \epsilon \left\|\begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix}\right\|^2\right).$$

Therefore

$$\begin{aligned}
&\left\|\hat{x}_\tau - x_\tau^*\right\|_{B_\tau}^2 + \epsilon \left\|\hat{\lambda}_\tau - \lambda_\tau^*\right\|^2 + \epsilon \left\|\hat{\mu}_\tau - \mu_\tau^*\right\|^2 \\
&\leq \left(\left(\rho^{(\mathrm{P})} \sqrt{\frac{\lambda_M}{\lambda_m}}\right)^2 + (1-\chi)\frac{\delta M_f(\delta, \lambda_m W)}{2\sqrt{\epsilon}} + \frac{\delta^2 M_f^2(\delta, \lambda_m W)}{4\epsilon}\right) \left\|\hat{x}_{\tau-1} - x_\tau^*\right\|_{\lambda_m W}^2 \\
&\quad + \left((1-\chi)^2 + (1-\chi)\frac{\delta M_f(\delta, \lambda_m W)}{2\sqrt{\epsilon}}\right) \epsilon \left\|\begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix}\right\|^2 \\
&\quad + 2\chi\sqrt{\epsilon} M_d \left(\frac{\delta M_f(\delta, \lambda_m W)}{2\sqrt{\epsilon}} \left\|\hat{x}_{\tau-1} - x_\tau^*\right\|_{\lambda_m W} + (1-\chi)\sqrt{\epsilon} \left\|\begin{bmatrix} \hat{\lambda}_{\tau-1} - \lambda_\tau^* \\ \hat{\mu}_{\tau-1} - \mu_\tau^* \end{bmatrix}\right\|\right) \\
&\quad + \epsilon \chi^2 M_d^2 \\
&\leq \left(\rho \left(\left\|\hat{x}_{\tau-1} - x_\tau^*\right\|_{\lambda_m W}^2 + \epsilon \left\|\hat{\lambda}_{\tau-1} - \lambda_\tau^*\right\|^2 + \epsilon \left\|\hat{\mu}_{\tau-1} - \mu_\tau^*\right\|^2\right)^{1/2} + \sqrt{2\epsilon}\chi M_d\right)^2,
\end{aligned}$$

where we used

$$\frac{\delta M_f(\delta,\lambda_m W)}{2\sqrt{\epsilon}}\left\|\hat{x}_{\tau-1}-x_\tau^*\right\|_{\lambda_m W}+(1-\chi)\sqrt{\epsilon}\left\|\begin{bmatrix}\hat{\lambda}_{\tau-1}-\lambda_\tau^*\\ \hat{\mu}_{\tau-1}-\mu_\tau^*\end{bmatrix}\right\|$$
$$\leq\sqrt{2}\left(\frac{\delta^2 M_f^2(\delta,\lambda_m W)}{4\epsilon}\left\|\hat{x}_{\tau-1}-x_\tau^*\right\|_{\lambda_m W}^2+(1-\chi)^2\epsilon\left\|\begin{bmatrix}\hat{\lambda}_{\tau-1}-\lambda_\tau^*\\ \hat{\mu}_{\tau-1}-\mu_\tau^*\end{bmatrix}\right\|^2\right)^{1/2}$$
$$\leq\sqrt{2}\rho\left(\left\|\hat{x}_{\tau-1}-x_\tau^*\right\|_{\lambda_m W}^2+\epsilon\left\|\hat{\lambda}_{\tau-1}-\lambda_\tau^*\right\|^2+\epsilon\left\|\hat{\mu}_{\tau-1}-\mu_\tau^*\right\|^2\right)^{1/2},$$

and so

$$\left\|\hat{z}_\tau-z_\tau^*\right\|_{\lambda_m W,\epsilon}\leq\rho\left\|\hat{z}_{\tau-1}-z_\tau^*\right\|_{\lambda_m W,\epsilon}+\sqrt{2\epsilon}\chi M_d. \tag{3.33}$$

Now, for $\tau=1$, since we have assumed that $\left\|\hat{z}_0-z_1^*\right\|_{\lambda_m W,\epsilon}\leq\delta$, we see that $\left\|\hat{x}_0-x_1^*\right\|_{\lambda_m W}\leq\delta$ and we can apply (3.33) to get

$$\left\|\hat{z}_1-z_1^*\right\|_{\lambda_m W,\epsilon}\leq\rho\left\|\hat{z}_0-z_1^*\right\|_{\lambda_m W,\epsilon}+\sqrt{2\epsilon}\chi M_d,$$

showing that (3.29) holds for $\tau=1$. Then suppose that (3.29) holds for some $\tau\in\mathcal{T}$. We then have

$$\left\|\hat{x}_\tau-x_{\tau+1}^*\right\|_{\lambda_m W}\leq\left\|\hat{z}_\tau-z_{\tau+1}^*\right\|_{\lambda_m W,\epsilon}\leq\left\|\hat{z}_\tau-z_\tau^*\right\|_{\lambda_m W,\epsilon}+\sigma_{\lambda_m W,\epsilon}$$
$$\leq\frac{\rho\sigma_{\lambda_m W,\epsilon}+\sqrt{2\epsilon}\chi M_d}{1-\rho}+\rho^\tau\left(\left\|\hat{z}_0-z_1^*\right\|_{\lambda_m W,\epsilon}-\frac{\sigma_{\lambda_m W,\epsilon}+\sqrt{2\epsilon}\chi M_d}{1-\rho}\right)+\sigma_{\lambda_m W,\epsilon}$$
$$\leq\frac{\sigma_{\lambda_m W,\epsilon}+\sqrt{2\epsilon}\chi M_d}{1-\rho}+\rho^\tau\left(\delta-\frac{\sigma_{\lambda_m W,\epsilon}+\sqrt{2\epsilon}\chi M_d}{1-\rho}\right).$$

By (3.28),

$$\frac{\sigma_{\lambda_m W,\epsilon}+\sqrt{2\epsilon}\chi M_d}{1-\rho}\leq\delta.$$

Thus we get $\left\|\hat{x}_\tau-x_{\tau+1}^*\right\|_{\lambda_m W}\leq\delta$, and we apply (3.33) to get

$$\left\|\hat{z}_{\tau+1}-z_{\tau+1}^*\right\|_{\lambda_m W,\epsilon}\leq\rho\left(\left\|\hat{z}_\tau-z_\tau^*\right\|_{\lambda_m W,\epsilon}+\sigma_{\lambda_m W,\epsilon}\right)+\sqrt{2\epsilon}\chi M_d$$
$$\leq\rho\cdot\frac{\rho\sigma_{\lambda_m W,\epsilon}+\sqrt{2\epsilon}\chi M_d}{1-\rho}+\rho\sigma_{\lambda_m W,\epsilon}$$
$$+\rho^{\tau+1}\left(\left\|\hat{z}_0-z_1^*\right\|_{\lambda_m W,\epsilon}-\frac{\sigma_{\lambda_m W,\epsilon}+\sqrt{2\epsilon}\chi M_d}{1-\rho}\right)+\sqrt{2\epsilon}\chi M_d$$
$$=\frac{\rho\sigma_{\lambda_m W,\epsilon}+\sqrt{2\epsilon}\chi M_d}{1-\rho}+\rho^{\tau+1}\left(\left\|\hat{z}_0-z_1^*\right\|_{\lambda_m W,\epsilon}-\frac{\sigma_{\lambda_m W,\epsilon}+\sqrt{2\epsilon}\chi M_d}{1-\rho}\right).$$

By induction we conclude that (3.29) holds for all $\tau\in\mathcal{T}$. □

The conditions and the conclusion of Theorem 3.4 are very similar to those of Theorem 2.1 for the regularized proximal primal-dual method. The expression of the quantities $\rho^{(\mathrm{P})}$ and ρ, on the other hand, are much simpler. Some direct implications from Theorem 3.4 are as follows.

1. The quantity $\rho^{(\mathrm{P})}$ plays a crucial role in the tracking error bound (3.29), and its definition is similar to (3.7) in Theorem 3.1. It's obvious that $\rho^{(\mathrm{P})}$ should be as small as possible in order to achieve good tracking performance, which implies that the matrices B_τ should satisfy

$$\nabla_x \mathcal{L}^s_\tau\big(\hat{x}_{\tau-1}, \lambda^*_\tau, \mu^*_\tau\big) - \nabla_x \mathcal{L}^s_\tau\big(x^*_\tau, \lambda^*_\tau, \mu^*_\tau\big) \approx B_\tau(\hat{x}_{\tau-1} - x^*_\tau),$$

or in other words, B_τ should approximate

$$\nabla^2 c_\tau(x) + \sum_{i=1}^m \lambda^*_{\tau,i} \nabla^2 f^{in}_\tau(x) + \sum_{j=1}^m \mu^*_{\tau,i} \nabla^2 f^{eq}_\tau(x)$$

along the direction $\hat{x}_{\tau-1} - x^*_\tau$. The appearance of λ^*_τ and μ^*_τ in the above expression adds difficulty in designing quasi-Newton algorithms for finding such B_τ. In practice, we may replace them by $\hat{\lambda}_{\tau-1}$ and $\hat{\mu}_{\tau-1}$, and employ the quasi-Newton methods designed for sequential quadratic programming (see, for example, [72, Section 18.3]) to generate B_τ.

2. The expression of ρ implies that the nonlinearity of the constraint functions will deteriorate the tracking performance. Further study seems to suggest that this negative effect cannot be eliminated even if the components of f^{in} are convex, which is possibly due to the fact that all the constraints are linearized in formulating (3.26) regardless of their convexity. However, if any component of f^{in}_τ (say, $f^{in}_{\tau,1}$) is convex and the constraint $f^{in}_{\tau,1} \leq 0$ is easy to handle computationally for each $\tau \in \mathcal{T}$, we may consider solving the following saddle-point problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \max_{\tilde{\lambda}\in\mathbb{R}^{m-1}_+, \mu\in\mathbb{R}^{m'}} \quad & \nabla c_\tau(\hat{x}_{\tau-1})^T(x-\hat{x}_{\tau-1}) + \frac{1}{2}(x-\hat{x}_{\tau-1})^T B_\tau (x-\hat{x}_{\tau-1}) \\ & + \begin{bmatrix} \tilde{\lambda} \\ \mu \end{bmatrix}^T \begin{bmatrix} \tilde{f}^{in}_\tau(\hat{x}_{\tau-1}) + J_{\tilde{f}^{in}_\tau}(\hat{x}_{\tau-1})(x-\hat{x}_{\tau-1}) \\ f^{eq}_\tau(\hat{x}_{\tau-1}) + J_{f^{eq}_\tau}(\hat{x}_{\tau-1})(x-\hat{x}_{\tau-1}) \end{bmatrix} \\ & - \frac{\epsilon}{2}\left(\big\|\tilde{\lambda} - \tilde{\lambda}^{\mathrm{reg}}_\tau\big\|^2 + \big\|\mu - \mu^{\mathrm{reg}}_\tau\big\|^2\right) + \tilde{h}_\tau(x) \end{aligned}$$

to generate $\hat{z}_\tau$, where $\tilde{\lambda}$, $\tilde{\lambda}^{\mathrm{reg}}_\tau$ and $\tilde{f}^{in}_\tau$ are formed by excluding the first components of λ, $\lambda^{\mathrm{reg}}_\tau$ and f^{in}_τ respectively, and

$$\tilde{h}_\tau(x) = h_\tau(x) + \mathcal{I}_{\left\{x : f^{in}_{\tau,1}(x) \leq 0\right\}}(x).$$

One can still apply Theorem 3.4 to this procedure and obtain the tracking error bound. In the extreme case where all the components of f_τ^{in} are convex and f_τ^{eq} is affine for each $\tau \in \mathcal{T}$, we see that we may consider solving

$$\begin{aligned}\min_{x\in\mathbb{R}^n} \quad & \nabla c_\tau(\hat{x}_{\tau-1})^T(x-\hat{x}_{\tau-1}) + \frac{1}{2}(x-\hat{x}_{\tau-1})^T B_\tau (x-\hat{x}_{\tau-1}) + h_\tau(x) \\ \text{s.t.} \quad & f_\tau^{in}(x) \le 0, \qquad f_\tau^{eq}(x) = 0\end{aligned}$$

to get $\hat{x}_\tau$. In this case Theorem 3.4 will still apply with $\rho = \rho^{(\mathrm{P})}\sqrt{\lambda_M/\lambda_m}$ [3]. It is interesting to observe that we don't need regularization on the dual variables associated with convex constraints.

3.4 Comparison of First-Order and Second-Order Methods

In this section we present a toy example to give a preliminary comparison between first-order and second-order methods for time-varying optimization.

We consider the following time-varying optimization problem:

$$\begin{aligned}\min_{x\in\mathbb{R}^{10}} \quad & \frac{1}{2}x^T A_U(t) x \\ \text{s.t.} \quad & x \in \mathcal{X}(t),\end{aligned} \tag{3.34}$$

where

$$A_U(t) = \frac{1+\cos t}{2}\operatorname{diag}(1,2,\ldots,10) + \frac{1-\cos t}{2} U \operatorname{diag}(1,2,\ldots,10) U^T,$$

for some orthoginal matrix $U \in \mathbb{R}^{10\times 10}$, and

$$\mathcal{X}(t) = \left\{ x \in \mathbb{R}^{10} : \left| x_i - 2\cos\left(t + \frac{(i-1)\pi}{10}\right)\right| \le \frac{1}{2}\right\}.$$

Suppose the time period is $[0, 2\pi]$, and the sampling interval is Δ. Let the sequence of sampled optimal points be denoted by $\left(x_\tau^{*,\Delta,U}\right)_\tau$. We apply both the first-order running projected gradient algorithm and the second-order algorithm based on L-BFGS-B on the time-varying problem (3.34), and get two sequences of points which we denote by $\left(\hat{x}_\tau^{\Delta,U,1}\right)_\tau$ and $\left(\hat{x}_\tau^{\Delta,U,2}\right)_\tau$ respectively. The initial point for both algorithms will be the optimal solution to (3.34) for $t = 0$. For the limited-memory BFGS representation of B_τ, we choose $d = 3$. We are interested in the averaged tracking error

$$\frac{1}{\lfloor 2\pi/\Delta \rfloor}\sum_\tau \left\| \hat{x}_\tau^{\Delta,U,k} - x_\tau^{*,\Delta,U} \right\|$$

[3] In fact Theorem 3.1 will also apply if we let $h_\tau(x)$ include the convex constraints using indicator functions.

	$\Delta = \pi/125$		$\Delta = \pi/250$	
	first-order	second-order	first-order	second-order
1	0.0207	0.0115	0.0105	0.0034
2	0.0254	0.0100	0.0127	0.0041
3	0.0334	0.0110	0.0154	0.0048
4	0.0270	0.0129	0.0137	0.0042
5	0.0266	0.0140	0.0130	0.0045
6	0.0218	0.0100	0.0096	0.0037
7	0.0216	0.0086	0.0102	0.0027
8	0.0192	0.0096	0.0093	0.0035
9	0.0229	0.0104	0.0112	0.0036
10	0.0213	0.0099	0.0098	0.0040

	$\Delta = \pi/500$		$\Delta = \pi/1000$	
	first-order	second-order	first-order	second-order
1	0.0052	0.0011	0.0026	3.8×10^{-4}
2	0.0063	0.0016	0.0032	6.7×10^{-4}
3	0.0073	0.0018	0.0036	7.4×10^{-4}
4	0.0069	0.0013	0.0035	4.7×10^{-4}
5	0.0064	0.0014	0.0032	4.5×10^{-4}
6	0.0045	0.0012	0.0022	4.6×10^{-4}
7	0.0049	0.0020	0.0024	3.2×10^{-4}
8	0.0046	0.0011	0.0023	4.1×10^{-4}
9	0.0056	0.0012	0.0027	4.2×10^{-4}
10	0.0046	0.0015	0.0022	5.0×10^{-4}

Table 3.1: Averaged tracking errors of the first-order method and the second-order method applied to the problem (3.34).

for $k = 1, 2$. We randomly generate 10 instances of U, and the resulting averaged tracking errors are recorded in Table 3.1. It can be seen that the second-order method achieves better tracking performance over the first-order method for all the instances, which is expected as utilizing second-order information can facilitate convergence toward the optimal trajectory. On the other hand, each iteration of the second-order method involves solving a quadratic-like program, which in general takes more time than performing a projected gradient step. These observations suggest that second-order methods should be employed in situations where the sampling interval Δ needs to be relatively large due to communication delays, non-negligible dynamics, delays from sensors, etc. and, as a consequence, first-order methods don't perform well.

3.5 Summary

In this chapter, we proposed and analyzed the approximate Newton methods that incorporate second-order information for time-varying optimization.

We first considered the special case where there are no explicit inequality or equality constraints. In this case each iteration of the approximate Newton method solves a convex program that can be viewed as a quadratic approximation of the original problem. It was shown that good estimation of second-order information is important for achieving satisfactory tracking performance. We also proposed a specific version of the approximate Newton method based on L-BFGS-B that handles box constraints.

Then, for the general case where there are nonlinear equality and inequality constraints, we proposed two variants of the approximate Newton method. The first variant employs penalty functions to obtain a modified version of the original problem, so that the approximate Newton method for the special case can be applied. We showed that the difference between optimal solutions of the original problem and the penalized problem can be upper bounded under certain conditions. The second variant can be viewed as an extension of sequential quadratic programming in the time-varying setting, and regularization on the dual variable is also employed. We performed a direct analysis of the tracking error bound with respect to the optimal trajectory and discussed its implications.

Finally, we used a toy example to compare first-order and second-order time-varying optimization algorithms, which helps us understand the situations where second-order methods are more appropriate than first-order methods.

3.A Proofs

This section provides the proofs of the theorems that have been skipped in the text.

Proof of Theorem 3.2

Denote

$$A = \begin{bmatrix} a_1 & \cdots & a_{m'} \end{bmatrix}^T \in \mathbb{R}^{m' \times n}$$

and $x^* = \mathcal{P}_{X^*}(\tilde{x}_\epsilon)$. The KKT conditions to (3.18) are given by

$$
\begin{aligned}
(x^*, \lambda^*, \mu^*) &\in \mathrm{dom}(h) \times \mathbb{R}_+^m \times \mathbb{R}^{m'}, \\
-\nabla c(x^*) - \begin{bmatrix} J_{f^{in}}(x^*) \\ A \end{bmatrix}^T \begin{bmatrix} \lambda^* \\ \mu^* \end{bmatrix} &\in \partial h(x^*), \\
f^{in}(x^*) \leq 0, \quad \lambda^{*T} f^{in}(x^*) &= 0, \\
Ax^* &= b,
\end{aligned}
$$

from which we can see that

$$
h(\tilde{x}_\epsilon) - h(x^*) \geq -(\tilde{x}_\epsilon - x^*)^T \left(\nabla c(x^*) + \begin{bmatrix} J_{f^{in}}(x^*) \\ A \end{bmatrix}^T \begin{bmatrix} \lambda^* \\ \mu^* \end{bmatrix} \right). \tag{3.35}
$$

Let

$$
\begin{aligned}
\mathcal{L}^\epsilon(x, \lambda, \mu) = c(x) + h(x) + \lambda^T f^{in}(x) + \mu^T (Ax - b) \\
- \frac{\epsilon^{\beta-1}}{\beta} \left(\sum_{i=1}^m |\lambda_i|^\beta + \sum_{j=1}^{m'} |\mu_j|^\beta \right),
\end{aligned}
$$

where $\beta = \kappa/(\kappa - 1)$, and

$$
\begin{aligned}
\tilde{\lambda}_{\epsilon,i} &= \epsilon^{-1} \left[f_i^{in}(\tilde{x}_\epsilon) \right]_+^{\kappa-1}, & i &= 1, \ldots, m, \\
\tilde{\mu}_{\epsilon,j} &= \epsilon^{-1} \operatorname{sgn}\left(a_j^T \tilde{x}_\epsilon - b_j \right) \left| a_j^T \tilde{x}_\epsilon - b_j \right|^{\kappa-1} & j &= 1, \ldots, m'.
\end{aligned}
$$

We have shown that the min-max problem

$$
\min_{x \in \mathcal{R}^n} \max_{\lambda \in \mathbb{R}_+^m, \mu \in \mathbb{R}^{m'}} \mathcal{L}^\epsilon(x, \lambda, \mu) \tag{3.36}
$$

is equivalent to the penalized problem (3.20), whose solution is given by $(\tilde{x}_\epsilon, \tilde{\lambda}_\epsilon, \tilde{\mu}_\epsilon)$. Then we have

$$
\begin{aligned}
& L^\epsilon\left(\tilde{x}_\epsilon, \tilde{\lambda}_\epsilon, \tilde{\mu}_\epsilon\right) \geq L^\epsilon\left(\tilde{x}_\epsilon, \lambda^*, \mu^*\right) \\
\implies \quad & \frac{\epsilon^{\beta-1}}{\beta} \left(\sum_{i=1}^m \left|\tilde{\lambda}_{\epsilon,i}\right|^\beta + \sum_{j=1}^{m'} \left|\tilde{\mu}_{\epsilon,j}\right|^\beta \right) \\
& \leq \begin{bmatrix} \tilde{\lambda}_\epsilon - \lambda^* \\ \tilde{\mu}_\epsilon - \mu^* \end{bmatrix}^T \begin{bmatrix} f^{in}(\tilde{x}_\epsilon) \\ A\tilde{x}_\epsilon - b \end{bmatrix} + \frac{\epsilon^{\beta-1}}{\beta} \left(\sum_{i=1}^m \left|\lambda_i^*\right|^\beta + \sum_{j=1}^{m'} \left|\mu_j^*\right|^\beta \right).
\end{aligned}
$$

By $f^{in}(x^*) \leq 0$ and the complementary slackness condition $\lambda^{*T} f^{in}(x^*) = 0$,

$$
\left(\tilde{\lambda}_\epsilon - \lambda^*\right)^T f^{in}(x^*) \leq 0,
$$

and together with $Ax^* = b$, we get

$$\begin{aligned} &\frac{\epsilon^{\beta-1}}{\beta}\left(\sum_{i=1}^{m}\left|\tilde{\lambda}_{\epsilon,i}\right|^{\beta} + \sum_{j=1}^{m'}\left|\tilde{\mu}_{\epsilon,j}\right|^{\beta}\right) \\ &\leq \begin{bmatrix}\tilde{\lambda}_{\epsilon}-\lambda^* \\ \tilde{\mu}_{\epsilon}-\mu^*\end{bmatrix}^T \begin{bmatrix} f^{in}(\tilde{x}_{\epsilon}) - f^{in}(x^*) \\ A\tilde{x}_{\epsilon} - Ax^*\end{bmatrix} + \frac{\epsilon^{\beta-1}}{\beta}\left(\sum_{i=1}^{m}\left|\lambda_i^*\right|^{\beta} + \sum_{j=1}^{m'}\left|\mu_j^*\right|^{\beta}\right). \end{aligned} \tag{3.37}$$

It can be checked that the optimality condition for (3.20) is given by

$$-\nabla c(\tilde{x}_{\epsilon}) - \begin{bmatrix} J_{f^{in}}(\tilde{x}_{\epsilon}) \\ A\end{bmatrix}^T \begin{bmatrix}\tilde{\lambda}_{\epsilon} \\ \tilde{\mu}_{\epsilon}\end{bmatrix} \in \partial_h(\tilde{x}_{\epsilon}),$$

and therefore

$$h(x^*) - h(\tilde{x}_{\epsilon}) \geq -(x^* - \tilde{x}_{\epsilon})^T\left(\nabla c(\tilde{x}_{\epsilon}) + \begin{bmatrix} J_{f^{in}}(\tilde{x}_{\epsilon}) \\ A\end{bmatrix}^T \begin{bmatrix}\tilde{\lambda}_{\epsilon} \\ \tilde{\mu}_{\epsilon}\end{bmatrix}\right). \tag{3.38}$$

By combining (3.38) with (3.35), we get

$$(\tilde{x}_{\epsilon} - x^*)^T\left(\nabla c(\tilde{x}_{\epsilon}) - \nabla c(x^*) + \begin{bmatrix} J_{f^{in}}(\tilde{x}_{\epsilon}) \\ A\end{bmatrix}^T \begin{bmatrix}\tilde{\lambda}_{\epsilon} \\ \tilde{\mu}_{\epsilon}\end{bmatrix} - \begin{bmatrix} J_{f^{in}}(x^*) \\ A\end{bmatrix}^T \begin{bmatrix}\lambda^* \\ \mu^*\end{bmatrix}\right) \leq 0.$$

The assumption (3.19) then indicates that

$$\nu\|\tilde{x}_{\epsilon} - x^*\|^2 \leq (\tilde{x}_{\epsilon} - x^*)^T\left(\begin{bmatrix} J_{f^{in}}(x^*) \\ A\end{bmatrix}^T \begin{bmatrix}\lambda^* \\ \mu^*\end{bmatrix} - \begin{bmatrix} J_{f^{in}}(\tilde{x}_{\epsilon}) \\ A\end{bmatrix}^T \begin{bmatrix}\tilde{\lambda}_{\epsilon} \\ \tilde{\mu}_{\epsilon}\end{bmatrix}\right). \tag{3.39}$$

Summing (3.39) and (3.37), we obtain

$$\begin{aligned} &\nu\|\tilde{x}_{\epsilon} - x^*\|^2 + \frac{\epsilon^{\beta-1}}{\beta}\left(\sum_{i=1}^{m}\left|\tilde{\lambda}_{\epsilon,i}\right|^{\beta} + \sum_{j=1}^{m'}\left|\tilde{\mu}_{\epsilon,j}\right|^{\beta}\right) \\ &\leq \tilde{\lambda}_{\epsilon}^T\left(f^{in}(\tilde{x}_{\epsilon}) + J_{f^{in}}(\tilde{x}_{\epsilon})(x^* - \tilde{x}_{\epsilon}) - f^{in}(x^*)\right) \\ &\quad \lambda^{*T}\left(f^{in}(x^*) + J_{f^{in}}(x^*)(\tilde{x}_{\epsilon} - x^*) - f^{in}(\tilde{x}_{\epsilon})\right) + \frac{\epsilon^{\beta-1}}{\beta}\left(\sum_{i=1}^{m}\left|\lambda_i^*\right|^{\beta} + \sum_{j=1}^{m'}\left|\mu_j^*\right|^{\beta}\right). \end{aligned}$$

Now, since each component of f^{in} is convex, we have

$$\begin{aligned} f^{in}(\tilde{x}_{\epsilon}) + J_{f^{in}}(\tilde{x}_{\epsilon})(x^* - \tilde{x}_{\epsilon}) - f^{in}(x^*) &\leq 0, \\ f^{in}(x^*) + J_{f^{in}}(x^*)(\tilde{x}_{\epsilon} - x^*) - f^{in}(\tilde{x}_{\epsilon}) &\leq 0, \end{aligned}$$

and then by $\tilde{\lambda}_{\epsilon} \geq 0$ and $\lambda^* \geq 0$, we get

$$\nu\|\tilde{x}_{\epsilon} - x^*\|^2 + \frac{\epsilon^{\beta-1}}{\beta}\left(\sum_{i=1}^{m}\left|\tilde{\lambda}_{\epsilon,i}\right|^{\beta} + \sum_{j=1}^{m'}\left|\tilde{\mu}_{\epsilon,j}\right|^{\beta}\right) \leq \frac{\epsilon^{\beta-1}}{\beta}\left(\sum_{i=1}^{m}\left|\lambda_i^*\right|^{\beta} + \sum_{j=1}^{m'}\left|\mu_j^*\right|^{\beta}\right),$$

which is just the bound (3.21). ∎

Proof of Theorem 3.3

Let $\mathcal{I}$ denote set of indices of active constraints in $f^{in}(x^*) \leq 0$ and $\mathcal{J}$ denote the set of indices of active constraints in $g^{in}(x^*) \leq 0$. Without loss of generality we assume that

$$\mathcal{I} = \{1, 2, \ldots, |\mathcal{I}|\}, \quad \mathcal{J} = \{1, 2, \ldots, |\mathcal{J}|\},$$

and consequently $\mathcal{I}^c = \{|\mathcal{J}| + 1, \ldots, m\}$, $\mathcal{J}^c = \{|\mathcal{J}| + 1, \ldots, p\}$. Define $u^* \in \mathbb{R}^m_+$ and $v^* \in \mathbb{R}^{m'}$ by

$$\begin{aligned} u_i^* &= \lambda_i^{*\frac{1}{\kappa-1}}, && i = 1, \ldots, m, \\ v_j^* &= \operatorname{sgn}(\mu_j^*)|\mu_j^*|^{\frac{1}{\kappa-1}}, && j = 1, \ldots, m'. \end{aligned}$$

Consider the following set of generalized equations in $z = (x, v, \upsilon, u, \zeta)$:

$$\begin{aligned} \nabla c(x) + \sum_{i=1}^{m} u_i^{\kappa-1} \nabla f_i^{in}(x) + \sum_{j=1}^{m'} \operatorname{sgn}(v_j)|v_j|^{\kappa-1} \nabla f_j^{eq}(x) + \begin{bmatrix} J_{g^{in}}(x) \\ J_{g^{eq}}(x) \end{bmatrix}^T \begin{bmatrix} \zeta \\ \upsilon \end{bmatrix} &= 0, \\ -f^{eq}(x) + \varepsilon v &= 0, \\ -g^{eq}(x) &= 0, \\ -f^{in}(x) + \varepsilon u + N_{\mathbb{R}^m_+}(u) &\ni 0, \\ -g^{in}(x) + N_{\mathbb{R}^p_+}(\zeta) &\ni 0. \end{aligned} \tag{3.40}$$

It can be checked that $z^* := (x^*, v^*, \upsilon^*, u^*, \zeta^*)$ is a solution to (3.40) for $\varepsilon = 0$, while for $\varepsilon = \epsilon^{1/(\kappa-1)} > 0$, any solution to (3.40) provides a local optimal solution to the penalized problem (3.23).

Denote $\mathcal{K} := \mathbb{R}^{n+m'+p'} \times \mathbb{R}^{m+p}_+$. We introduce the map $\Phi : \mathbb{R}_+ \times \mathcal{K} \to \mathcal{K}$ such that

$$\Phi(\varepsilon, z) + N_{\mathcal{K}}(z) \ni 0$$

is a shorthand form of (3.40). Let

$$\Psi(z) := \Phi(0, z^*) + J_{\Phi,z}(0, z^*)(z - z^*) + N_{\mathcal{K}}(z).$$

The set-valued map $\Psi(z)$ can be viewed as the "linearization" of the set-valued map $\Phi(0, z) + N_{\mathcal{K}}(z)$ near z^*. Now consider the generalized equation $w = \Psi(z)$ where

$w \in \mathcal{K}$ is some given vector. It can be checked that $w = \Psi(z)$ is explicitly given by

$$\begin{aligned} H^*(x-x^*) + J_{f^{in}_{\mathcal{I}}}(x^*)^T \tilde{U}^* \left(u_{\mathcal{I}} - u^*_{\mathcal{I}}\right) + J_{f^{eq}}(x^*)^T \tilde{V}^*(v-v^*) & \\ + J_{g^{in}}(x^*)^T(\zeta-\zeta^*) + J_{g^{eq}}(x^*)^T(\upsilon-\upsilon^*) &= w_x, \\ -J_{f^{eq}}(x^*)(x-x^*) &= w_v, \\ -J_{g^{eq}}(x^*)(x-x^*) &= w_\upsilon, \\ -J_{f^{in}_{\mathcal{I}}}(x^*)(x-x^*) + N_{\mathbb{R}_+^{|\mathcal{I}|}}(u_{\mathcal{I}}) &\ni w_{u,\mathcal{I}}, \\ -f^{in}_{\mathcal{I}^c}(x^*) - J_{f^{in}_{\mathcal{I}^c}}(x^*)(x-x^*) + N_{\mathbb{R}_+^{|\mathcal{I}^c|}}(u_{\mathcal{I}^c}) &\ni w_{u,\mathcal{I}^c}, \\ -J_{g^{in}_{\mathcal{J}}}(x^*)(x-x^*) + N_{\mathbb{R}_+^{|\mathcal{J}|}}(\zeta_{\mathcal{J}}) &\ni w_{\zeta,\mathcal{J}}, \\ -g^{in}_{\mathcal{J}^c}(x^*) - J_{g^{in}_{\mathcal{J}^c}}(x^*)(x-x^*) + N_{\mathbb{R}_+^{|\mathcal{J}^c|}}(\zeta_{\mathcal{J}^c}) &\ni w_{\zeta,\mathcal{J}^c}, \end{aligned} \tag{3.41}$$

where

$$\begin{aligned} H^* &= \nabla^2_{xx}\mathcal{L}(x^*, \lambda^*, \mu^*, \zeta^*, \upsilon^*), \\ \tilde{U}^* &= (\kappa-1)\,\mathrm{diag}\left({u_1^*}^{\kappa-2}, \ldots, {u^*_{|\mathcal{I}|}}^{\kappa-2}\right), \\ \tilde{V}^* &= (\kappa-1)\,\mathrm{diag}\left(\left|v_1^*\right|^{\kappa-2}, \ldots, \left|v^*_{m'}\right|^{\kappa-2}\right), \end{aligned}$$

and $w_x, w_v, w_\upsilon, w_{u,\mathcal{I}}, w_{u,\mathcal{I}^c}, w_{\zeta,\mathcal{J}}, w_{\zeta,\mathcal{J}^c}$ are the corresponding subvectors of w. By the positive definiteness of H^*, the linear independence constraint qualification, the strict complementary slackness, and the assumption that the entries of μ^* are all nonzero, we can see that the matrix

$$\begin{bmatrix} H^* & J_{f^{eq}}(x^*)^T\tilde{V}^* & J_{g^{eq}}(x^*)^T & J_{f^{in}_{\mathcal{I}}}(x^*)^T\tilde{U}^* & J_{g^{in}_{\mathcal{J}}}(x^*)^T \\ J_{f^{eq}}(x^*) & 0 & 0 & 0 & 0 \\ J_{g^{eq}}(x^*) & 0 & 0 & 0 & 0 \\ J_{f^{in}_{\mathcal{I}}}(x^*) & 0 & 0 & 0 & 0 \\ J_{g^{in}_{\mathcal{J}}}(x^*) & 0 & 0 & 0 & 0 \end{bmatrix}$$

is invertible. Then it can be verified that when $\|w\|$ is sufficiently small, the generalized equation (3.41) can be reduced to

$$\begin{bmatrix} H^* & J_{f^{eq}}(x^*)^T\tilde{V}^* & J_{g^{eq}}(x^*)^T & J_{f^{in}_{\mathcal{I}}}(x^*)^T\tilde{U}^* & J_{g^{in}_{\mathcal{J}}}(x^*)^T \\ J_{f^{eq}}(x^*) & 0 & 0 & 0 & 0 \\ J_{g^{eq}}(x^*) & 0 & 0 & 0 & 0 \\ J_{f^{in}_{\mathcal{I}}}(x^*) & 0 & 0 & 0 & 0 \\ J_{g^{in}_{\mathcal{J}}}(x^*) & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x-x^* \\ v-v^* \\ \upsilon-\upsilon^* \\ u_{\mathcal{I}}-u^*_{\mathcal{I}} \\ \zeta_{\mathcal{J}}-\zeta^*_{\mathcal{J}} \end{bmatrix} = \begin{bmatrix} w_x \\ w_v \\ w_\upsilon \\ w_{u,\mathcal{I}} \\ w_{\zeta,\mathcal{J}} \end{bmatrix},$$

$$\begin{aligned} u_{\mathcal{I}^c} &= 0, \\ \zeta_{\mathcal{J}^c} &= 0. \end{aligned}$$

It's not hard to verify that this set of equations has a unique solution that depends Lipschitz continuously on w. In other words, we have shown that Ψ^{-1}, the inverse of the set-valued map Ψ, has a Lipschitz continuous single-valued localization around 0 for z. It's also not hard to check that $\Phi(\varepsilon, z)$ and $J_{\Phi,\varepsilon}(\varepsilon, z)$ is continuous over $(\varepsilon, z) \in \mathbb{R}_+ \times \mathcal{K}$. Therefore by [40, Theorem 2B.1], there exist some $\bar{\varepsilon} > 0$, $M > 0$ and a unique continuous function $\tilde{z} : [0, \bar{\varepsilon}) \to \mathcal{K}$ such that $\tilde{z}(0) = z^*$ and $\tilde{z}(\varepsilon)$ solves the generalized equation (3.40) for each $\varepsilon \in (0, \bar{\varepsilon})$. Moreover,

$$\|\tilde{z}(\varepsilon) - z^*\|_{\ell_\kappa} \leq M \left\|\Phi\left(\varepsilon, z^*\right) - \Phi(0, z^*)\right\|_{\ell_\kappa} = \varepsilon M \left(\sum_{i=1}^{m} \left|\lambda_i^*\right|^{\frac{\kappa}{\kappa-1}} + \sum_{i=1}^{m} \left|\mu_i^*\right|^{\frac{\kappa}{\kappa-1}}\right)^{\frac{1}{\kappa}}.$$

Then by noting that the first n entries of $\tilde{z}\left(\epsilon^{1/(\kappa-1)}\right)$ form a local solution to the penalized problem (3.23), we get the desired conclusions, with $\bar{\epsilon} = \bar{\varepsilon}^{\kappa-1}$. ■

Chapter 4

APPLICATIONS IN POWER SYSTEM OPERATION

4.1 The Time-Varying Optimal Power Flow Problem

In this chapter, we discuss the applications of time-varying optimization in power system operation, with a focus on the time-varying optimal power flow problem and real-time optimal power flow algorithms.

As mentioned in Chapter 1, in future smart grids, the fluctuations and uncertainties introduced by large penetration of renewable generation make the operation of smart grids challenging, while controllable devices provide diverse control capabilities that can be employed to overcome the challenges. Extensive real-time measurement data will also become available by smart meters and other advanced measurement equipment. When time-varying optimization tools are applied for real-time power system operation, these factors should be taken into consideration to tailor the algorithms so that the structures and properties of smart grids can be utilized.

We now present the formulation of the time-varying optimal power flow problem. Suppose we have a single-phase power network[1] with a topology represented by a connected graph $(\mathcal{N}^+, \mathcal{E})$ where $\mathcal{N}^+ := \{0\} \cup \mathcal{N}$, $\mathcal{N} = \{1, 2, \ldots, n\}$ and $\mathcal{E} \subseteq \mathcal{N}^+ \times \mathcal{N}^+$. Bus 0 will be the slack bus, and the phase angle of its voltage will be the reference and taken as zero. Let $t \in [0, T]$ be an arbitrary time instant. We use $v(t) \in \mathbb{R}^{\mathcal{N}^+}$ and $\theta(t) \in \mathbb{R}^{\mathcal{N}}$ to denote respectively the vector of voltage magnitudes and the vector of voltage phase angles, so that $V_i(t) := v_i(t)e^{\mathrm{j}\theta_i(t)}$ is the voltage phasor at bus $i \in \mathcal{N}$ and $V_0(t) := v_0(t)$ is the voltage phasor at the slack bus. We use $\ell(t) \in \mathbb{R}^{\mathcal{E}}$ to record the squared current magnitudes through the lines, where $\ell_{ik}(t)$ denotes the squared current magnitude associated with the edge $(i, k) \in \mathcal{E}$. We use $p(t) \in \mathbb{R}^{\mathcal{N}^+}$ and $q(t) \in \mathbb{R}^{\mathcal{N}^+}$ to denote the vectors of real power injections and reactive power injections of the controllable devices respectively, so that $p_i(t) + \mathrm{j}q_i(t)$ is the complex power injection of the controllable devices connected to bus i. We use $p^{\mathrm{L}}(t)$ and $q^{\mathrm{L}}(t)$ to denote the vectors of real and reactive uncontrollable loads respectively, and $p(t) - p^{\mathrm{L}}(t)$ and $q(t) - q^{\mathrm{L}}(t)$ then represent the net real and reactive power injections.

[1] The basic ideas and principles of real-time optimal power flow algorithms can be applied to the operation of three-phase networks as well, which for simplicity will be omitted here.

The relation between $v(t)$, $\theta(t)$, $\ell(t)$, $p(t)$, $q(t)$, $p^{\mathrm{L}}(t)$ and $q^{\mathrm{L}}(t)$ is described by physical laws and usually can be written as a set of algebraic equations (when one is interested in the steady state behavior) or ordinary differential equations (when dynamics needs to be accounted for). For steady states, the *power flow equations*

$$p_i(t) - p_i^{\mathrm{L}}(t) + \mathrm{j}\big(q_i(t) - \mathrm{j}q_i^{\mathrm{L}}(t)\big) = \sum_{k\in\mathcal{N}^+} V_i(t)V_k^*(t)Y_{ik}^*, \qquad \forall i \in \mathcal{N}^+$$

or

$$\begin{aligned}
0 = &-(p_i(t) - p_i^{\mathrm{L}}(t)) \\
&+ \sum_{k\in\mathcal{N}^+} v_i(t)v_k(t)(G_{ik}\cos(\theta_i(t)-\theta_k(t)) + B_{ik}\sin(\theta_i(t)-\theta_k(t))), \\
0 = &-(q_i(t) - q_i^{\mathrm{L}}(t)) \\
&+ \sum_{k\in\mathcal{N}^+} v_i(t)v_k(t)(G_{ik}\sin(\theta_i(t)-\theta_k(t)) - B_{ik}\cos(\theta_i(t)-\theta_k(t))), \quad \forall i \in \mathcal{N}^+
\end{aligned} \tag{4.1a}$$

are satisfied, where $Y_{ik} = G_{ik} + \mathrm{j}B_{ik}$ is the (i,k)'th entry of the admittance matrix of the network, and we let $\theta_0(t) \equiv 0$. The squared current magnitudes satisfy

$$\begin{aligned}
0 &= -\ell_{ik}(t) + |Y_{ik}(V_i(t) - V_k(t))|^2 \\
&= -\ell_{ik}(t) + |Y_{ik}|^2\left(v_i^2(t) + v_k^2(t) - 2v_i(t)v_k(t)\cos\left(\theta_i(t)-\theta_k(t)\right)\right), \quad \forall (i,k)\in\mathcal{E}
\end{aligned} \tag{4.1b}$$

We abbreviate the set of equations (4.1) as

$$\mathcal{G}\big(p(t) - p^{\mathrm{L}}(t), q(t) - p^{\mathrm{L}}(t), v(t), \theta(t), \ell(t)\big) = 0, \tag{4.2}$$

where the vector-valued function $\mathcal{G}$ has $2(n+1) + |\mathcal{E}|$ entries that are given by the right-hand side of (4.1).

For each bus, there are physical constraints on how much power can be injected by the controllable devices. We assume that they can be modeled by time-varying constraints

$$s_i(p_i(t), q_i(t), t) \le 0, \quad i \in \mathcal{N}^+,$$

where each $s_i(\cdot, t)$ is a vector-valued continuously differentiable function with convex components over $\mathbb{R}^2$ for every $t \in \mathcal{T}$, and the set $\{(p_i, q_i) : s_i(p_i, q_i, t) \le 0\}$ is compact. Information about the capabilities of controllable devices can be encoded in these time-dependent constraints.

The voltages and currents also need to be bounded for operational reasons. We assume that they are given by

$$\begin{aligned}
\underline{v}_i \le v_i(t) &\le \overline{v}_i, \quad i \in \mathcal{N}^+, \\
\ell_{ik}(t) &\le \overline{\ell}_{ik}, \quad (i,k)\in\mathcal{E}.
\end{aligned}$$

For each bus i, we assume that a cost $c_i(p_i, q_i, t)$ will be incurred when complex power $p_i + \mathrm{j}q_i$ is injected by the controllable devices at bus i into the network at time t. The cost functions can be potentially time dependent. We also assume that they are all convex functions of (p_i, q_i) and are twice continuously differentiable for each fixed t, and that the Hessian of c_i with respect to (p_i, q_i) is continuous over $(p_i, q_i, t) \in \mathbb{R}^2 \times [0, T]$.

The time-varying optimal power flow problem we shall consider is then formulated as

$$\begin{aligned} \min_{p,q,v,\theta,\ell} \quad & \sum_{i\in\mathcal{N}^+} c_i(p_i, q_i, t) \\ \text{s.t.} \quad & \mathcal{G}\big(p - p^{\mathrm{L}}(t), q - p^{\mathrm{L}}(t), v, \theta, \ell\big) = 0, \\ & s_i(p_i, q_i, t) \le 0, \quad i \in \mathcal{N}^+ \\ & \underline{v}_i \le v_i \le \overline{v}_i, \quad i \in \mathcal{N}^+ \\ & \ell_{ik} \le \overline{\ell}_{ik}, \quad (i,k) \in \mathcal{E}. \end{aligned} \tag{4.3}$$

The goal is to find a sufficiently good sub-optimal solution $\hat{p}_\tau \in \mathbb{R}^{\mathcal{N}^+}$, $\hat{q}_\tau \in \mathbb{R}^{\mathcal{N}^+}$, $\hat{v}_\tau \in \mathbb{R}^{\mathcal{N}^+}$ $\hat{\theta}_\tau \in \mathbb{R}^{\mathcal{N}}$ and $\hat{\ell}_\tau \in \mathbb{R}^{\mathcal{E}}$ to (4.3) for each $t = \tau\Delta$. We see that (4.3) is within the general framework of time-varying optimization[2]. When a time-varying optimization algorithm is tailored and applied to (4.3), we call the resulting algorithm a *real-time optimal power flow* algorithm.

As usual, we let $(p^*(t), q^*(t), v^*(t), \theta^*(t), \ell^*(t))$ denote a trajectory of local optimal solutions to (4.3).

Power System as an Implicit Map

While we can directly apply the regularized proximal primal-dual method or the approximate Newton method to (4.3), we note that the power flow equations (4.2) will in general not be satisfied by the resulting solution $(\hat{p}_\tau, \hat{q}_\tau, \hat{v}_\tau, \hat{\theta}_\tau, \hat{\ell}_\tau)$. Moreover, we observe that, given a subvector of $(p(t), q(t), v(t), \theta(t), \ell(t))$ with appropriate dimension and value, the remaining entries can be determined implicitly by the power flow equations (4.2). This observation suggests that we can partition $(p(t), q(t), v(t), \theta(t), \ell(t))$ into two subvectors, which we denote by $x(t)$ and $y(t)$, so that there is an implicit function derived from the power flow equations [in some

[2] Strictly speaking, the formulation (4.3) assumes that the behavior of the power network can be described by the steady-state power flow equations for all t, which is an approximation considering that the power network has its own dynamics that needs to be described by differential equations. How to incorporate the dynamics will be left for future work.

neighborhood of $(p^*(t), q^*(t), v^*(t), \theta^*(t), \ell^*(t))$] that maps $x(t)$ to $y(t)$. We refer to $x(t)$ as the input variable, and $y(t)$ as the state variable.

Specifically, we construct two operators

$$\mathsf{P}_x : \mathbb{R}^{\mathcal{N}^+} \times \mathbb{R}^{\mathcal{N}^+} \times \mathbb{R}^{\mathcal{N}^+} \times \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^{\mathcal{E}} \to \mathbb{R}^{d_x},$$
$$\mathsf{P}_y : \mathbb{R}^{\mathcal{N}^+} \times \mathbb{R}^{\mathcal{N}^+} \times \mathbb{R}^{\mathcal{N}^+} \times \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^{\mathcal{E}} \to \mathbb{R}^{d_y},$$

where $d_x, d_y \in \mathbb{N}$ and $d_x + d_y = 4n + 3 + |\mathcal{E}|$, such that $(\mathsf{P}_x, \mathsf{P}_y)$ permutes the entries of the vectors in $\mathbb{R}^{\mathcal{N}^+} \times \mathbb{R}^{\mathcal{N}^+} \times \mathbb{R}^{\mathcal{N}^+} \times \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^{\mathcal{E}}$ to form a new vector in $\mathbb{R}^{4n+3+|\mathcal{E}|}$. The partition of $(p(t), q(t), v(t), \theta(t), \ell(t))$ into $x(t)$ and $y(t)$ can then be represented by

$$x(t) = \mathsf{P}_x(p(t), q(t), v(t), \theta(t), \ell(t)), \qquad y(t) = \mathsf{P}_y(p(t), q(t), v(t), \theta(t), \ell(t)).$$

We assume that this partition does not depend on t. Denote

$$x^L(t) := \mathsf{P}_x\big(p^{\mathrm{L}}(t), q^{\mathrm{L}}(t), 0, 0, 0\big),$$
$$y^L(t) := \mathsf{P}_y\big(p^{\mathrm{L}}(t), q^{\mathrm{L}}(t), 0, 0, 0\big),$$

and

$$x^*(t) := \mathsf{P}_x\big(p^*(t), q^*(t), v^*(t), \theta^*(t), \ell^*(t)\big),$$
$$y^*(t) := \mathsf{P}_y\big(p^*(t), q^*(t), v^*(t), \theta^*(t), \ell^*(t)\big).$$

The power flow equations (4.2) can then be rewritten in the form

$$\mathcal{G}^{\mathsf{P}}\big(x(t) - x^L(t), y(t) - y^L(t)\big) = 0. \tag{4.4}$$

By [53, Proposition 1], the $2(n+1) + |\mathcal{E}|$ equations in (4.4) are independent generically. Then the requirement that an implicit function can be derived that maps $x(t)$ to $y(t)$ in some neighborhood of $(x^*(t), y^*(t))$ suggests that $d_x = 2n + 1$ and $d_y = 2(n+1) + |\mathcal{E}|$.

Proposition 4.1. *Suppose $x^*(t)$, $y^*(t)$, $x^{\mathrm{L}}(t)$ and $y^{\mathrm{L}}(t)$ are all continuous functions over $t \in [0, T]$. Let $\mathcal{U} \subset \mathbb{R}^{2n+1}$ be open and simply connected such that $x^*(t) - x^{\mathrm{L}}(t) \in \mathcal{U}$ for all $t \in [0, T]$, and let $\mathcal{V} \subseteq \mathbb{R}^{2(n+1)+|\mathcal{E}|}$ be open such that $y^*(t) - y^{\mathrm{L}}(t) \in \mathcal{V}$ for all $t \in [0, T]$. Let $\mathcal{C} \subseteq \mathcal{U} \times \mathcal{V}$ denote the path-connected component of*

$$\{(x, y) \in \mathcal{U} \times \mathcal{V} : \mathcal{G}^{\mathsf{P}}(x, y) = 0\}$$

such that $\big(x^*(t) - x^{\mathrm{L}}(t), y^*(t) - y^{\mathrm{L}}(t)\big) \in \mathcal{C}$ for all $t \in [0, T]$. Suppose the following assumptions hold:

1. *For every $x \in \mathcal{U}$, there exists $y \in \mathcal{V}$ such that $(x, y) \in C$ and the Jacobian $J_{\mathcal{G}^{\mathsf{P}},y}(x, y)$ is nonsingular.*
2. *For any $(x_0, y_0) \in C$ and $x_1 \in \mathcal{U}$, there exists a continuous path $\gamma : [0, 1] \to C$ such that $\gamma(0) = (x_0, y_0)$ and $\pi_x \circ \gamma(t) = (1 - t)x_0 + tx_1$ for any $t \in [0, 1]$, where $\pi_x : C \to \mathcal{U}$ denotes the canonical projection that maps $(x, y) \in C$ to $x \in \mathcal{U}$.*

Then there exists a unique smooth map $\tilde{\mathcal{F}} : \mathcal{U} \to \mathcal{V}$ such that for each $t \in [0, T]$,

$$x^*(t) - x^{\mathsf{L}}(t) \in \mathcal{U}, \qquad y^*(t) - y^{\mathsf{L}}(t) = \tilde{\mathcal{F}}(x^*(t) - x^{\mathsf{L}}(t)),$$

and

$$\big(x, \tilde{\mathcal{F}}(x)\big) \in C, \qquad \forall x \in \mathcal{U}.$$

Proof. Let $(x_0, y_0) \in C$ be arbitrary. By the implicit function theorem, there exist an open neighborhood $U_0 \in \mathcal{U}$ of x_0, an open neighborhood $V_0 \in \mathcal{V}$ of y_0, and a diffeomorphism $f_0 : U_0 \to V_0$ such that

$$\{(x, f_0(x)) : x \in U_0\} = C \cap (U_0 \times V_0).$$

This implies that $\pi_x : C \to \mathcal{U}$ is a local diffeomorphism. Then by the assumptions and [76, Theorem 1.1], $\pi_x : C \to \mathcal{U}$ is a covering map. Finally, since C is path-connected and $\mathcal{U}$ is simply connected, by [70, Theorem 54.4], we conclude that $\pi_x : C \to \mathcal{U}$ is a homeomorphism. Let $\tilde{\mathcal{F}} : \mathcal{U} \to \mathcal{V}$ be defined by $\tilde{\mathcal{F}} := \pi_y \pi_x^{-1}$, where $\pi_y : C \to \mathcal{V}$ denotes the canonical projection that maps $(x, y) \in C$ to $y \in \mathcal{V}$. It can be verified that $\tilde{\mathcal{F}}$ satisfies the properties stated in the proposition. The smoothness of $\tilde{\mathcal{F}}$ is guaranteed by the smoothness of $\mathcal{G}^{\mathsf{P}}$. □

Proposition 4.1 is a version of the global implicit function theorems [22]. From now on we assume that, for the partition $(\mathsf{P}_x, \mathsf{P}_y)$, there exists an open set $\mathcal{U} \subseteq \mathbb{R}^{2n+1}$ and an implicit power flow map $\tilde{\mathcal{F}} : \mathcal{U} \to \mathbb{R}^{2(n+1)+|\mathcal{E}|}$ that accords with the optimal trajectory $(x^*(t), y^*(t))$. We denote $\mathcal{U}(t) := \mathcal{U} + x^{\mathsf{L}}(t)$, and

$$\mathcal{F}(x, t) := \tilde{\mathcal{F}}\big(x - x^{\mathsf{L}}(t)\big) + y^{\mathsf{L}}(t)$$

for any (x, t) satisfying $x \in \mathcal{U}(t)$.

By modelling the behavior of the power network through the implicit map $\mathcal{F}$, we can reformulate the time-varying optimal power flow problem (4.3) in the form

$$\begin{aligned} \min_{x \in \mathcal{U}(t)} \quad & c\big(x, \mathcal{F}(x, t), t\big) \\ \text{s.t.} \quad & x \in \mathcal{X}(t), \\ & h\big(x, \mathcal{F}(x, t), t\big) \leq 0, \end{aligned} \tag{4.5}$$

for some $c : \mathbb{R}^{2n+1} \times \mathbb{R}^{2(n+1)+|\mathcal{E}|} \times [0,T] \to \mathbb{R}$, some set-valued map $\mathcal{X} : [0,T] \to 2^{\mathbb{R}^{2n+1}}$ and some map h that is defined over $\mathbb{R}^{2n+1} \times \mathbb{R}^{2(n+1)+|\mathcal{E}|} \times [0,T]$. Without loss of generality, we can assume that $c(x,y,t)$ is twice continuously differentiable over $(x,y) \in \mathbb{R}^{2n+1} \times \mathbb{R}^{2(n+1)+|\mathcal{E}|}$, $\mathcal{X}(t)$ is convex and compact, and $h(x,y,t)$ is continuously differentiable over $(x,y) \in \mathbb{R}^{2n+1} \times \mathbb{R}^{2(n+1)+|\mathcal{E}|}$ for each t. We see that explicit power flow equations are eliminated in (4.5), and we only need to optimize over $x = \mathsf{P}_x(p,q,v,\theta,\ell)$, whose advantage will be discussed in the next subsection[3].

The sampled version of (4.5) will be denoted by

$$\begin{aligned} \min_{x\in\mathcal{U}_\tau} \quad & c_\tau\big(x, \mathcal{F}_\tau(x)\big) \\ \text{s.t.} \quad & x \in \mathcal{X}_\tau, \\ & h_\tau\big(x, \mathcal{F}_\tau(x)\big) \le 0. \end{aligned}$$

Remark 4.1. Note that (4.5) is in general not equivalent to (4.3), unless the implicit function $\mathcal{F}$ is the unique implicit function that can be derived from the power flow equations (4.4) over the feasible set of (4.3). In addition, Proposition 4.1 imposes conditions on the Jacobian $J_{\mathcal{G}^{\mathsf{P}},y}(x,y)$, and consequently imposes constraints on the partition $(\mathsf{P}_x, \mathsf{P}_y)$ as well. In practice, the following factors should be taken into account when we design the partition $(\mathsf{P}_x, \mathsf{P}_y)$:

1. We have seen in Chapters 2 and 3 that the set inclusion constraint $\hat{x}_\tau \in \mathcal{X}_\tau$ will always be satisfied, while the inequality constraints can be violated for the time-varying optimization algorithms we propose. Thus for real-world applications, we usually partition the variables in a way such that $x(t) \in \mathcal{X}(t)$ consists of the constraints that cannot be violated physically (e.g., lower and upper limits of real power injections), while $h(x(t), \mathcal{F}(x(t),t),t) \le 0$ consists of the constraints that can be violated temporarily (e.g., limits on the voltage or current magnitudes).

2. It is usually favored to choose $x(t)$ to be those variables that can be directly controlled (such as power injections), and $y(t)$ to contain the conventional power network state variables (voltage magnitudes and phase angles) and the squared current magnitudes. In practice, it has been empirically observed that, given a "reasonable" set of injections, there is a unique solution to the power flow equations that satisfies the operational constraints in most situations. While counterexamples do exist [5, 59], recent works [18, 41, 42] have identified conditions under which there is a unique power flow solution within a certain domain.

[3] The idea of partitioning variables into two subvectors to eliminate equality constraints has been proposed in the reduced gradient method [1] and has been used for solving optimal power flow problems [38].

As an example, in distribution networks, we usually let

$$x(t) = (v_0(t), p_1(t), q_1(t), \ldots, p_n(t), q_n(t)),$$
$$y(t) = \big(p_0(t), q_0(t), v_1(t), \theta_1(t), \ldots, v_n(t), \theta_n(t), (\ell_{ij}(t))_{(i,j)\in\mathcal{E}}\big).$$

In transmission networks, the situation is more complicated. Bulk generators are connected to PV buses whose real power injections and voltage magnitudes are easier to control directly, suggesting that it is reasonable to put $(p_i(t), v_i(t))$ instead of $(p_i(t), q_i(t))$ in the vector $x(t)$ for a PV bus; however, one has to pay the price that the constraint on the reactive power injection could be violated. For the slack bus, although there are hard physical constraint on the real power injection $p_0(t)$, we still put it in the vector $y(t)$, as in transmission networks we usually have $\sum_{i\in\mathcal{N}^+} p_i(t) \approx 0$ and numerical issues are very likely to occur if we let $x(t)$ consist of all $p_i(t)$. We shall not discuss in detail here heuristic methods that address the issue of violation of hard physical constraints. ■

Remark 4.2. When the partition $(\mathsf{P}_x, \mathsf{P}_y)$ is chosen appropriately, it can be the case that the function h depends on x purely through $\mathcal{F}(x,t)$. In other words, the inequality constraint in (4.5) can be written as $h(\mathcal{F}(x,t),t) \leq 0$. We shall denote the sampled version of h by $h_\tau(y) := h(y, \tau\Delta)$ in this case. ■

From now on we neglect the constraint $x \in \mathcal{U}(t)$ in (4.5). In well-designed power networks, under optimal operations, there should be sufficiently large margins from voltage collapse and other instabilities. When the partition $(\mathsf{P}_x, \mathsf{P}_y)$ is properly chosen, this means that the distance from the optimal solution to (4.3) to the set

$$\Big\{(x,y) \in \mathbb{R}^{2n+1} \times \mathbb{R}^{2(n+1)+|\mathcal{E}|} : \mathcal{G}^{\mathsf{P}}\big(x - x^{\mathsf{L}}(t), y - y^{\mathsf{L}}(t)\big) = 0,$$
$$J_{\mathcal{G}^{\mathsf{P}},y}\big(x - x^{\mathsf{L}}(t), y - y^{\mathsf{L}}(t)\big) \text{ is singular}\Big\}$$

should be sufficiently large, and the boundary of $\mathcal{U}(t)$ can be pushed sufficiently far from $x^*(t)$ for each t. It is then usually safe to neglect the constraint $x \in \mathcal{U}(t)$ as long as the tracking error is sufficiently small without considering this constraint.

Incorporating Feedback Measurements

In the previous subsection, we provide a mathematical theory on the implicit power flow map, which is employed as a model for the power system. In this subsection, we discuss why we employ the implicit power flow map from an engineering perspective.

Figure 4.1 shows the diagram of the control of a time-varying system. Here the system's state variable $y(t)$ is determined by the input variable $x(t)$ through the time-varying relation $y(t) = \mathcal{F}(x(t), t)$. The time-varying components can be introduced

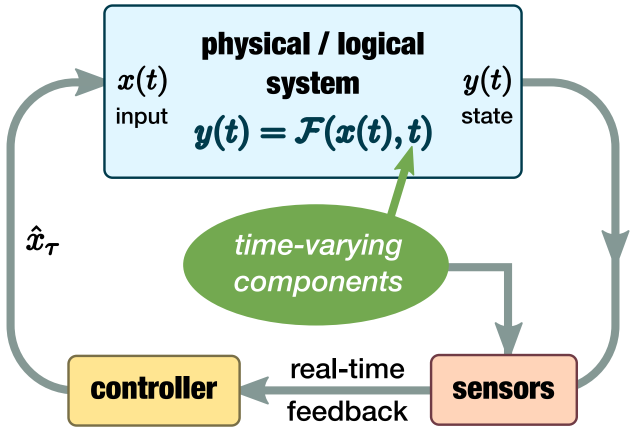

Figure 4.1: Diagram of the control of a time-varying system. The system's input-state relation is given by $\mathcal{F}$, which is influenced by some time-varying components.

by some exogenous sources that interact with the system but are hard to predict. Sensors are installed that measure the state of the system and collect relevant data of time-varying components in real-time. The controller uses feedback measurement provided by sensors to determine a setpoint $\hat{x}_\tau$ for the system at each sampled time instant τ. We can readily recognize that Figure 4.1 provides a paradigm of real-time control of future power networks, where distributed controllable devices are jointly operated to handle fast-timescale fluctuations and uncertainties introduced by renewable generation, with the help of real-time measurements provided by advanced measurement equipment.

The modelling of power systems as an input-state map allows us to better incorporate the real-time measurement data. Mathematically, the computation of the implicit power flow map $\mathcal{F}$ requires solving the power flow equations. On the other hand, when real-time measurement data are available, the power network itself can then be used as a solver for the power flow equations.

To illustrate the basic idea, let us consider an ideal case where accurate measurement data can be obtained without delays and the dynamics of the power network can be safely ignored. Then when we want to get the value of $\mathcal{F}(x, t)$ for some x, we can apply x to the power network, and then use the sensors to measure the state y.

The use of real-time feedback measurements is a central idea in the real-time optimal power flow algorithms, and leads to the following benefits:

1. The power flow equations are a central constraint in the optimal power flow problem, and most optimization algorithms either explicitly or implicitly solve the power flow equations. The use of feedback measurement data potentially helps reduce the computation burden caused by solving the power flow equations and increase efficiency, which is crucial in the time-varying setting.

2. In practice, it is usually difficult to obtain accurate parameters (the admittance matrix, very detailed topology, etc.) of the power network. Utilizing feedback measurement data can potentially increase the robustness of the controller against model mismatch.

3. Real-time feedback measurements help us closely keep track of the time-varying components that are hard to predict.

Now, let us assume that we want to modify certain time-varying optimization algorithm, whose iteration is given by

$$\begin{aligned} \hat{z}_\tau &= \mathsf{T}(\hat{z}_{\tau-1}; \mathsf{D}_\tau), \\ \hat{x}_\tau &= \Pi \hat{z}_\tau, \end{aligned}$$

to incorporate feedback measurements to solve the time-varying optimal power flow problem (4.5). A general framework is given as follows:

For each $\tau = 1, 2, \ldots, \lfloor T/\Delta \rfloor$,

1. Measure the input variable, the state variable as well as relevant time-varying components at time $t = \tau\Delta$. Let $\check{x}_\tau$ denote the newly measured input variable, $\check{y}_\tau$ denote the newly measured state variable, and D_τ denote the newly collected problem data.

2. Construct $\check{z}_\tau$ by replacing the entries of $\hat{z}_{\tau-1}$ associated with the canonical projection map Π such that $\Pi\check{z}_\tau = \check{x}_\tau$, and keeping other entries unchanged. Compute

$$\begin{aligned} \hat{z}_\tau &= \tilde{\mathsf{T}}\big(\check{z}_\tau; \check{y}_\tau, \mathsf{D}_\tau\big), \\ \hat{x}_\tau &= \Pi \hat{z}_\tau. \end{aligned} \tag{4.6}$$

3. Apply $\hat{x}_\tau$ to the system.

We first note that, in order for the real-time optimal power flow algorithm to work smoothly, the time spent on the measurement and computation procedure should be less than Δ. Secondly, we use a modified version of the operator T, denoted by $\tilde{\mathsf{T}}$, so that computation of the implicit power flow map $\mathcal{F}_\tau$ is replaced by using the measurement data $\check{y}_\tau$. Thirdly, notice that the we use $\check{z}_\tau$ as the starting point for the iteration (4.6), to make sure that the algorithm uses data that are in accordance with the implicit power flow map $\mathcal{F}_\tau$. The quantity $\check{x}_\tau$ can be different from $\hat{x}_{\tau-1}$ because of the following reasons:

1. For a controllable device, there might be some discrepancy between the control command it receives and its actual power injection. The cause of this discrepancy could be, for example, that the settling time of the device is larger than the interval between real-time updates, or that $\mathcal{X}(t)$ is only an approximation of the actual operating region. See [13] for some related discussions.

2. Since the feasible set of input variable $x(t)$ is time-varying, the setpoint $\hat{x}_{\tau-1}$, which satisfies the constraints specified by $\mathcal{X}_{\tau-1}$, may not lie in the feasible set specified by $\mathcal{X}_\tau$. During the period $\big((\tau-1)\Delta, \tau\Delta\big)$, the controllable devices may need to adjust their setpoints so that the hard physical constraints will not be violated.

3. The measurement error is not negligible.

The details of how the controllable devices implement the received control commands and how they adjust their setpoints due to changes in physical conditions are out of the scope of this paper. Here we employ the following assumption:

Assumption 4.1. *For any τ, the difference between the setpoint $\hat{x}_{\tau-1}$ applied at the end of the last iteration and the measured input variable $\check{x}_\tau$ at the beginning of the current iteration is bounded by*

$$-(e_x + \varsigma\upsilon) \le \hat{x}_{\tau-1} - \check{x}_\tau \le e_x + \varsigma\upsilon, \tag{4.7}$$

where Δ is the sampling interval, $e_x \in \mathbb{R}_+^{d_x}$ is a constant vector, $\varsigma > 0$ is a constant scalar, and each entry of $\upsilon \in \mathbb{R}_+^{d_x}$ is defined by

$$\upsilon_i = \sup_\tau \sup_{x \in \mathcal{X}_\tau} \left| x_i - \big(\mathcal{P}_{\mathcal{X}_{\tau+1}}(x)\big)_i \right|, \qquad i = 1, \ldots, d_x.$$

Basically, (4.7) says that the difference $\hat{x}_{\tau-1} - \check{x}_\tau$ is bounded by two terms. The vector e_x gives an upper bound on the discrepancies between the received control commands and the measured value of setpoints that are independent of the change in $\mathcal{X}_\tau$. The vector υ characterizes the maximum rate of change of the feasible region X_τ, and $\varsigma\upsilon$ gives an upper bound on how much the controllable devices adjust their setpoints due to changes in $\mathcal{X}_\tau$.

Finally, we note that the framework given above assumes that the input variable, the state variable and relate problem data can all be measured or collected in real-time. The measurement and collection of these data may require real-time power system state estimation, which we shall not expand as it is out of the scope of this thesis.

In the next two sections we present two real-time optimal power flow algorithms, one based on the regularized proximal primal-dual gradient algorithm, and the other based on the approximate Newton method with the penalty approach.

4.2 A First-Order Real-Time Optimal Power Flow Algorithm

In this section, we introduce a first-order real-time optimal power flow algorithm.

The algorithm is based on the regularized proximal primal-dual gradieng algorithm discussed in Chapter 2, and is designed for operation of a distribution feeder in which the power injections at some of the buses are controllable. A similar real-time optimal power flow algorithm has first been proposed in [35] and generalized in [14, 36], where regularization on the primal variable has also been introduced.

We choose the partition $(\mathsf{P}_x, \mathsf{P}_y)$ as

$$\begin{aligned} x(t) &= (v_0(t), p_1(t), q_1(t), \ldots, p_n(t), q_n(t)), \\ y(t) &= \big(p_0(t), q_0(t), v_1(t), \theta_1(t), \ldots, v_n(t), \theta_n(t), (\ell_{ij}(t))_{(i,j)\in\mathcal{E}}\big). \end{aligned}$$

The slack bus of the distribution network is placed at the substation, which connects the distribution feeder to the transmission network. For simplicity we assume that $v_0(t)$ is fixed at 1 p.u. We assume that for each time t, the constraint on the power injection of the controllable devices connected to bus $i \in \mathcal{N}$ is given by $(p_i(t), q_i(t)) \in \mathcal{X}_i(t)$, where $\mathcal{X}_i(t) \subset \mathbb{R}^2$ is convex and compact and sufficiently simple so that the projection onto $\mathcal{X}_i(t)$ can be easily computed. The set $\mathcal{X}(t)$ is then given by

$$\mathcal{X}(t) = \{1\} \times \prod_{i=1}^{n} \mathcal{X}_i(t). \tag{4.8}$$

The constraint on $(p_0(t), q_0(t))$ is given by a time-varying box constraint

$$\begin{aligned} \underline{p}_0(t) \leq & p_0(t) \leq \overline{p}_0(t), \\ \underline{q}_0(t) \leq & q_0(t) \leq \overline{q}_0(t), \end{aligned} \tag{4.9a}$$

where $\underline{p}_0(t)$, $\overline{p}_0(t)$, $\underline{q}_0(t)$ and $\overline{p}_0(t)$ are known at time t. We assume that there is no explicit cost associated with the power injection at the slack bus, i.e., $c_0 = 0$.

The constraints on voltage magnitudes and squared current magnitudes are still given by

$$\underline{v}_i \leq v_i(t) \leq \overline{v}_i, \quad i \in \mathcal{N}, \tag{4.9b}$$

$$\ell_{ik}(t) \leq \overline{\ell}_{ik}, \quad (i,k) \in \mathcal{E}. \tag{4.9c}$$

By employing the implicit power flow map, the constraints on the state variable $y(t)$ (4.9) can then be written in the form

$$h(\mathcal{F}(x(t),t),t) \leq 0. \tag{4.10}$$

The dimension (number of entries) of h will be denoted by m. Note that h depends on x purely through $\mathcal{F}(x,t)$ because of the choice of the partition $(\mathsf{P}_x, \mathsf{P}_y)$. Moreover, the Jacobian $J_{h,y}(y,t)$ is a constant matrix over $(y,t) \in \mathbb{R}^{d_y} \times [0,T]$. We denote $H := J_{h,y}(y,t)$.

We notice that when the regularized proximal primal-dual gradient algorithm is applied to our problem, it's natural to incorporate measured value of the state variable, which we denote by $\check{y}_\tau$, in the dual update step (2.10b) as

$$\hat{\lambda}_\tau = \mathcal{P}_{\mathbb{R}^m_+}\left[\hat{\lambda}_{\tau-1} + \eta\alpha\left(h_\tau(\check{y}_\tau) - \epsilon\big(\hat{\lambda}_{\tau-1} - \lambda_{\text{prior}}\big)\right)\right].$$

On the other hand, in the primal step (2.10a), we need to compute the Jacobian of $h_\tau(\mathcal{F}_\tau(\check{x}_\tau))$, which is equal to $HJ_{\mathcal{F}_\tau}(\check{x}_\tau)$. The computation of the Jacobian $J_{\mathcal{F}_\tau}(\check{x}_\tau) = J_{\tilde{\mathcal{F}}}(\check{x}_\tau - x^{\mathrm{L}}(\tau\Delta))$ can be a daunting task, as the implicit function theorem gives

$$J_{\tilde{\mathcal{F}}}(x) = -\left[J_{\mathcal{G}^{\mathsf{P}},y}\big(x,\tilde{\mathcal{F}}(x)\big)\right]^{-1} J_{\mathcal{G}^{\mathsf{P}},x}\big(x,\tilde{\mathcal{F}}(x)\big),$$

which involves the inversion of a large matrix $J_{\mathcal{G}^{\mathsf{P}},y}\big(x,\tilde{\mathcal{F}}(x)\big)$. Existing literature has proposed different methods for reducing the computational complexity. Reference [33] employed rectangular representations of the power flow equations and reported improvement in efficiency over traditional methods; [18, 24, 35, 47] employed particular approximations or linearizations that can be computed offline; [47] also proposed an iterative approach that converges fast empirically by exploiting the radial topology of distribution networks; [90] utilized the observation that there will only be a small number of buses or lines whose voltages or currents violate their constraints in practical situations. We shall introduce the method in [90] in the next section. For now, we make the following assumption that there exists a method to produce an approximate Jacobian efficiently for each iteration:

Assumption 4.2. *Let τ be arbitrary. Let $\tilde{\mathcal{F}}$ be the implicit map derived in Proposition 4.1. Define*

$$\mathcal{O}(\delta) = \bigcup_{t\in[0,T]} \left(\delta B_{d_x} + \left(x^*(t) - x^{\mathrm{L}}(t)\right)\right).$$

Then there exist some sufficiently large $\bar{\delta} > 0$, a map $\mathsf{J} : \mathcal{O}(\bar{\delta}) \times \mathbb{R}^{d_y} \times \mathbb{R}^m_+ \to \mathbb{R}^{d_y \times d_y}$ and a function $e_J : \mathbb{R}_{++} \to \mathbb{R}_+$ such that we have

$$\left\| \lambda^T H \left(\mathsf{J}(x, \tilde{\mathcal{F}}(x), \lambda) - J_{\tilde{\mathcal{F}}}(x) \right) \right\| \le e_J(\delta) \|\lambda\| \tag{4.11}$$

for any $x \in \mathcal{O}(\delta)$, any $\lambda \in \mathbb{R}^m_+$ and any $\delta \in (0, \bar{\delta}]$.

Moreover, $J(x, y, \lambda)$ is continuous over (x, y) for each λ, and the map J is computationally inexpensive so that each iteration of (4.12) *can be completed within the interval $(\tau\Delta, (\tau + 1)\Delta)$.*

The first-order real-time optimal power flow algorithm based on the regular proximal primal-dual gradient algorithm (2.10) is then given as follows:

Initialize α, η, ϵ, λ_{prior}, $\hat{\lambda}_0$.

For each $\tau = 1, 2, \ldots, \lfloor T/\Delta \rfloor$,

1. At time $t = \tau\Delta$, measure the input and the state variables as well as the loads $p^{\mathrm{L}}(\tau\Delta)$, $q^{\mathrm{L}}(\tau\Delta)$, and collect data on the cost functions $c_i(\cdot, \cdot, \tau\Delta)$, the sets $\mathcal{X}_i(\tau\Delta)$ and $\underline{p}_0(\tau\Delta)$, $\overline{p}_0(\tau\Delta)$, $\underline{q}_0(\tau\Delta)$, $\overline{q}_0(\tau\Delta)$.
   Let $\check{p}_\tau \in \mathbb{R}^{\mathcal{N}^+}$ and $\check{q}_\tau \in \mathbb{R}^{\mathcal{N}^+}$ denote the newly measured real and reactive power injections, and let $\check{v}_\tau \in \mathbb{R}^{\mathcal{N}}$, $\check{\theta}_\tau \in \mathbb{R}^{\mathcal{N}}$, $\check{\ell}_\tau \in \mathbb{R}^{\mathcal{N}}$ denote the newly measured voltage magnitudes, phase angles and squared current magnitudes, respectively. Let
   $$\begin{aligned} \check{x}_\tau &= (1, \check{p}_{\tau,1}, \check{q}_{\tau,1}, \ldots, \check{p}_{\tau,n}, \check{q}_{\tau,n}), \\ \check{y}_\tau &= \left(\check{p}_{\tau,0}, \check{q}_{\tau,0}, \check{v}_{\tau,1}, \check{\theta}_{\tau,1}, \ldots, \check{v}_{\tau,n}, \check{\theta}_{\tau,n}, (\check{\ell}_{\tau,ij})_{(i,j)\in\mathcal{E}}\right). \end{aligned}$$

2. Compute $\hat{p}_\tau$ and $\hat{q}_\tau$ by
   $$\begin{aligned} \begin{bmatrix} \hat{p}_{\tau,i} \\ \hat{q}_{\tau,i} \end{bmatrix} = \mathcal{P}_{\mathcal{X}_{i,\tau}} \Bigg[\begin{bmatrix} \check{p}_{\tau,i} \\ \check{q}_{\tau,i} \end{bmatrix} - \alpha \Big(\nabla c_{i,\tau}(\check{p}_{\tau,i}, \check{q}_{\tau,i}) \\ + \mathsf{J}_{p_i,q_i}\left(\check{x}_\tau - x^{\mathrm{L}}_\tau, \check{y}_\tau - y^{\mathrm{L}}_\tau, \hat{\lambda}_{\tau-1}\right)^T H^T \hat{\lambda}_{\tau-1} \Big) \Bigg], \quad i \in \mathcal{N}, \end{aligned} \tag{4.12a}$$
   $$\hat{\lambda}_\tau = \mathcal{P}_{\mathbb{R}^m_+} \left[\hat{\lambda}_{\tau-1} + \eta\alpha \left(h_\tau(\check{y}_\tau) - \epsilon\left(\hat{\lambda}_{\tau-1} - \lambda_{\mathrm{prior}}\right) \right) \right], \tag{4.12b}$$

where J_{p_i,q_i} denotes the columns of the matrix-valued map J that correspond to (p_i, q_i), and $x^{\mathrm{L}}_\tau := x^{\mathrm{L}}(\tau\Delta)$, $y^{\mathrm{L}}_\tau := y^{\mathrm{L}}(\tau\Delta)$, $\mathcal{X}_{i,\tau} := \mathcal{X}_i(\tau\Delta)$, $c_{i,\tau}(\cdot,\cdot) := c_i(\cdot,\cdot,\tau\Delta)$.

3. For each $i \in \mathcal{N}$, set the real and reactive power injections at bus i to be $\hat{p}_{\tau,i}$ and $\hat{q}_{\tau,i}$ respectively.

Distributed implementation Suppose for each bus i with controllable devices, there exists a local agent that operates and measures the controllable devices connected to bus i. Apart from the local agents, there exists a central operator that collects the measurement data of the voltage phasors, current magnitudes and relevant uncontrollable loads across the network. There are communication links between the central operator and each of the local agents. Further, suppose that we use a constant matrix J to approximate the Jacobian $J_{\tilde{\mathcal{F}}}$ (the matrix J can be derived, for example, from the linear DistFlow model [65]). Then the first-order real-time optimal power flow algorithm can be implemented in a distributed fashion naturally. Specifically, at time $t = \tau\Delta$, the central operator will broadcast the Lagrange multiplier $\hat{\lambda}_{\tau-1}$, and then update the Lagrange multipliers by (4.12b) from the measured state variable $\check{y}_\tau$. Each local agent stores their corresponding columns of the matrix HJ offline, and upon receiving $\hat{\lambda}_{\tau-1}$, each local agent will carry out the update (4.12a) individually. This distributed implementation has been suggested in [35, 36].

Tracking Performance

Now we study the tracking performance of the first-order real-time optimal power flow algorithm.

We have formulated the time-varying optimal power flow problem as

$$\begin{aligned} \min_x \quad & c\big(x,t\big) \\ \text{s.t.} \quad & x \in \mathcal{X}(t), \\ & h\big(\mathcal{F}(x,t),t\big) \le 0, \end{aligned} \tag{4.13}$$

where $c(x,t)$ is twice continuously differentiable for each fixed $t \in [0,T]$ and $\nabla^2_{xx}c(x,t)$ is continuous over $(x,t) \in \mathbb{R}^{2n+1} \times [0,T]$. Since $\mathcal{F}(\cdot,t)$ and $h(\cdot,t)$ are both smooth functions for each $t \in [0,T]$, the constraint function $h(\mathcal{F}(x,t),t)$ is twice continuously differentiable with respect to x for each $t \in [0,T]$. It can also be checked that $\nabla^2_{xx}h(\mathcal{F}(x,t),t)$ is continuous over (x,t) provided that $p^{\mathrm{L}}(t)$ and $q^{\mathrm{L}}(t)$

are continuous functions over $t \in [0, T]$. We let

$$h(\mathcal{F}(x,t),t) = f^c(x,t) + f^{nc}(x,t)$$

be an arbitrary decomposition of $h(\mathcal{F}(x,t),t)$ such that $f^c(\cdot,t)$ has convex component for each $t \in [0,T]$, and $\nabla^2_{xx} f^c(x,t)$ is continuous over (x,t).

Similarly as in Chapter 2, we assume that for the problem (4.13) there exists a Lipschitz continuous KKT trajectory $z^*(t) = (x^*(t), \lambda^*(t))$ over $t \in [0,T]$. We then define

$$\begin{aligned}
\sigma_\eta &:= \sup_{\substack{t_1,t_2\in[0,T],\\ t_1\neq t_2}} \frac{\|z^*(t_2) - z^*(t_1)\|_\eta}{|t_2 - t_1|} = \operatorname*{ess\,sup}_{t\in[0,T]} \left\| \frac{d}{dt} z^*(t) \right\|_\eta, \\
M_d &:= \sup_{t\in[0,T]} \left\|\lambda^*(t) - \lambda_{\text{prior}}\right\|, \qquad M_\lambda := \sup_{t\in[0,T]} \|\lambda^*(t)\|, \\
M_{nc}(\delta) &:= \sup_{t\in[0,T]} \sup_{\substack{u:\|u\|\leq\delta,\\ x^*(t)+u\in\mathcal{X}(t)}} \left\| D^2_{xx} f^{nc}(x^*(t)+u,t) \right\|, \\
M_c(\delta) &:= \sup_{t\in[0,T]} \sup_{\substack{u:\|u\|\leq\delta,\\ x^*(t)+u\in\mathcal{X}(t)}} \left\| D^2_{xx} f^{c}(x^*(t)+u,t) \right\|, \\
L_f(\delta) &:= \sup_{t\in[0,T]} \sup_{\substack{u:\|u\|\leq\delta,\\ x^*(t)+u\in\mathcal{X}(t)}} \left\| HJ_{\mathcal{F},x}(x^*(t)+u,t) \right\|, \\
D(\delta,\eta) &:= \sqrt{\eta} L_f(\delta) + M_c(\delta) M_\lambda,
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{L}^{nc}(x,\lambda,t) &:= c(x,t) + \lambda^T f^{nc}(x,t), \\
\overline{H}_{\mathcal{L}^{nc}}(u,t) &:= \int_0^1 \nabla^2_{xx} \mathcal{L}^{nc}(x^*(t)+\theta u, \lambda^*(t), t)\, d\theta, \\
\overline{H}_{f_i^c}(u,t) &:= \int_0^1 2(1-\theta) \nabla^2_{xx} f_i^c(x_t^* + \theta u, t)\, d\theta,
\end{aligned}$$

$$\rho^{(\mathrm{P})}(\delta,\alpha,\eta,\epsilon) := \sup_{t\in[0,T]} \sup_{\substack{u:\|u\|\leq\delta,\\ x^*(t)+u\in\mathcal{X}(t)}} \left\| \left(I - \alpha \overline{H}_{\mathcal{L}^{nc}}(u,t)\right)^2 - \alpha(1-\eta\alpha\epsilon) \sum_{i=1}^m \lambda_i^*(t) \overline{H}_{f_i^c}(u,t) \right\|,$$

$$\begin{aligned}
\rho(\delta,\alpha,\eta,\epsilon) := \Bigg[&\max\left\{\rho^{(\mathrm{P})}(\delta,\alpha,\eta,\epsilon), (1-\eta\alpha\epsilon)^2\right\} + \alpha(1-\eta\alpha\epsilon) \frac{\sqrt{\eta}\delta M_{nc}(\delta)}{2} \\
&+ \alpha^2 \left(2 \sup_{t\in[0,T]} \sup_{\substack{u:\|u\|\leq\delta,\\ x^*(t)+u\in\mathcal{X}(t)}} \left\| \eta\epsilon I - \overline{H}_{\mathcal{L}^{nc}}(u,t) \right\| D(\delta,\eta) + D^2(\delta,\eta) \right) \Bigg]^{1/2},
\end{aligned}$$

$$\kappa(\delta,\alpha,\eta,\epsilon) := \max\left\{1, \frac{1-\eta\alpha\epsilon}{\rho(\delta,\alpha,\eta,\epsilon)}, \frac{\sqrt{\eta}\alpha L_f(\delta)}{\rho(\delta,\alpha,\eta,\epsilon)}\right\}.$$

Notice that when taking the supremum over $u : \|u\| \le \delta$ in these definitions, we also restrict u to lie in the set $\mathcal{X}(t) - x^*(t)$, which is different from what we have done in Chapter 2. This is because we use $\check{x}_\tau \in \mathcal{X}_\tau$ instead of $\hat{x}_{\tau-1}$ in the iterations.

For simplicity of analysis, we assume that the measurement noise is negligible when we collect data at the beginning of each iteration.

Lemma 4.1. *Let τ be arbitrary, and we assume that Assumption 4.2 holds. Let $\check{z}_\tau = (\check{x}_\tau, \hat{\lambda}_{\tau-1})$, and suppose*

$$\left\|\check{z}_\tau - z^*_\tau\right\|_\eta \le \delta$$

for some $\delta \in (0, \bar{\delta}]$. Then if $\hat{z}_\tau = (\hat{x}_\tau, \hat{\lambda}_\tau)$ is generated by (4.12)*, we have*

$$\left\|\hat{z}_\tau - z^*_\tau\right\|_\eta \le \rho(\delta, \alpha, \eta, \epsilon) \left\|\check{z}_\tau - z^*_\tau\right\|_\eta + \kappa(\delta, \alpha, \eta, \epsilon)\sqrt{\eta}\alpha\epsilon M_d + \alpha e_J(\delta) \left\|\hat{\lambda}_{\tau-1}\right\|, \tag{4.14}$$

where $\kappa(\delta, \alpha, \eta, \epsilon)$ is upper bounded by $\sqrt{2}$ and satisfies

$$\lim_{\alpha\to 0^+} \kappa(\delta, \alpha, \eta, \epsilon) = 1.$$

Proof. We introduce the auxiliary quantity $\tilde{z}_\tau = (\tilde{x}_\tau, \tilde{\lambda}_\tau)$ by

$$\begin{aligned} \tilde{x}_\tau &= \mathcal{P}_{\mathcal{X}_\tau}\left[\check{x}_\tau - \alpha\left(\nabla c_\tau(\check{x}_\tau) + J_{\tilde{\mathcal{F}}}\left(\check{x}_\tau - x^{\mathrm{L}}_\tau\right)^T H^T \hat{\lambda}_{\tau-1}\right)\right], \\ \tilde{\lambda}_\tau &= \mathcal{P}_{\mathbb{R}^m_+}\left[\hat{\lambda}_{\tau-1} + \eta\alpha\left(h_\tau\left(\tilde{F}\left(\check{x}_\tau - x^{\mathrm{L}}_\tau\right) + y^{\mathrm{L}}_\tau\right) - \epsilon\left(\hat{\lambda}_{\tau-1} - \lambda_{\mathrm{prior}}\right)\right)\right]. \end{aligned}$$

It can be seen that $\tilde{\lambda}_\tau = \hat{\lambda}_\tau$ as we assume negligible measurement noise on $\check{y}_\tau$. On the other hand, by employing the same approach as in the proof of Lemma 2.3, we can show that

$$\left\|\tilde{z}_\tau - z^*_\tau\right\|_\eta \le \rho(\delta, \alpha, \eta, \epsilon) \left\|\check{z}_\tau - z^*_\tau\right\|_\eta + \kappa(\delta, \alpha, \eta, \epsilon)\sqrt{\eta}\alpha\epsilon M_d,$$

with $\kappa(\delta, \alpha, \eta, \epsilon)$ satisfying the desired properties. Now we bound the difference between $\tilde{z}_\tau$ and $\hat{z}_\tau$. We have

$$\begin{aligned} &\|\hat{z}_\tau - \tilde{z}_\tau\|_\eta = \|\hat{x}_\tau - \tilde{x}_\tau\| \\ \le\ & \alpha\left\|\hat{\lambda}^T_{\tau-1}\left(HJ_{\tilde{\mathcal{F}}}\left(\check{x}_\tau - x^{\mathrm{L}}_\tau\right) - H\mathsf{J}(\check{x}_\tau - x^{\mathrm{L}}_\tau, \check{y}_\tau - y^{\mathrm{L}}_\tau, \hat{\lambda}_{\tau-1})\right)\right\| \\ \le\ & \alpha e_J(\delta)\left\|\hat{\lambda}_{\tau-1}\right\|, \end{aligned}$$

where we used

$$\left\|\check{x}_\tau - x^{\mathrm{L}}_\tau - \left(x^*_\tau - x^{\mathrm{L}}_\tau\right)\right\| = \left\|\check{x}_\tau - x^*_\tau\right\| \le \left\|\check{z}_\tau - z^*_\tau\right\|_\eta \le \delta$$

to employ the bound (4.11) in Assumption 4.2. The desired inequality then follows from the triangle inequality. ◻

The following theorem then characterizes the tracking performance of the first-order real-time optimal power flow algorithm.

Theorem 4.1. *Let* $(\hat{z}_\tau)_\tau$ *denote the sequence generated by the first-order real-time optimal power flow algorithm* (4.12)*, and suppose Assumptions 4.1 and 4.2 hold. Define*

$$\begin{aligned}\mathscr{E}_1 &:= \sigma_\eta \Delta + \varsigma\|\upsilon\| + \|e_x\|,\\ \mathscr{E}_2 &:= \kappa(\delta,\alpha,\eta,\epsilon)\alpha\sqrt{\eta}\epsilon M_d + \alpha e_J(\delta) M_\lambda,\\ \tilde{\rho}(\delta,\alpha,\eta,\epsilon) &:= \rho(\delta,\alpha,\eta,\epsilon) + \alpha\sqrt{\eta}e_J(\delta).\end{aligned}$$

Further, suppose there exist parameters $\delta \in (0,\bar{\delta}]$*,* $\alpha > 0$*,* $\eta > 0$ *and* $\epsilon > 0$ *such that*

$$(1-\tilde{\rho}(\delta,\alpha,\eta,\epsilon))\,\delta \geq \mathscr{E}_1 + \mathscr{E}_2. \tag{4.15a}$$

If initially we have

$$\left\|\check{z}_1 - z_1^*\right\|_\eta = \left\|\begin{bmatrix}\check{x}_1 - x_1^*\\ \hat{\lambda}_0 - \lambda_1^*\end{bmatrix}\right\|_\eta \leq \delta, \tag{4.15b}$$

then

$$\begin{aligned}\left\|\hat{z}_\tau - z_\tau^*\right\|_\eta \leq{}& \frac{\rho(\delta,\alpha,\eta,\epsilon)\mathscr{E}_1 + \mathscr{E}_2}{1-\tilde{\rho}(\delta,\alpha,\eta,\epsilon)}\\ &+ \tilde{\rho}^\tau(\delta,\alpha,\eta,\epsilon)\left(\left\|\check{z}_1 - z_1^*\right\|_\eta - \frac{\rho(\delta,\alpha,\eta,\epsilon)\mathscr{E}_1/\tilde{\rho}(\delta,\alpha,\eta,\epsilon) + \mathscr{E}_2}{1-\tilde{\rho}(\delta,\alpha,\eta,\epsilon)}\right)\end{aligned} \tag{4.16}$$

for all τ.

Proof. For notational simplicity, we just use ρ to denote $\rho(\delta,\alpha,\eta,\epsilon)$, use $\tilde{\rho}$ to denote $\tilde{\rho}(\delta,\alpha,\eta,\epsilon)$, and use κ to denote $\kappa(\delta,\alpha,\eta,\epsilon)$.

The proof is by induction. By the condition (4.15b), we can apply Lemma 4.1 to get

$$\left\|\hat{z}_1 - z_1^*\right\|_\eta \leq \rho\left\|\check{z}_1 - z_1^*\right\|_\eta + \kappa\sqrt{\eta}\alpha\epsilon M_d + \alpha e_J(\delta)\left\|\hat{\lambda}_0\right\|.$$

We notice that

$$\left\|\hat{\lambda}_0\right\| \leq \left\|\hat{\lambda}_0 - \lambda_1^*\right\| + \left\|\lambda_1^*\right\| \leq \sqrt{\eta}\left\|\check{z}_1 - z_1^*\right\|_\eta + M_\lambda.$$

Therefore

$$\begin{aligned}\left\|\hat{z}_1 - z_1^*\right\|_\eta &\leq \rho\left\|\check{z}_1 - z_1^*\right\|_\eta + \kappa\sqrt{\eta}\alpha\epsilon M_d + \alpha e_J(\delta)\left(\sqrt{\eta}\left\|\check{z}_1 - z_1^*\right\|_\eta + M_\lambda\right)\\ &= \tilde{\rho}\left\|\check{z}_1 - z_1^*\right\|_\eta + \mathscr{E}_2,\end{aligned}$$

which indicates that (4.16) holds for $\tau = 1$. Now suppose that for some τ the inequality (4.16) holds. Then we have

$$\begin{aligned}
&\left\|\check{z}_{\tau+1} - z^*_{\tau+1}\right\|_\eta \le \|\check{z}_{\tau+1} - \hat{z}_\tau\|_\eta + \left\|\hat{z}_\tau - z^*_\tau\right\|_\eta + \left\|z^*_\tau - z^*_{\tau+1}\right\|_\eta \\
&\le \|e_x + \varsigma v\| + \frac{\rho\mathscr{E}_1 + \mathscr{E}_2}{1-\tilde{\rho}} + \tilde{\rho}^\tau \left(\left\|\check{z}_1 - z^*_1\right\|_\eta - \frac{\rho\mathscr{E}_1/\tilde{\rho} + \mathscr{E}_2}{1-\tilde{\rho}}\right) + \sigma_\eta \Delta \\
&\le \mathscr{E}_1 + \frac{\rho\mathscr{E}_1 + \mathscr{E}_2}{1-\tilde{\rho}} + \tilde{\rho}^\tau \left(\left\|\check{z}_1 - z^*_1\right\|_\eta - \frac{\rho\mathscr{E}_1/\tilde{\rho} + \mathscr{E}_2}{1-\tilde{\rho}}\right) \\
&\le \mathscr{E}_1 + \frac{\rho\mathscr{E}_1 + \mathscr{E}_2}{1-\tilde{\rho}} + \tilde{\rho}^\tau \left(\delta - \frac{\rho\mathscr{E}_1/\tilde{\rho} + \mathscr{E}_2}{1-\tilde{\rho}}\right),
\end{aligned}$$

where we used the assumption (4.7) and the condition (4.15b). The condition (4.15a) implies $\tilde{\rho} < 1$ and $\delta > (\mathscr{E}_1 + \mathscr{E}_2)/(1-\tilde{\rho})$, and together with the definition of $\tilde{\rho}$, we can further get $\delta > (\rho\mathscr{E}_1/\tilde{\rho} + \mathscr{E}_2)/(1-\tilde{\rho})$. Therefore

$$\begin{aligned}
\left\|\check{z}_{\tau+1} - z^*_{\tau+1}\right\|_\eta &\le \mathscr{E}_1 + \frac{\rho\mathscr{E}_1 + \mathscr{E}_2}{1-\tilde{\rho}} + \tilde{\rho}\left(\delta - \frac{\rho\mathscr{E}_1/\tilde{\rho} + \mathscr{E}_2}{1-\tilde{\rho}}\right) \\
&= \mathscr{E}_1 + \mathscr{E}_2 + \tilde{\rho}\delta \le \delta,
\end{aligned}$$

where the last step follows from (4.15a). We also have

$$\begin{aligned}
&\left\|\hat{\lambda}_\tau\right\| \le \left\|\hat{\lambda}_\tau - \lambda^*_\tau\right\| + \left\|\lambda^*_\tau\right\| \\
&\le \sqrt{\eta}\left(\frac{\rho\mathscr{E}_1 + \mathscr{E}_2}{1-\tilde{\rho}} + \tilde{\rho}^\tau\left(\left\|\check{z}_1 - z^*_1\right\|_\eta - \frac{\rho\mathscr{E}_1/\tilde{\rho} + \mathscr{E}_2}{1-\tilde{\rho}}\right)\right) + M_\lambda.
\end{aligned}$$

Then by Lemma 4.1, we get

$$\begin{aligned}
&\left\|\hat{z}_{\tau+1} - z^*_{\tau+1}\right\|_\eta \le \rho\left\|\check{z}_{\tau+1} - z^*_{\tau+1}\right\|_\eta + \kappa\sqrt{\eta}\alpha\epsilon M_d + \alpha e_J(\delta)\left\|\hat{\lambda}_\tau\right\| \\
&\le \rho\left(\mathscr{E}_1 + \frac{\rho\mathscr{E}_1 + \mathscr{E}_2}{1-\tilde{\rho}} + \tilde{\rho}^\tau\left(\left\|\check{z}_1 - z^*_1\right\|_\eta - \frac{\rho\mathscr{E}_1/\tilde{\rho} + \mathscr{E}_2}{1-\tilde{\rho}}\right)\right) \\
&\quad + \kappa\sqrt{\eta}\alpha\epsilon M_d + \alpha e_J(\delta) M_\lambda \\
&\quad + \alpha\sqrt{\eta}e_J(\delta)\left(\frac{\rho\mathscr{E}_1 + \mathscr{E}_2}{1-\tilde{\rho}} + \tilde{\rho}^\tau\left(\left\|\check{z}_1 - z^*_1\right\|_\eta - \frac{\rho\mathscr{E}_1/\tilde{\rho} + \mathscr{E}_2}{1-\tilde{\rho}}\right)\right) \\
&\le \rho\mathscr{E}_1 + \tilde{\rho}\cdot\frac{\rho\mathscr{E}_1 + \mathscr{E}_2}{1-\tilde{\rho}} + \mathscr{E}_2 + \tilde{\rho}^{\tau+1}\left(\left\|\check{z}_1 - z^*_1\right\|_\eta - \frac{\rho\mathscr{E}_1/\tilde{\rho} + \mathscr{E}_2}{1-\tilde{\rho}}\right) \\
&= \frac{\rho\mathscr{E}_1 + \mathscr{E}_2}{1-\tilde{\rho}} + \tilde{\rho}^{\tau+1}\left(\left\|\check{z}_1 - z^*_1\right\|_\eta - \frac{\rho\mathscr{E}_1/\tilde{\rho} + \mathscr{E}_2}{1-\tilde{\rho}}\right).
\end{aligned}$$

The tracking error bound (4.16) then follows by induction. □

Compared with the bound (2.24) in Theorem 2.1, the bound (4.16) further takes into account the effect of employing an approximate Jacobian $\mathsf{J}(x, y, \lambda)$ of the implicit

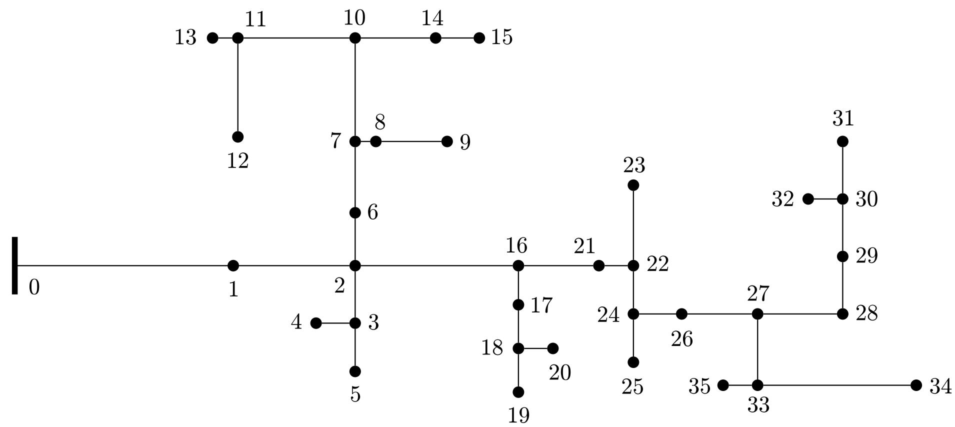

Figure 4.2: Topology of the distribution test feeder.

Bus no.	3	6	9	12	16	19	21	22	25
Normalized area	2	1	1	2	2	2	2	2	2
Rated kVA	200	200	100	200	200	200	200	200	200
Bus no.	27	28	29	30	31	32	33	34	35
Normalized area	1	2	2	2	1	2	2	2	3.5
Rated kVA	200	200	200	200	200	200	200	200	350

Table 4.1: The locations and normalized areas of the PV panels, and the rated apparent power of their inverters.

power flow map as well as the difference between $\hat{x}_{\tau-1}$ and $\check{x}_\tau$ as in Assumption 4.1. It is interesting to see that the error in the approximate Jacobian $e_J(\delta)$ affects the tracking error bound differently from the bound on the difference $\hat{x}_{\tau-1} - \check{x}_\tau$. We can also see that

$$\frac{\rho(\delta, \alpha, \eta, \epsilon)\mathscr{E}_1 + \mathscr{E}_2}{1 - \tilde{\rho}(\delta, \alpha, \eta, \epsilon)} \tag{4.17}$$

gives the eventual tracking error bound.

Numerical Example

In this subsection we present a numerical example of the proposed first-order real-time optimal power flow algorithm applied to a power system test case.

The test case is based on a single-phase version of the IEEE 37 node test feeder with high penetration of photovoltaic (PV) systems. The topology of the network is shown in Figure 4.2. A total of 18 PV systems are installed, and Table 4.1 shows their locations specified by the bus numbers, normalized areas and their inverters'

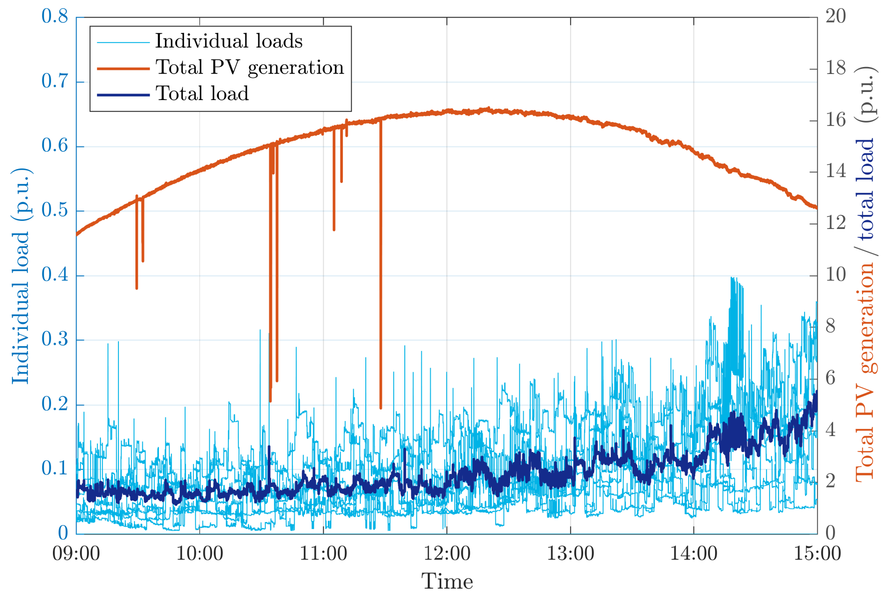

Figure 4.3: Profiles of individual loads $p^{\mathrm{L}}_{\tau,i}, i \in \mathcal{N}$, total load $\sum_{i\in\mathcal{N}} p^{\mathrm{L}}_{\tau,i}$ and total photovoltaic (PV) generation $\sum_{i\in\mathcal{N}_{\mathrm{PV}}} p^{\mathrm{PV}}_{\tau,i}$.

rated apparent power. The sampling interval Δ is 1 sec.

The objective function is given by

$$c_\tau(p,q) = \sum_{i\in\mathcal{N}_{\mathrm{PV}}} c_p\left(p_i - p^{\mathrm{PV}}_{\tau,i}\right)^2 + c_q q_i^2,$$

where $\mathcal{N}_{\mathrm{PV}}$ denotes the set of buses with PV systems installed, and $p^{\mathrm{PV}}_{\tau,i}$ is the maximum real power available for the PV system at bus i at time $t = \tau\Delta$. The term $c_p\left(p_i - p^{\mathrm{PV}}_{\tau,i}\right)^2$ represents the cost for curtailment of the real power generated by the PV system, and $c_q q_i^2$ represents the cost of injecting reactive power q_i by the inverter at bus i. The cost coefficients are $c_p = 3$ and $c_q = 1$. For $i \in \mathcal{N}_{\mathrm{PV}}$, the set $\mathcal{X}_{i,\tau}$ is given by

$$\mathcal{X}_{i,\tau} = \left\{(p,q) \in \mathbb{R}^2 : p^2 + q^2 \le \overline{S}_i^2,\ 0 \le p \le p^{\mathrm{PV}}_{\tau,i}\right\},$$

where $\overline{S}_i$ is the rated apparent power for the inverter at bus i (see Table 4.1). For $i \in \mathcal{N}\backslash\mathcal{N}_{\mathrm{PV}}$ we set $\mathcal{X}_{i,\tau} = \{(0,0)\}$. We require the voltage magnitudes at each bus to be within $[0.95, 1.05]$ p.u., i.e., $\overline{v}_i = 1.05$ and $\underline{v}_i = 0.95$ for each $i \in \mathcal{N}$; they constitute the state variable constraint $h_\tau(\mathcal{F}_\tau(x)) \le 0$.

The load profiles p^{L}_τ are based on real load data measured from feeders in Anatolia, CA during the week of August 2012, and we assume that $q^{\mathrm{L}}_\tau = \frac{1}{2}p^{\mathrm{L}}_\tau$, i.e., a fixed

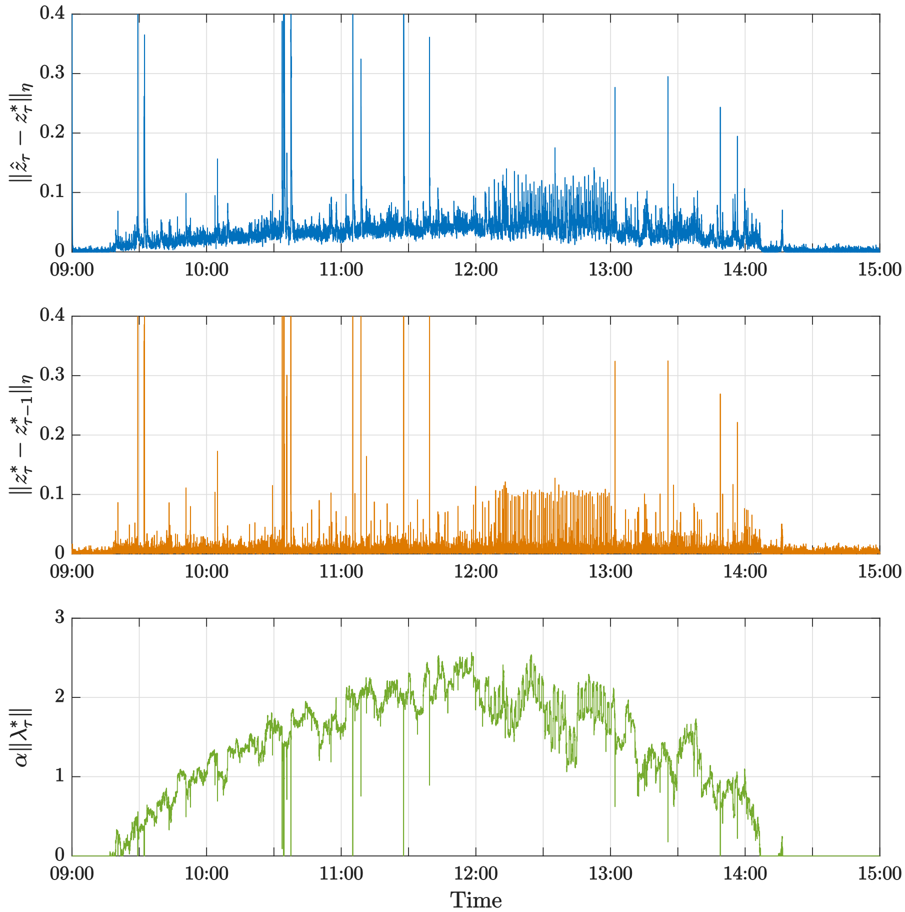

Figure 4.4: Illustrations of $\left\|\hat{z}_\tau - z^*_\tau\right\|_\eta$, $\left\|z^*_\tau - z^*_{\tau-1}\right\|_\eta$ and $\alpha\left\|\lambda^*_\tau\right\|$.

constant power factor of $2/\sqrt{5}$ for each uncontrollable load. The generation profiles of PV systems $\left(p^{\mathrm{PV}}_{\tau,i}\right)_{i\in\mathcal{N}_{\mathrm{PV}}}$ are simulated based on real solar irradiance data scaled by the normalized areas [10]. The time period is from 09:00 to 15:00. The load and generation profiles are shown in Figure 4.3.

We employ the following constant matrix to approximate the Jacobian

$$\mathsf{J}(x, y, \lambda) = J_{\tilde{\mathcal{F}}}(0) = -\left[J_{\mathcal{G}^{\mathsf{P}},y}\left(0, \tilde{\mathcal{F}}(0)\right)\right]^{-1} J_{\mathcal{G}^{\mathsf{P}},x}\left(0, \tilde{\mathcal{F}}(0)\right),$$

i.e., the Jacobian of $\tilde{\mathcal{F}}$ when there is no net power injection at each non-slack bus. The parameters of the algorithm are $\epsilon = 10^{-5}$, $\alpha = 0.04$, $\eta = 0.75/\epsilon$ and $\lambda_{\mathrm{prior}} = 0$. In order to evaluate the tracking performance, an optimal KKT trajectory z^*_τ is computed by the conventional projected primal-dual gradient algorithm in the batch scheme starting from the initial point $(\hat{x}_\tau, \hat{\lambda}_\tau)$ for each τ.

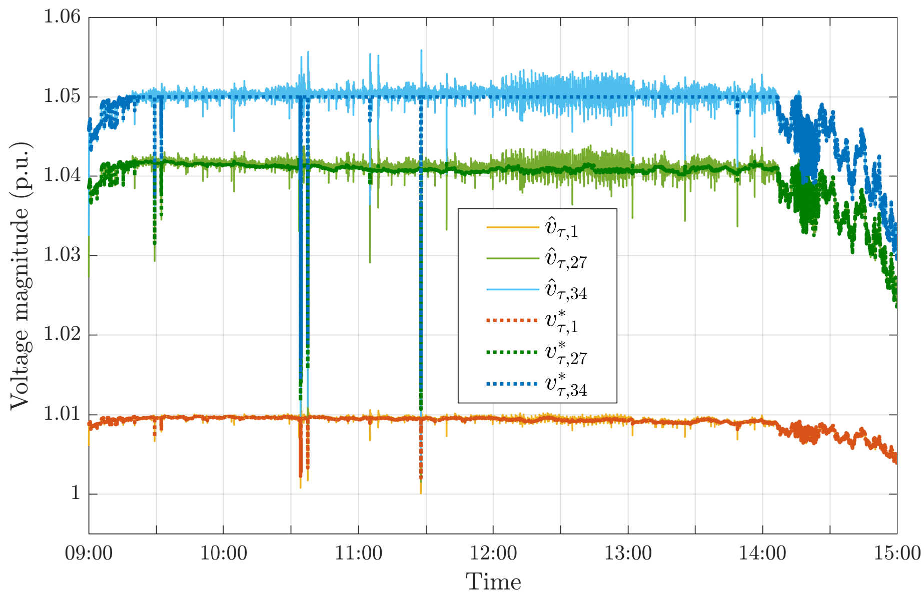

Figure 4.5: The voltage profiles $\hat{v}_{\tau,i}$ and $v^*_{\tau,i}$ for $i = 1$, 27 and 34.

Figure 4.4 shows the resulting tracking error $\left\|\hat{z}_\tau - z^*_\tau\right\|_\eta$, together with $\|z^*_\tau - z^*_{\tau-1}\|_\eta$ and $\alpha\left\|\lambda^*_\tau\right\|$. It can be seen that apart from some spikes, the tracking error is bounded below 0.2 for all τ. We also have the following statistics:

$$\frac{1}{K}\sum_{\tau}\|\hat{z}_\tau - z^*_\tau\|_\eta = 3.26\times 10^{-2}, \qquad \frac{1}{K}\sum_{\tau}\frac{\|\hat{z}_\tau - z^*_\tau\|_\eta}{\|z^*_\tau\|_\eta} = 8.87\times 10^{-3}.$$

As a comparison, we have $K^{-1}\sum_\tau \|z^*_\tau - z^*_{\tau-1}\|_\eta = 8.55\times 10^{-3}$. Moreover, the illustrations seem to suggest that $\left\|\hat{z}_\tau - z^*_\tau\right\|_\eta$ is strongly correlated with $\|z^*_\tau - z^*_{\tau-1}\|_\eta \approx \left\|\frac{d}{dt}z^*(\tau\Delta)\right\|\Delta$ and $\alpha\left\|\lambda^*_\tau\right\|$ in a way similar to the eventual tracking error bound (4.17).

Figure 4.5 compares the voltage magnitudes $\hat{v}_{\tau,i}$ corresponding to $\hat{x}_\tau$ and the voltage magnitudes $v^*_{\tau,i}$ corresponding to x^*_τ for buses $i = 1$, 27 and 34. It can be seen that the constraint on the voltage magnitude of bus 34 is slightly violated for certain time instants. The violation is very small though, as simulation gives

$$\frac{1}{K}\sum_{\tau,i}\left(\left[\hat{v}_{\tau,i} - \overline{v}_i\right]_+ + \left[\underline{v}_i - \hat{v}_{\tau,i}\right]_+\right) = 3.58\times 10^{-4},$$

where $[\cdot]_+$ denotes the positive part of a real number. The violation is partly a consequence of the regularization, which drives the dual variables towards zero and leads to an underestimation of the optimal dual variables.

We emphasize that this numerical example is derived from real-world data, and the theoretical assumptions may not apply here. Nevertheless, the simulations results seem to be in accordance with the theory presented above.

4.3 A Second-Order Real-Time Optimal Power Flow Algorithm

In this section, we introduce a second-order real-time optimal power flow algorithm.

The algorithm is based on the approximate Newton method with the penalty approach discussed in Chapter 3, and is designed for operation of power networks where delays incurred by measurement, communication, etc. are significant and first-order methods do not achieve satisfactory tracking performance. The algorithm can be tailored and applied in both transmission and distribution networks.

Here for the sake of simplicity we employ the same partition as in Section 4.2,

$$x(t) = (v_0(t), p_1(t), q_1(t), \ldots, p_n(t), q_n(t)),$$
$$y(t) = \big(p_0(t), q_0(t), v_1(t), \theta_1(t), \ldots, v_n(t), \theta_n(t), (\ell_{ij}(t))_{(i,j)\in\mathcal{E}}\big),$$

but other partitions can also be considered depending on specific situations. This time we do not fix the voltage magnitude of the slack bus, and assume it can take values between $\underline{v}_0$ and $\overline{v}_0$ just like other buses. We also associate the power injection at the slack bus with a cost function given by $c_0(p_0, q_0, t)$.

Other settings are the same as in Section 4.2. The resulting constraints are given by

$$x(t) \in \mathcal{X}(t) = \big[\underline{v}_0, \overline{v}_0\big] \times \prod_{i=1}^{n} \mathcal{X}_i(t), \tag{4.18}$$

and

$$\underline{p}_0(t) \le p_0(t) \le \overline{p}_0(t), \quad \underline{q}_0(t) \le q_0(t) \le \overline{q}_0(t), \tag{4.19a}$$
$$\underline{v}_i \le v_i(t) \le \overline{v}_i, \qquad i \in \mathcal{N}, \tag{4.19b}$$
$$\ell_{ik}(t) \le \overline{\ell}_{ik}, \qquad (i,k) \in \mathcal{E}. \tag{4.19c}$$

The constraints on the state variable (4.19) can still be written in the form

$$h(\mathcal{F}(x(t), t), t) \le 0.$$

We shall slightly abuse the notation and use $p_0(x,t)$, $q_0(x,t)$, $v_1(x,t)$, $\theta_1(x,t)$, $\ldots$, $v_n(x,t)$, $\theta_n(x,t)$, and $\ell_{ij}(x,t), (i,j) \in \mathcal{E}$ to denote the entries of $\mathcal{F}(x,t)$.

In the approximate Newton method with the penalty approach, we employ penalty functions to handle the constraints on the state variable. Specifically, let

$$
\begin{aligned}
F_\tau^\epsilon(x) := {} & c_0\big(p_0(x,\tau\Delta), q_0(x,\tau\Delta), \tau\Delta\big) + \sum_{i\in\mathcal{N}} c_i\big(p_i, q_i, \tau\Delta\big) \\
& + \sum_{i\in\mathcal{N}} \frac{1}{\kappa\epsilon_{v_i}} \Big(\big[v_i(x,\tau\Delta) - \overline{v}_i\big]_+^\kappa + \big[\underline{v}_i - v_i(x,\tau\Delta)\big]_+^\kappa \Big) \\
& + \sum_{(i,j)\in\mathcal{E}} \frac{1}{\kappa\epsilon_{\ell_{ij}}} \big[\ell_{ij}(x,\tau\Delta) - \overline{\ell}_{ij}\big]_+^\kappa \\
& + \frac{1}{\kappa\epsilon_{p_0}} \Big(\Big[p_0(x,\tau\Delta) - \overline{p}_0(\tau\Delta)\Big]_+^\kappa + \Big[\underline{p}_0(\tau\Delta) - p_0(x,\tau\Delta)\Big]_+^\kappa \Big) \\
& + \frac{1}{\kappa\epsilon_{q_0}} \Big(\Big[q_0(x,\tau\Delta) - \overline{q}_0(\tau\Delta)\Big]_+^\kappa + \Big[\underline{q}_0(\tau\Delta) - q_0(x,\tau\Delta)\Big]_+^\kappa \Big),
\end{aligned}
$$

where $\kappa > 2$, and ϵ_{v_i}, $\epsilon_{\ell_{ij}}$, ϵ_{p_0}, ϵ_{q_0} are positive constants. We apply the approximate Newton method to the following penalized version of the time-varying optimal power flow problem:

$$
\begin{aligned}
\min_x \quad & F_\tau^\epsilon(x) \\
\text{s.t.} \quad & (p_i, q_i) \in \mathcal{X}_i(t), \\
& v_0 \in \big[\underline{v}_0, \overline{v}_0\big].
\end{aligned} \tag{4.20}
$$

The procedure of the second-order real-time optimal power flow algorithm is then given as follows:

For each $\tau = 1, 2, \ldots, \lfloor T/\Delta \rfloor$,

1. At time $t = \tau\Delta$, measure the current input and state variables as well as the loads $p^{\mathrm{L}}(\tau\Delta)$, $q^{\mathrm{L}}(\tau\Delta)$, and collect data on the cost functions $c_i(\cdot,\cdot,\tau\Delta)$, the sets $\mathcal{X}_i(\tau\Delta)$ and $\underline{p}_0(\tau\Delta)$, $\overline{p}_0(\tau\Delta)$, $\underline{q}_0(\tau\Delta)$, $\overline{q}_0(\tau\Delta)$.

   Denote the newly measured input variable by $\check{x}_\tau$.

2. Utilize the new measurement and problem data to compute the gradient

   $$g_\tau := \nabla F_\tau^\epsilon(\check{x}_\tau),$$

   as well as other quantities that will be used for Hessian estimation.

3. Let $\hat{x}_\tau$ be an approximate solution to the following problem:

   $$
   \begin{aligned}
   \min_x \quad & g_\tau^T(x - \check{x}_\tau) + \frac{1}{2}(x - \check{x}_\tau)^T B_\tau (x - \check{x}_\tau) \\
   \text{s.t.} \quad & (p_i, q_i) \in \mathcal{X}_i(\tau\Delta), \quad i \in \mathcal{N} \\
   & v_0 \in \big[\underline{v}_0, \overline{v}_0\big].
   \end{aligned} \tag{4.21}
   $$

Here the matrix B_τ is a positive definite matrix which serves as an estimate of the Hessian of $\mathcal{F}_\tau^\epsilon$.

4. Change the setpoints of the devices in the network according to $\hat{x}_\tau$.
5. Update the Hessian estimate.

Just as explained in Chapter 3, we do not specify the method of producing the positive definite matrix B_τ, and different implementations of this algorithm can use different methods and result in different tracking performance and efficiency. In the following, we shall first study the tracking performance of the second-order real-time optimal power flow algorithm, and then discuss specific implementation details and propose a distributed algorithm for solving the quadratic approximation (4.21).

Tracking Performance

In this subsection, we analyze the tracking performance of the second-order real-time optimal power flow algorithm. We pick up an arbitrary trajectory of local optimal solutions to the penalized problem (4.20), and denote it by $\left(x_\tau^{*,\mathrm{p}}\right)_\tau$. We define

$$\sigma_W := \sup_\tau \left\| x_\tau^{*,\mathrm{p}} - x_{\tau-1}^{*,\mathrm{p}} \right\|_W ,$$

where $W \in \mathbb{R}^{d_x \times d_x}$ is any positive definite matrix. For simplicity we assume that measurement noise is negligible when we collect data at the beginning of each iteration.

Theorem 4.2. *Let $(\hat{x}_\tau)_\tau$ be the sequence generated by the second-order real-time optimal power flow algorithm. Let $W \in \mathbb{R}^{d_x \times d_x}$ be a positive definite matrix, and let $|W|$ denote the matrix obtained by taking the absolute values of all the entries of W. Assume that Assumption 4.1 holds, and for the approximate solution $\tilde{x}_\tau$, we assume that*

$$\|\hat{x}_\tau - \tilde{x}_\tau\|_W \le e_{\mathrm{sol}}$$

for each τ for some $e_{\mathrm{sol}} > 0$, where $\tilde{x}_\tau$ denotes the exact solution to (4.21). *We also define*

$$e_{x,W} := \sqrt{e_x^T |W| e_x}, \qquad \upsilon_W := \sqrt{\upsilon^T |W| \upsilon},$$

and

$$\lambda_M := \sup_\tau \inf\{\lambda \in \mathbb{R} : \lambda W \succeq B_\tau\},$$

$$\lambda_m := \inf_\tau \sup\{\lambda \in \mathbb{R} : \lambda W \preceq B_\tau\}.$$

If for each τ, we have

$$\rho := \sup_{\tau} \frac{\left\| B_{\tau}^{-1}\left(\nabla c_{\tau}(\check{x}_{\tau}) - \nabla c_{\tau}(x_{\tau}^{*,\mathrm{p}})\right) - \left(\check{x}_{\tau} - x_{\tau}^{*,\mathrm{p}}\right)\right\|_{B_{\tau}}}{\left\|\check{x}_{\tau} - x_{\tau}^{*,\mathrm{p}}\right\|_{B_{\tau}}} < \sqrt{\frac{\lambda_m}{\lambda_M}}, \tag{4.22}$$

then

$$\begin{aligned} \left\|\hat{x}_{\tau} - x_{\tau}^{*,\mathrm{p}}\right\|_{W} \le & \frac{\rho\sqrt{\lambda_M/\lambda_m}}{1-\rho\sqrt{\lambda_M/\lambda_m}}\left(\sigma_W + \varsigma\upsilon_W + e_{x,W}\right) + \frac{e_{\mathrm{sol}}}{1-\rho\sqrt{\lambda_M/\lambda_m}} \\ & + \left(\rho\sqrt{\frac{\lambda_M}{\lambda_m}}\right)^{\tau}\left(\left\|\check{x}_1 - x_1^{*,\mathrm{p}}\right\|_W - \frac{\sigma_W + \varsigma\upsilon_W + e_{x,W} + e_{\mathrm{sol}}}{1-\rho\sqrt{\lambda_M/\lambda_m}}\right) \end{aligned} \tag{4.23}$$

bounds the tracking error for each τ.

Proof. We first observe that, Lemma 3.1 can be applied to get the following inequality for each τ

$$\left\|\tilde{x}_{\tau} - x_{\tau}^{*,\mathrm{p}}\right\|_{B_{\tau}} \le \rho\left\|\check{x}_{\tau} - x_{\tau}^{*,\mathrm{p}}\right\|_{B_{\tau}}. \tag{4.24}$$

Then we prove the bound (4.23) by induction. For the initial time step,

$$\begin{aligned} \left\|\hat{x}_1 - x_1^{*,\mathrm{p}}\right\|_W \le & \left\|\hat{x}_1 - \tilde{x}_1\right\|_W + \left\|\tilde{x}_1 - x_1^{*,\mathrm{p}}\right\|_W \le e_{\mathrm{sol}} + \sqrt{\lambda_m^{-1}}\,\rho\left\|\check{x}_1 - x_1^{*,\mathrm{p}}\right\|_{B_1} \\ \le & \, e_{\mathrm{sol}} + \rho\sqrt{\frac{\lambda_M}{\lambda_m}}\left\|\check{x}_1 - x_1^{*,\mathrm{p}}\right\|_W, \end{aligned}$$

showing that (4.23) holds for $\tau = 1$.

Now suppose for some τ the tracking error bound (4.23) holds. Then

$$\begin{aligned} & \left\|\hat{x}_{\tau+1} - x_{\tau+1}^{*,\mathrm{p}}\right\|_W \\ \le & \left\|\hat{x}_{\tau+1} - \tilde{x}_{\tau+1}\right\|_W + \left\|\tilde{x}_{\tau+1} - x_{\tau+1}^{*,\mathrm{p}}\right\|_W \le e_{\mathrm{sol}} + \sqrt{\lambda_m^{-1}}\left\|\tilde{x}_{\tau+1} - x_{\tau+1}^{*,\mathrm{p}}\right\|_{B_{\tau+1}} \\ \le & \, e_{\mathrm{sol}} + \rho\sqrt{\lambda_m^{-1}}\left\|\check{x}_{\tau+1} - x_{\tau+1}^{*,\mathrm{p}}\right\|_{B_{\tau+1}} \le e_{\mathrm{sol}} + \rho\sqrt{\frac{\lambda_M}{\lambda_m}}\left\|\check{x}_{\tau+1} - x_{\tau+1}^{*,\mathrm{p}}\right\|_W \\ \le & \, e_{\mathrm{sol}} + \rho\sqrt{\frac{\lambda_M}{\lambda_m}}\left(\left\|\check{x}_{\tau+1} - \hat{x}_{\tau}\right\|_W + \left\|\hat{x}_{\tau} - x_{\tau}^{*,\mathrm{p}}\right\|_W + \sigma_W\right), \end{aligned} \tag{4.25}$$

where we used (4.24). Now by (4.7), we can see that

$$\begin{aligned} & \left\|\check{x}_{\tau+1} - \hat{x}_{\tau}\right\|_W \\ \le & \sup\{\|u\|_W : -(e_x + \varsigma\upsilon) \le u \le e_x + \varsigma\upsilon\} \\ = & \sup\{\|u_1 + u_2\|_W : -e_x \le u_1 \le e_x, -\varsigma\upsilon \le u_2 \le \varsigma\upsilon\} \\ \le & \sup\{\|u_1\|_W + \|u_2\|_W : -e_x \le u_1 \le e_x, -\varsigma\upsilon \le u_2 \le \varsigma\upsilon\} \\ = & \sup\{\|u\|_W : -e_x \le u \le e_x\} + \sup\{\|u\|_W : -\varsigma\upsilon \le u \le \varsigma\upsilon\}. \end{aligned}$$

We notice that

$$\sup\{\|u\|_W : -e_x \le u \le e_x\} = \sup\left\{\sqrt{\sum_{i,j} u_i W_{ij} u_j} : -e_x \le u \le e_x\right\}$$
$$\le \sup\left\{\sqrt{\sum_{i,j} u_i |W_{ij}| u_j} : -e_x \le u \le e_x\right\} = e_{x,W},$$

and similarly,

$$\sup\{\|u\|_W : -\varsigma\upsilon \le u \le \varsigma\upsilon\} = \varsigma\upsilon_W.$$

Therefore

$$\|\check{x}_{\tau+1} - \hat{x}_\tau\|_W \le e_{x,W} + \varsigma\upsilon_W. \tag{4.26}$$

Now by plugging (4.26), (4.23) into (4.25), we can show that

$$\begin{aligned}\left\|\hat{x}_{\tau+1} - x^{*,\mathrm{p}}_{\tau+1}\right\|_W \le{}& \frac{\rho\sqrt{\lambda_M/\lambda_m}}{1-\rho\sqrt{\lambda_M/\lambda_m}}\left(\sigma_W + \varsigma\upsilon_W + e_{x,W}\right) + \frac{e_{\mathrm{sol}}}{1-\rho\sqrt{\lambda_M/\lambda_m}} \\ &+ \left(\rho\sqrt{\frac{\lambda_M}{\lambda_m}}\right)^{\tau+1}\left(\left\|\check{x}_1 - x^{*,\mathrm{p}}_1\right\|_W - \frac{\sigma_W + \varsigma\upsilon_W + e_{x,W} + e_{\mathrm{sol}}}{1-\rho\sqrt{\lambda_M/\lambda_m}}\right).\end{aligned}$$

By induction we see that (4.23) holds for all τ. □

Remark 4.3. The tracking error bound in Theorem 4.2 is with respect to an optimal trajectory of the penalized problem (4.20). One can further obtain tracking error bounds with respect to $(x^*_\tau)_\tau$, an optimal trajectory of the original problem, if the difference $\|x^*_\tau - x^{*,\mathrm{p}}_\tau\|_W$ can be upper bounded. ■

Computation of the Gradient

In order to compute the gradient of F^ϵ_τ, we need to find the partial derivatives of $v_i(\cdot, \tau\Delta)$, $\ell_{ij}(\cdot, \tau\Delta)$, $p_0(\cdot, \tau\Delta)$, and $q_0(\cdot, \tau\Delta)$ with respect to the entries of the input variable. As mentioned before, this leads to inverting the Jacobian of the power flow equations, and although the Jacobian is in most cases sparse, its inverse is not and the computation will in general be very time consuming as the number of buses becomes large.

On the other hand, we note that in practical situations, there will only be a small number of buses or lines whose voltages or currents violate their constraints at each time instant. Since the derivative of $[x]^\kappa_+$ with respect to x is exactly zero when $x \le 0$, we only need to find $\nabla_x v_i(\check{x}_\tau, \tau\Delta)$ and $\nabla_x \ell_{ij}(\check{x}_\tau, \tau\Delta)$ that correspond to voltages and currents that have violated their constraints. They will involve only a small fraction of the inverted Jacobian matrix, which could make the computation much faster.

From now on until the end of this subsection, we consider a fixed time instant τ and omit the time indices of variables temporarily. We write the power flow equation (4.1) in the form

$$
\begin{aligned}
p_0 &= f_0^p(v_0, v_{1:n}, \theta_{1:n}), \quad q_0 = f_0^q(v_0, v_{1:n}, \theta_{1:n}),\\
p_i &= f_i^p(v_0, v_{1:n}, \theta_{1:n}), \quad q_i = f_i^q(v_0, v_{1:n}, \theta_{1:n}), \ i \in N,
\end{aligned}
$$

where we use the subscript $1:n$ to denote the collection of entries with indices in $\{1, \ldots, n\}$. The partial derivatives of the functions f_{p_0}, f_{q_0}, f_{p_i} and f_{q_i} are given, for example, by [12, Equations (10.40) and (10.41)], and their computation is purely arithmetic as long as the measurement data $\check{v}_i$, $\check{\theta}_i$, $\check{p}_i$ and $\check{q}_i$ are available for use. By viewing p_0,q_0 and $v_i, \theta_i,\ i \in \mathcal{N}$ as functions of v_0, $p_{1:n}$ and $q_{1:n}$ and taking the derivatives of the above equations, we get

$$
\begin{bmatrix} \frac{\partial f_{1:n}^p}{\partial v_{1:n}} & \frac{\partial f_{1:n}^p}{\partial \theta_{1:n}} \\ \frac{\partial f_{1:n}^q}{\partial v_{1:n}} & \frac{\partial f_{1:n}^q}{\partial \theta_{1:n}} \end{bmatrix} \begin{bmatrix} \frac{\partial v_{1:n}}{\partial p_{1:n}} & \frac{\partial v_{1:n}}{\partial q_{1:n}} \\ \frac{\partial \theta_{1:n}}{\partial p_{1:n}} & \frac{\partial \theta_{1:n}}{\partial q_{1:n}} \end{bmatrix} = I_{2n}, \tag{4.27a}
$$

$$
\begin{bmatrix} \frac{\partial f_{1:n}^p}{\partial v_{1:n}} & \frac{\partial f_{1:n}^p}{\partial \theta_{1:n}} \\ \frac{\partial f_{1:n}^q}{\partial v_{1:n}} & \frac{\partial f_{1:n}^q}{\partial \theta_{1:n}} \end{bmatrix} \begin{bmatrix} \frac{\partial v_{1:n}}{\partial v_0} \\ \frac{\partial \theta_{1:n}}{\partial v_0} \end{bmatrix} = \begin{bmatrix} -\frac{\partial f_{1:n}^p}{\partial v_0} \\ -\frac{\partial f_{1:n}^q}{\partial v_0} \end{bmatrix}, \tag{4.27b}
$$

$$
\begin{bmatrix} \frac{\partial p_0}{\partial p_{1:n}} & \frac{\partial p_0}{\partial q_{1:n}} \\ \frac{\partial q_0}{\partial p_{1:n}} & \frac{\partial q_0}{\partial q_{1:n}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_0^p}{\partial v_{1:n}} & \frac{\partial f_0^p}{\partial \theta_{1:n}} \\ \frac{\partial f_0^q}{\partial v_{1:n}} & \frac{\partial f_0^q}{\partial \theta_{1:n}} \end{bmatrix} \begin{bmatrix} \frac{\partial v_{1:n}}{\partial p_{1:n}} & \frac{\partial v_{1:n}}{\partial q_{1:n}} \\ \frac{\partial \theta_{1:n}}{\partial p_{1:n}} & \frac{\partial \theta_{1:n}}{\partial q_{1:n}} \end{bmatrix}, \tag{4.27c}
$$

$$
\begin{bmatrix} \frac{\partial p_0}{\partial v_0} \\ \frac{\partial q_0}{\partial v_0} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_0^p}{\partial v_0} \\ \frac{\partial f_0^q}{\partial v_0} \end{bmatrix} + \begin{bmatrix} \frac{\partial f_0^p}{\partial v_{1:n}} & \frac{\partial f_0^p}{\partial \theta_{1:n}} \\ \frac{\partial f_0^q}{\partial v_{1:n}} & \frac{\partial f_0^q}{\partial \theta_{1:n}} \end{bmatrix} \begin{bmatrix} \frac{\partial v_{1:n}}{\partial v_0} \\ \frac{\partial \theta_{1:n}}{\partial v_0} \end{bmatrix}, \tag{4.27d}
$$

where we use the notation

$$
\frac{\partial a_{1:n}}{\partial b} = \left[\frac{\partial a_i}{\partial b}\right]_{i=1,\ldots,n} \in \mathbb{R}^{n\times 1}, \qquad \frac{\partial a}{\partial b_{1:n}} = \left[\frac{\partial a}{\partial b_j}\right]_{j=1,\ldots,n} \in \mathbb{R}^{1\times n},
$$

$$
\frac{\partial a_{1:n}}{\partial b_{1:n}} = \left[\frac{\partial a_i}{\partial b_j}\right]_{\substack{i=1,\ldots,n\\ j=1,\ldots,n}} \in \mathbb{R}^{n\times n}.
$$

First we note from (4.27a) that

$$\begin{bmatrix} \frac{\partial v_{1:n}}{\partial p_{1:n}} & \frac{\partial v_{1:n}}{\partial q_{1:n}} \\ \frac{\partial \theta_{1:n}}{\partial p_{1:n}} & \frac{\partial \theta_{1:n}}{\partial q_{1:n}} \end{bmatrix} \begin{bmatrix} \frac{\partial f^p_{1:n}}{\partial v_{1:n}} & \frac{\partial f^p_{1:n}}{\partial \theta_{1:n}} \\ \frac{\partial f^q_{1:n}}{\partial v_{1:n}} & \frac{\partial f^q_{1:n}}{\partial \theta_{1:n}} \end{bmatrix} = I_{2n}.$$

Now let

$$\begin{aligned} \mathcal{J} &= \left\{ i \in \mathcal{N} : \check{\ell}_{ij} \geq \overline{\ell}_{ij} \text{ for some } j \in N^+ \right\} \\ &\quad \cup \{ i \in \mathcal{N} : i \text{ is a neighbor of the slack bus} \}, \\ \mathcal{I} &= \mathcal{J} \cup \left\{ i \in \mathcal{N} : \check{v}_i \geq \overline{v}_i \text{ or } \check{v}_i \leq \underline{v}_i \right\}. \end{aligned}$$

Then we have

$$\begin{bmatrix} \frac{\partial v_{\mathcal{I}}}{\partial p_{1:n}} & \frac{\partial v_{\mathcal{I}}}{\partial q_{1:n}} \\ \frac{\partial \theta_{\mathcal{J}}}{\partial p_{1:n}} & \frac{\partial \theta_{\mathcal{J}}}{\partial q_{1:n}} \end{bmatrix} \begin{bmatrix} \frac{\partial f^p_{1:n}}{\partial v_{1:n}} & \frac{\partial f^p_{1:n}}{\partial \theta_{1:n}} \\ \frac{\partial f^q_{1:n}}{\partial v_{1:n}} & \frac{\partial f^q_{1:n}}{\partial \theta_{1:n}} \end{bmatrix} = \begin{bmatrix} I_{\mathcal{I}} & 0 \\ 0 & I_{\mathcal{J}} \end{bmatrix}, \tag{4.28}$$

where $I_{\mathcal{I}}$ is the submatrix formed by the rows of I_n corresponding to $\mathcal{I}$, and similarly for $I_{\mathcal{J}}$. It can be seen that this is a set of $|\mathcal{I}| + |\mathcal{J}|$ linear systems with a common sparse coefficient matrix, and can be solved efficiently when $|\mathcal{I}| + |\mathcal{J}| \ll n$.

After we find $\partial v_{\mathcal{I}}/\partial p_{1:n}$, $\partial v_{\mathcal{I}}/\partial q_{1:n}$, $\partial \theta_{\mathcal{J}}/\partial p_{1:n}$ and $\partial \theta_{\mathcal{J}}/\partial q_{1:n}$, we use (4.27b) to get

$$\begin{aligned} \begin{bmatrix} \frac{\partial v_{1:n}}{\partial v_0} \\ \frac{\partial \theta_{1:n}}{\partial v_0} \end{bmatrix} &= \begin{bmatrix} \frac{\partial f^p_{1:n}}{\partial v_{1:n}} & \frac{\partial f^p_{1:n}}{\partial \theta_{1:n}} \\ \frac{\partial f^q_{1:n}}{\partial v_{1:n}} & \frac{\partial f^q_{1:n}}{\partial \theta_{1:n}} \end{bmatrix}^{-1} \begin{bmatrix} -\frac{\partial f^p_{1:n}}{\partial v_0} \\ -\frac{\partial f^q_{1:n}}{\partial v_0} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial v_{1:n}}{\partial p_{1:n}} & \frac{\partial v_{1:n}}{\partial q_{1:n}} \\ \frac{\partial \theta_{1:n}}{\partial p_{1:n}} & \frac{\partial \theta_{1:n}}{\partial q_{1:n}} \end{bmatrix} \begin{bmatrix} -\frac{\partial f^p_{1:n}}{\partial v_0} \\ -\frac{\partial f^q_{1:n}}{\partial v_0} \end{bmatrix}. \end{aligned}$$

and so

$$\begin{bmatrix} \frac{\partial v_{\mathcal{I}}}{\partial v_0} \\ \frac{\partial \theta_{\mathcal{J}}}{\partial v_0} \end{bmatrix} = \begin{bmatrix} \frac{\partial v_{\mathcal{I}}}{\partial p_{1:n}} & \frac{\partial v_{\mathcal{I}}}{\partial q_{1:n}} \\ \frac{\partial \theta_{\mathcal{J}}}{\partial p_{1:n}} & \frac{\partial \theta_{\mathcal{J}}}{\partial q_{1:n}} \end{bmatrix} \begin{bmatrix} -\frac{\partial f^p_{1:n}}{\partial v_0} \\ -\frac{\partial f^q_{1:n}}{\partial v_0} \end{bmatrix}. \tag{4.29}$$

Then by (4.27c) and (4.27d), and noting that $\partial f^p_0/\partial v_i = \partial f^p_0/\partial \theta_i = \partial f^q_0/\partial v_i = \partial f^q_0/\partial \theta_i = 0$ if i is not a neighbor of the slack bus, we get

$$\begin{bmatrix} \frac{\partial p_0}{\partial p_{1:n}} & \frac{\partial p_0}{\partial q_{1:n}} \\ \frac{\partial q_0}{\partial p_{1:n}} & \frac{\partial q_0}{\partial q_{1:n}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f^p_0}{\partial v_{\mathcal{I}}} & \frac{\partial f^p_0}{\partial \theta_{\mathcal{J}}} \\ \frac{\partial f^q_0}{\partial v_{\mathcal{I}}} & \frac{\partial f^q_0}{\partial \theta_{\mathcal{J}}} \end{bmatrix} \begin{bmatrix} \frac{\partial v_{\mathcal{I}}}{\partial p_{1:n}} & \frac{\partial v_{\mathcal{I}}}{\partial q_{1:n}} \\ \frac{\partial \theta_{\mathcal{J}}}{\partial p_{1:n}} & \frac{\partial \theta_{\mathcal{J}}}{\partial q_{1:n}} \end{bmatrix}, \tag{4.30}$$

and

$$\begin{bmatrix} \frac{\partial p_0}{\partial v_0} \\ \frac{\partial q_0}{\partial v_0} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_0^p}{\partial v_0} \\ \frac{\partial f_0^q}{\partial v_0} \end{bmatrix} + \begin{bmatrix} \frac{\partial f_0^p}{\partial v_{\mathcal{I}}} & \frac{\partial f_0^p}{\partial \theta_{\mathcal{J}}} \\ \frac{\partial f_0^q}{\partial v_{\mathcal{I}}} & \frac{\partial f_0^q}{\partial \theta_{\mathcal{J}}} \end{bmatrix} \begin{bmatrix} \frac{\partial v_{\mathcal{I}}}{\partial v_0} \\ \frac{\partial \theta_{\mathcal{J}}}{\partial v_0} \end{bmatrix}, \tag{4.31}$$

By the definition of $\mathcal{I}$ and $\mathcal{J}$, we can see that (4.28), (4.29), (4.30) and (4.31) give all the partial derivatives for calculating $\nabla_x v_i(\check{x}_\tau, \tau\Delta)$ and $\nabla_x \ell_{ij}(\check{x}_\tau, \tau\Delta)$ that correspond to violated constraints as well as $\nabla_x p_0(\check{x}_\tau, \tau\Delta)$ and $\nabla_x q_0(\check{x}_\tau, \tau\Delta)$.

A Distributed Implementation Based on L-BFGS

In this subsection, we present a distributed implementation for the second-order real-time optimal power flow algorithm, and especially for solving the quadratic approximation (4.21).

We assume that for each bus i with controllable devices, there exists a local agent that is responsible for operating and measuring the controllable devices connected to bus i. Apart from the local agents, there exists a central operator that is responsible for collecting the measurement data of the voltage phasors, current magnitudes and uncontrollable loads across the network. There are communication links between the central operator and each of the local agents.

At the beginning of each time instant $t = \tau\Delta$, the local agents will measure their associated power injections, and report the measurement data back to the central operator. The central operator will collect other measurement data, and then compute the objective value $F_\tau^\epsilon(\check{x}_\tau)$ and the gradient g_τ.

We then employ the L-BFGS method to produce and update the approximate Hessian B_τ, so that B_τ can be represented by

$$B_\tau = \vartheta_\tau I - K_\tau M_\tau K_\tau^T. \tag{4.32}$$

Here $K_\tau \in \mathbb{R}^{n\times 2d}$ and $M_\tau \in \mathbb{R}^{2d\times 2d}$ for some $d \in \mathbb{N}$ which is typically between 3 and 20, and ϑ_τ is a scalar. They are constructed from the

$$Y_\tau := \begin{bmatrix} y_{\tau-d} & \cdots & y_{\tau-1} \end{bmatrix}, \qquad S_\tau := \begin{bmatrix} s_{\tau-d} & \cdots & s_{\tau-1} \end{bmatrix},$$

where

$$s_\tau := \hat{x}_\tau - \check{x}_\tau, \qquad y_\tau := \nabla F_\tau^\epsilon(\hat{x}_\tau) - \nabla F_\tau^\epsilon(\check{x}_\tau).$$

We refer to Section 3.2 for a brief introduction of L-BFGS and to [26, 27] for more details. It can be seen that B_τ is equal to a scaling matrix plus a small rank correction, and because we use the limited-memory approach, computation involving B_τ can be done in a very efficient way even when n is large. On the other hand, the matrix B_τ in general is not sparse, making distributed computation quite difficult. We use the central operator to store and update B_τ and compute its multiplication with vectors.

The central step of the second-order real-time optimal power flow algorithm is to solve (4.21). It can be seen that, while B_τ is in general not sparse, the constraints in (4.21) on the input variable x are separable, suggesting that some computation can be done on local agents in a distributed manner. To be specific, we assume that the feasible region $\mathcal{X}_i(t)$ is represented by

$$\mathcal{X}_i(t) = \left\{ x \in \mathbb{R}^2 : A_i(t)x \le b_i(t),\ \frac{1}{2}x^T Q_i(t)x + w_i(t)^T x + r_i(t) \le 0 \right\}, \tag{4.33}$$

where $A_i(t)$ is a matrix with 2 columns, $b_i(t)$ is a vector, $Q_i(t)$ is a positive semidefinite matrix, $w_i(t) \in \mathbb{R}^2$ and $r_i(t)$ is a scalar. We also assume that $\mathcal{X}_i(t)$ has a nonempty interior[4]. This kind of specification for the feasible region $\mathcal{X}_i(t)$ can cover most of the situations for optimal power flow problems. We also introduce

$$A_0(t) := \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad b_0(t) := \begin{bmatrix} \overline{v}_0 \\ -\underline{v}_0 \end{bmatrix},$$

and $Q_0(t) := 0$, $w_0(t) := 0$, $r_0(t) := -1$.

The problem (4.21) can now be formulated as

$$\begin{aligned} \min_x \quad & g^T(x - \check{x}) + \frac{1}{2}(x - \check{x})^T B(x - \check{x}) \\ \text{s.t.} \quad & A_i x^{(i)} \le b_i, \\ & \frac{1}{2} x^{(i)T} Q_i x^{(i)} + w_i^T x_i^{(i)} + r_i \le 0, i = 0, 1, \dots, n, \end{aligned}$$

where for notational clarity we temporarily drop the time indices, and denote

$$x^{(i)} = (p_i, q_i), \qquad x^{(0)} = v_0.$$

[4] The proposed implementation can be generalized or tailored to the cases where there are multiple convex quadratic constraints or where $\mathcal{X}_i(t)$ has an affine dimension less than 2.

The KKT conditions for this problem are given by

$$
\begin{aligned}
g + B(x - \check{x}) + \left[A_i^T \lambda_i + \nu_i (Q_i x^{(i)} + w_i)\right]_{i=0}^n &= 0, \\
\operatorname{diag}(\lambda_i)(b_i - A_i x^{(i)}) &= \varepsilon_i \mathbf{1}, \\
-\nu_i \left(\frac{1}{2} x^{(i)T} Q_i x^{(i)} + w_i^T x^{(i)} + r_i\right) &= \varepsilon_i, \\
A_i x^{(i)} \le b_i, \qquad \frac{1}{2} x^{(i)T} Q_i x^{(i)} + w_i^T x^{(i)} + r_i &\le 0, \\
\lambda_i \ge 0, \qquad \nu_i &\ge 0,
\end{aligned}
\tag{4.34}
$$

with $\varepsilon_i = 0$ for each $i = 0, \dots, n$. Here $\mathbf{1}$ denotes the vector whose entries are all 1, and we use the notation $[Q_i]_{i=0}^n$ to denote the symmetric matrix whose diagonal blocks are given by Q_i for each $i = 0, 1, \dots, n$.

We solve (4.34) by multiple iterations in an interior-point-like fashion. The Newton step for (4.34) with positive ϵ_i is given by

$$
\begin{aligned}
&\left(B + [\nu Q_i]_{i=0}^n\right) \delta_x + \left[A_i^T \delta_{\lambda_i} + \left(Q_i x^{(i)} + w_i\right) \delta_{\nu_i}\right]_{i=0}^n \\
&= - g - B(x - \check{x}) - \left[A_i^T \lambda_i + \nu_i \left(Q_i x^{(i)} + w_i\right)\right]_{i=0}^n,
\end{aligned}
$$

$$
- \operatorname{diag}(\lambda_i) A_i \delta_{x^{(i)}} + \operatorname{diag}\left(b_i - A_i x^{(i)}\right) \delta_{\lambda_i} = \varepsilon_i \mathbf{1} - \operatorname{diag}\left(b_i - A_i x^{(i)}\right) \lambda_i,
$$

and

$$
\begin{aligned}
&-\nu_i (Q_i x^{(i)} + w_i)^T \delta_{x^{(i)}} - \left(\frac{1}{2} x^{(i)T} Q_i x^{(i)} + w_i^T x^{(i)} + r_i\right) \delta_{\nu_i} \\
&= \varepsilon_i + \left(\frac{1}{2} x^{(i)T} Q_i x^{(i)} + w_i^T x^{(i)} + r_i\right) \nu_i .
\end{aligned}
$$

It can be shown that this set of linear equations for $\left(\delta_x, (\delta_{\lambda_i})_{i=0}^n, (\delta_{\nu_i})_{i=0}^n\right)$ is equivalent to

$$
\begin{aligned}
\left([D_i]_{i=0}^n - K M K^T\right) \delta_x = & - g - \left(\vartheta I - K M K^T\right) (x - \check{x}) \\
& - \left[\varepsilon_i \left(A_i^T G_i^{-1} \mathbf{1} + z_i^{-1} \left(Q_i x^{(i)} + w_i\right)\right)\right]_{i=0}^n,
\end{aligned}
\tag{4.35a}
$$

$$
\delta_{\lambda_i} = G_i^{-1} \left(\varepsilon_i \mathbf{1} + \operatorname{diag}(\lambda_i) A_i \delta_{x^{(i)}}\right) - \lambda_i, \tag{4.35b}
$$

$$
\delta_{\nu_i} = z_i^{-1} \left(\varepsilon_i + \nu_i \left(Q_i x^{(i)} + w_i\right)^T \delta_{x^{(i)}}\right) - \nu_i, \tag{4.35c}
$$

where we denote

$$
G_i = \operatorname{diag}\left(b_i - A_i x^{(i)}\right), \tag{4.36a}
$$

$$
z_i = - \left(\frac{1}{2} x^{(i)T} Q_i x^{(i)} + w_i^T x^{(i)} + r_i\right), \tag{4.36b}
$$

and

$$D_i = \vartheta I + \nu_i Q_i + A_i^T G_i^{-1} \operatorname{diag}(\lambda_i) A_i + z_i^{-1} \nu_i \big(Q_i x^{(i)} + w_i\big)\big(Q_i x^{(i)} + w_i\big)^T, \qquad (4.37)$$

in which we used the L-BFGS representation (4.32). It can be seen that (4.35b) and (4.35c) can be computed in a distributed fashion by local agents after δ_x is solved, and δ_x can be solved by the central operator after it collects enough information from the local agents. Based on these observations, we propose the following iterative method for solving (4.34):

---

1. The central operator construct ϑ, K and M^{-1} by the L-BFGS method [26, 27]. The parameter ϑ is then broadcast to each local agent. Each local agent sets the initial values of $x^{(i)}$, λ_i and ν_i that are strictly feasible, and initializes the barrier parameter ε_i.

2. Each local agent calculates G_i and z_i by (4.36), and then computes

$$\begin{aligned} D_i^{-1} &\leftarrow \left(\vartheta I + \nu_i Q_i + A_i^T G_i^{-1} \operatorname{diag}(\lambda_i) A_i + z_i^{-1} \nu_i \big(Q_i x^{(i)} + w_i\big)\big(Q_i x^{(i)} + w_i\big)^T\right)^{-1}, \\ u_i &\leftarrow -\varepsilon_i D_i^{-1} \left(A_i^T G_i^{-1} \mathbf{1} + z_i^{-1} (Q_i x^{(i)} + w_i)\right), \end{aligned}$$

   and sends $x^{(i)}$, D_i^{-1} and u_i to the central operator.

3. The central operator computes

$$\begin{aligned} \overline{K} &\leftarrow \left[D_i^{-1}\right]_{i=0}^n K, \\ \psi &\leftarrow \left[D_i^{-1}\right]_{i=0}^n \left(-g - \vartheta \left(\left(x^{(i)}\right)_{i=0}^n - \check{x}\right)\right) + \overline{K} M K^T \left(\left(x^{(i)}\right)_{i=0}^n - \check{x}\right), \end{aligned}$$

   and then solves for δ_x by

$$\overline{\psi} \leftarrow \overline{K} \left(M^{-1} - K^T \overline{K}\right)^{-1} K^T \left((u_i)_{i=0}^m + \psi\right), \qquad (4.38a)$$

$$\delta_x \leftarrow (u_i)_{i=0}^n + \psi + \overline{\psi}. \qquad (4.38b)$$

4. The central operator sends $\delta_{x^{(i)}}$ to the i'th local agent for each i.

5. Each local agent i computes

$$\delta_{\lambda_i} \leftarrow G_i^{-1} (\varepsilon_i \mathbf{1} + \operatorname{diag}(\lambda_i) A_i \delta_{x^{(i)}}) - \lambda_i, \qquad (4.39a)$$

$$\delta_{\nu_i} \leftarrow z_i^{-1} \left(\varepsilon_i + \nu_i \big(Q_i x^{(i)} + w_i\big)^T \delta_{x^{(i)}}\right) - \nu_i, \qquad (4.39b)$$

and does the following updates:

$$x^{(i)} \leftarrow x^{(i)} + \alpha_i \delta_{x^{(i)}}, \tag{4.40a}$$

$$\lambda_i \leftarrow \lambda_i + \alpha_i \delta_{\lambda_i}, \tag{4.40b}$$

$$\nu_i \leftarrow \nu_i + \alpha_i \delta_{\nu_i}, \tag{4.40c}$$

where $\alpha_i \in (0, 1]$ is chosen such that the updated $x^{(i)}$, λ_i and ν_i are strictly feasible.

6. Each local agent updates their barrier parameter ε_i.
7. Terminate if some stopping criterion is met; otherwise go to Step 2.

In (4.38) we employ the matrix inversion lemma to efficiently solve for δ_x. Then it's not hard to see that the above procedure is essentially an interior-point algorithm that seeks the solution to the KKT conditions (4.35). We have carefully chosen the variables computed for each step, so that the communication between a local agent and the central operator will only involve a small number of low-dimension vectors and matrices for each iteration. Moreover, the central operator does not need to know anything about $\mathcal{X}_i(t)$. As a result, not only will the communication burden between the central operator and the local agents be relieved, but also the information leakage from the local agents to the central operator could be potentially reduced.

Numerical Example

In this section, we give a numerical example of the proposed distributed implementation of the second-order real-time optimal power flow.

The power network is a modified IEEE 118-bus system. We adopt the original topology, line impedances, and generator data which are taken from MatPower [98]. The time-varying load profiles for each bus are shown in Figure 4.6 which span a period of 6 hours; they are based on the ECO data set [11] and the original load data of the IEEE 118-bus system. The upper and lower bounds of voltage magnitudes in the original IEEE 118-bus system are 1.06 and 0.94 p.u. respectively, but in this simulation we use $\overline{v}_i = 1.045$ and $\underline{v}_i = 0.955$ for all $i \in \mathcal{N}^+$. The corresponding penalty function is then given by

$$\frac{1}{\kappa \epsilon_v} \sum_{i \in \mathcal{N}} \left([v_i(x, \tau\Delta) - 1.045]_+^\kappa + [0.955 - v_i(x, \tau\Delta)]_+^\kappa \right),$$

with $\kappa = 2.5$ and $\epsilon_v = 1.6 \times 10^{-4}$. In other words, we shrink the bounds on the voltage magnitudes to $[0.955, 1.045]$ so that it is more unlikely for the original bounds on voltage magnitudes to be violated.

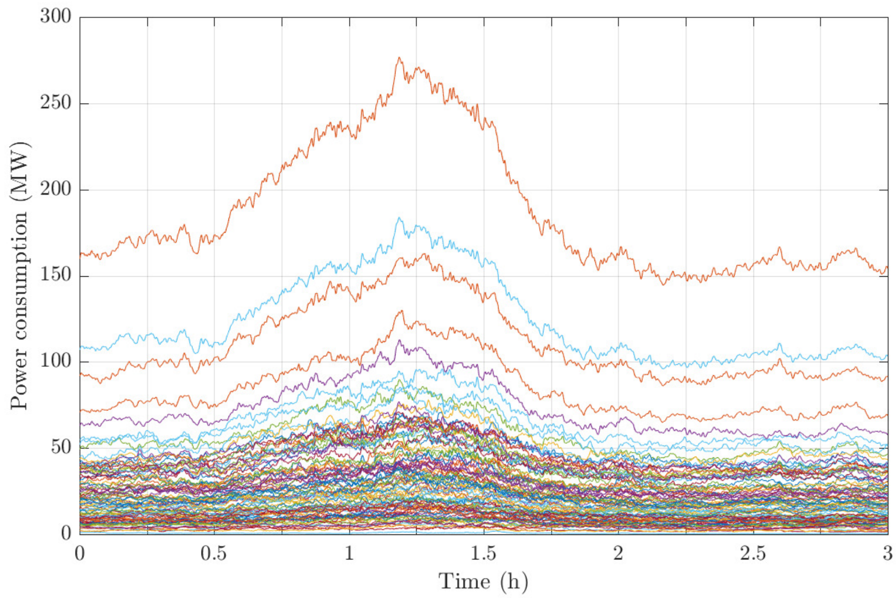

Figure 4.6: Load profiles used for simulation.

At time $t = 0$ we run an offline OPF solver and apply a local optimal solution to the power network as a starting point. Then we run the proposed implementation of the second-order real-time optimal power flow algorithm with a sampling interval $\Delta = 5$ sec. For the distributed L-BFGS algorithm, the number of correction pairs is $d = 8$, and we stop the iterations when

$$\|\delta_x\|_\infty \leq 10^{-5} \quad \text{and} \quad \max_i \max\{\|\delta_{\lambda_i}\|_\infty, |\delta_{\nu_i}|\} \leq 10^{-5}$$
$$\text{and} \quad \max_i \max\{\|G_i\lambda_i\|_\infty, z_i\nu_i\} \leq 10^{-6},$$

or when the total number of iterations has reached 40. We also add a backtracking step before applying $\hat{x}_\tau$ to ensure sufficient decrease in the (penalized) objective value.

In Figure 4.7, we show the curves of the quantities $\left\|\hat{x}_\tau - x_\tau^{*,\mathrm{p}}\right\|$, $\left\|x_\tau^{*,\mathrm{p}} - x_{\tau-1}^{*,\mathrm{p}}\right\|$, $(F_\tau^\epsilon(x_\tau^{*,\mathrm{p}}) - F_\tau^\epsilon(\hat{x}_\tau))/F_\tau^\epsilon(x_\tau^{*,\mathrm{p}})$ and $F_\tau^\epsilon(x_\tau^{*,\mathrm{p}})$, where $x_\tau^{*,\mathrm{p}}$ denotes local optimal solution to the penalized problem. It can be seen that, while the tracking error remains bounded from above and is relatively stable, it is somewhat large compared to the quantity $\left\|x_\tau^{*,\mathrm{p}} - x_{\tau-1}^{*,\mathrm{p}}\right\|$. This seems to suggest that the problem is not very well-conditioned. On the other hand, the relative gap

$$\frac{F_\tau^\epsilon(x_\tau^{*,\mathrm{p}}) - F_\tau^\epsilon(\hat{x}_\tau)}{F_\tau^\epsilon(x_\tau^{*,\mathrm{p}})}$$

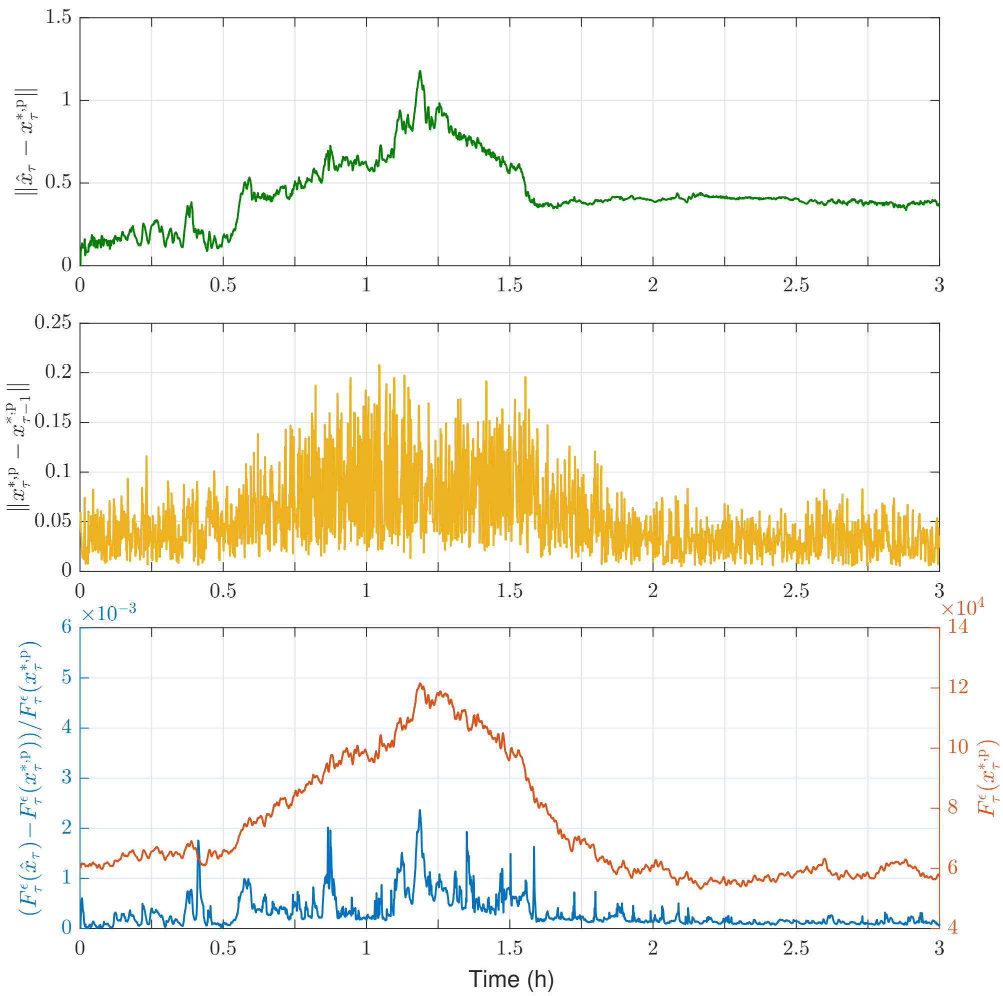

Figure 4.7: Illustrations of $\|\hat{x}_\tau - x_\tau^{*,\mathrm{p}}\|$, $\|x_\tau^{*,\mathrm{p}} - x_{\tau-1}^{*,\mathrm{p}}\|$, $(F_\tau^\epsilon(x_\tau^{*,\mathrm{p}}) - F_\tau^\epsilon(\hat{x}_\tau))/F_\tau^\epsilon(x_\tau^{*,\mathrm{p}})$ and $F_\tau^\epsilon(x_\tau^{*,\mathrm{p}})$.

is very small for the whole period, and its average value turns out to be 3.19×10^{-4}. This suggests that, while the tracking error may seem a bit large, the resulting sequence of setpoints are still good sub-optimal solutions.

Figure 4.8 shows the voltage profiles of the 9 buses whose voltage magnitudes have ever violated the original constraints $0.94 \le v_i \le 1.06$. It can be seen that for most of the time the voltages are within the original bounds; only for the period between 0.75 h and 1.5 h do we see some relatively apparent violations, and those violations still remain at an acceptable level.

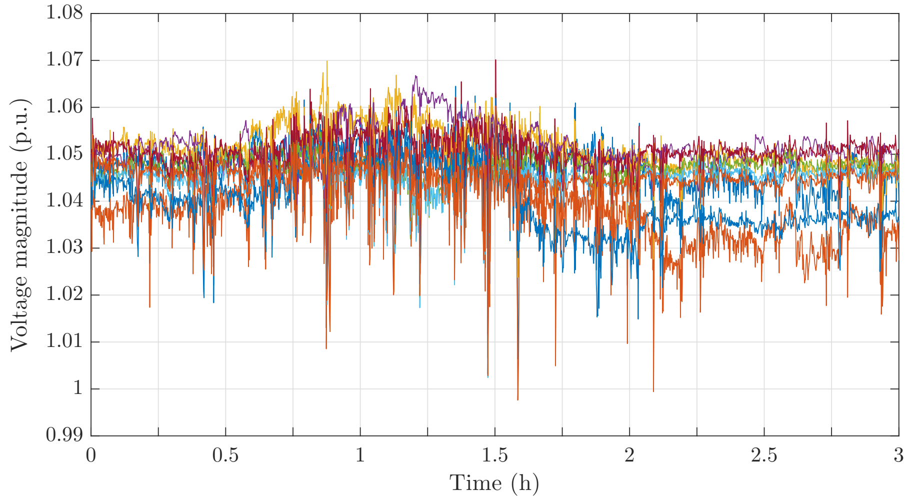

Figure 4.8: Voltage profiles of the buses whose voltages have ever violated the constraints $0.94 \leq v_i \leq 1.06$ for some t.

4.4 Summary

In this chapter, we applied the time-varying optimization algorithms presented in previous chapters to power system operation.

We formulated the time-varying optimal power flow problem, in which the power flow equations are central constraints that describe the underlying physics of power networks. We partitioned the decision variable into two subvectors, one called the input variable and the other called the state variable, and introduced the implicit power flow map derived from the power flow equations that relates the input and state variables. After reformulating the time-varying optimal power flow problem using the implicit power flow map, we discussed how real-time feedback measurement data can be naturally incorporated in the framework of real-time optimal power flow algorithms, so that computation efficiency and robustness against model mismatch can be improved.

We then presented two real-time optimal power flow algorithms in detail. One is based on the regularized proximal primal-dual gradient algorithm. We analyzed its tracking error in the situation where approximate Jacobian of the implicit power flow map is employed for the primal update. The other is based on the approximate Newton method with the penalty approach. We studied how to improve efficiency in computing the gradient vector of the penalized objective function, and proposed a distributed implementation based on the L-BFGS method that produces and stores the approximate Hessian by a compact representation.

Numerical examples were presented for both algorithms. For the first-order real-time optimal power flow algorithm, we tested its performance on a distribution feeder test case, while for the second-order algorithm, we tested its performance on a transmission network test case.

Chapter 5

CONCLUDING REMARKS ON FUTURE DIRECTIONS

In this chapter, we make some concluding remarks on future directions that are worth exploring.

Different metrics for tracking performance Throughout this thesis, the metric for evaluating the tracking performance has been almost exclusively based on

$$e_\tau := \left\|\hat{x}_\tau - x_\tau^*\right\|,$$

i.e., the distance between the solution generated by the time-varying optimization algorithm and the optimal solution it tracks. But there are also other metrics for evaluating the tracking performance as discussed in Section 1.1.

For example, [84] used the fixed-point residual to evaluate the tracking performance of several running algorithms; especially, $c(x, t)$ is not required to be locally strongly convex around $x^*(t)$ for the running projected gradient algorithm to achieve a bounded fixed-point residual. Another example is [48], which considered online learning problems with weakly pseudo-convex loss functions, and derived bounds on the dynamic regret.

These results suggest that, weaker conditions for guaranteed tracking performance may be derived if we use different metrics for tracking performance, and we are interested in whether the results or techniques can be applied to more general time-varying nonconvex problems.

Tracking optimal trajectories that are not Lipschitz continuous In the study of the regularized proximal primal-dual gradient algorithm, we assume that the optimal KKT trajectory $z^*(t)$ is Lipschitz continuous, and the Lipschitz constant plays a crucial role in the tracking error bound. In the study of the approximate Newton method, we also make the assumption that the distance between consecutive optimal points is upper bounded. However, in practice it is not always the case that we can find such a Lipschitz continuous trajectory over the whole period $[0, T]$. Reference [49] discusses situations where a KKT trajectory can emerge, terminate or bifurcate, or several KKT trajectories can merge during the period $(0, T)$, which is not yet covered in our study. There are also cases where the trajectory is only absolutely continuous, or even only of bounded variation with jumps allowed to appear. We are interested in whether reliable methods can be developed to deal with these issues in the time-varying nonconvex setting.

Distributed and asynchronous algorithms in networked systems In this thesis, we have only considered distributed implementation of the real-time optimal power flow algorithms with a particular assumption on the structure of the cyber layer (a central operator with local agents), and we have not yet studied asynchronous algorithms for optimizing a time-varying networked system. References [64, 86, 95] are some representative existing works on distributed time-varying algorithms. Specifically, they proposed different distributed running algorithms for the time-varying consensus optimization problem

$$\min_{x \in \mathcal{X}} \sum_{j=1}^{m} c_{j,\tau}(x),$$

where each local agent is associated with a local time-varying cost function $c_{j,\tau}$ and the local agents are connected by a communication network with an arbitrary topology. In [15], the authors proposed an algorithmic framework for tracking fixed points of time-varying contraction mappings, where only imperfect information of the map is available and communication delays and packet drops lead to asynchronous algorithmic updates.

We are interested in developing and analyzing more general distributed and asynchronous running algorithms for optimizing time-varying networked system, where the communication graph (the cyber layer) can have a general topology.

Better approaches for handling constraints In this thesis, we handle the constraints either by Lagrange multipliers or by penalty functions, and whenever we introduce Lagrange multipliers there will be an accompanying regularization term on the dual variable to ensure that the resulting iterations will have a contraction-like behavior. However, since we essentially modify the original problem to derive the iterations in both approaches, the resulting tracking error bound has a second term which is related to the regularization or penalty coefficient, as can be seen from (2.27), (3.29) or implied by Theorems 3.2 and 3.3.

On the other hand, we notice that a recent work [78] has proved exponential stability of the primal-dual gradient dynamics on the augmented Lagrangian. Specifically, the paper introduced the primal-dual dynamics

$$\begin{aligned}\frac{dx}{dt} &= -\nabla_x L_{\mathrm{aug}}(x,\lambda)\\ \frac{d\lambda}{dt} &= \eta\nabla_\lambda L_{\mathrm{aug}}(x,\lambda),\end{aligned}$$

on the augment Lagrangian L_{aug} defined by

$$L_{\mathrm{aug}}(x,\lambda) := c(x) + \sum_{j=1}^{m} \frac{\left[\rho\left(a_j^T x - b_j\right) + \lambda_j\right]_+^2 - \lambda_j^2}{2\rho},$$

where $c : \mathbb{R}^n \to \mathbb{R}$ is a strongly convex and strongly smooth function, each a_j is a vector and each b_j is a scalar. It has been shown that, under certain conditions, this primal-dual gradient dynamics will converge exponentially to the solution to

$$\begin{aligned}\min_{x} \quad & c(x)\\ \text{s.t.} \quad & a_j^T x \le b_j, \qquad j = 1,\dots,m.\end{aligned}$$

This result suggests that we might be able to obtain linear convergence for the primal-dual gradient method without altering the original problem by dual variable regularization or penalty. It is interesting to see whether the techniques can be used to establish tracking error bounds in the time-varying setting.

Incorporating coupling in the time domain In our formulation of time-varying optimization problems, each time instant is associated with an optimization problem that does not explicitly depend on information from other time instants, and each problem instance can be solved independently without referring to other problem instances. In other words, there is no explicit coupling in the time domain in our formulation, which can be limited in some applications. There are already some pioneering works that consider time domain coupling [30, 34, 37, 61, 62]. For example, [30] considered online convex optimization with switching cost where the cost function over the whole period is given by

$$\sum_{\tau=1}^{K} c_\tau(x_\tau) + \beta\left\|x_\tau - x_{\tau-1}\right\|,$$

and analyzed the competitive ratio of the Averaging Fixed Horizon Control algorithm against the time-varying optimal strategies; [62] considered a similar problem with the overall cost function being

$$\sum_{\tau=1}^{K} c_\tau(x_\tau) + \frac{\beta}{2}\left\|x_\tau - x_{\tau-1}\right\|^2,$$

and analyzed the dynamic regret of the Receding Horizon Gradient Descent algorithm. In these two papers, the source of the time domain coupling is the switching cost. The paper [37] proposed a Newton-type running algorithm for the nonlinear optimal control problem

$$\begin{aligned} \min_{(x_\tau, u_\tau)_{\tau=1}^K} \quad & \sum_{\tau=1}^{K} L_\tau(x_\tau, u_\tau) + Q(x_K) \\ \text{s.t.} \quad & x_\tau = f_\tau(x_{\tau-1}, u_\tau), \end{aligned}$$

for any given initial point x_0, and [34] considered the problem of regulating the output of a linear time-invariant system

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu + B_w w, \\ y_1 &= C_1 x + D_{1w} w, \\ y_2 &= C_2 x + D_{2w} w, \end{aligned}$$

to track the solution to a time-varying constrained optimization problem that optimizes the steady-state trajectory of the system. Here the time domain coupling comes from the underlying dynamical systems. We are interested in generalizing our theories and algorithms to handle time domain coupling.

BIBLIOGRAPHY

[1] J. Abadie and J. Carpentier. Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints. In R. Fletcher, editor, *Optimization*, pages 37–47. Academic Press, London, 1969.

[2] S. Adly, T. Haddad, and L. Thibault. Convex sweeping process in the framework of measure differential inclusions and evolution variational inequalities. *Mathematical Programming*, 148:5–47, 2014.

[3] E. L. Allgower and K. Georg. Numerical path following. In P. G. Ciarlet and J. L. Lions, editors, *Handbook of Numerical Analysis*, volume 5, pages 3–207. Elsevier Science, Amsterdam, 1997.

[4] L. Andrew, S. Barman, K. Ligett, M. Lin, A. Meyerson, A. Roytman, and A. Wierman. A tale of two metrics: Simultaneous bounds on competitiveness and regret. In *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 741–763, Princeton, NJ, USA, June 2013.

[5] A. Araposthatis, S. Sastry, and P. Varaiya. Analysis of power-flow equation. *International Journal of Electrical Power & Energy Systems*, 3(3):115–126, 1981.

[6] M. S. Asif and J. Romberg. Sparse recovery of streaming signals using ℓ_1-homotopy. *IEEE Transactions on Signal Processing*, 62(16):4209–4223, 2014.

[7] J.-P. Aubin and H. Frankowska. *Set-Valued Analysis*. Birkhäuser Boston, 1990.

[8] B. Baingana, P. Traganitis, G. Giannakis, and G. Mateos. Big data analytics for social networks. In I. Pitas, editor, *Graph-Based Social Media Analysis*, pages 293–339. Chapman & Hall/CRC, 2015.

[9] A. Balavoine, C. J. Rozell, and J. Romberg. Discrete and continuous-time soft-thresholding for dynamic signal recovery. *IEEE Transactions on Signal Processing*, 63(12):3165–3176, 2015.

[10] J. Bank and J. Hambrick. Development of a high resolution, real time, distribution-level metering system and associated visualization modeling, and data analysis functions. Technical Report NREL/TP-5500-56610, National Renewable Energy Laboratory, May 2013.

[11] C. Beckel, W. Kleiminger, R. Cicchetti, T. Staake, and S. Santini. The ECO data set and the performance of non-intrusive load monitoring algorithms. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, pages 80–89, Memphis, TN, USA, Nov. 2014.

[12] A. R. Bergen and V. Vittal. *Power Systems Analysis*. Prentice Hall, 2nd edition, 2000.

[13] A. Bernstein, N. J. Bouman, and J.-Y. L. Boudec. Design of resource agents with guaranteed tracking properties for real-time control of electrical grids, 2015, arXiv:1511.08628.

[14] A. Bernstein and E. Dall'Anese. Real-time feedback-based optimization of distribution grids: A unified approach, 2017, arXiv:1711.01627.

[15] A. Bernstein and E. Dall'Anese. Asynchronous and distributed tracking of time-varying fixed points. In *Proceedings of the 57th IEEE Conference on Decision and Control (CDC)*, pages 3237–3243, Miami Beach, FL, USA, Dec. 2018.

[16] A. Bernstein, E. Dall'Anese, and A. Simonetto. Online optimization with feedback, 2018, arXiv:1804.05159.

[17] A. Bernstein, L. Reyes-Chamorro, J.-Y. Le Boudec, and M. Paolone. A composable method for real-time control of active distribution networks with explicit power setpoints. Part I: Framework. *Electric Power Systems Research*, 125:254–264, 2015.

[18] A. Bernstein, C. Wang, E. Dall'Anese, J.-Y. Le Boudec, and C. Zhao. Load flow in multiphase distribution networks: Existence, uniqueness, non-singularity and linear models. *IEEE Transactions on Power Systems*, 33(6):5832–5843, 2018.

[19] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, 1999.

[20] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, Belmont, Massachusetts, 2003.

[21] O. Besbes, Y. Gur, and A. Zeevi. Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244, 2015.

[22] J. Blot. On global implicit functions. *Nonlinear Analysis: Theory, Methods & Applications*, 17(10):947–959, 1991.

[23] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numerica*, 4:1–51, 1995.

[24] S. Bolognani, G. Cavraro, and S. Zampieri. A distributed feedback control approach to the optimal reactive power flow problem. In D. C. Tarraf, editor, *Control of Cyber-Physical Systems*, pages 259–277. Springer International Publishing Switzerland, 2013.

[25] F. Bullo. *Lectures on Network Systems*. CreateSpace, 2018. With contributions by J. Cortés, F. Dörfler, and S. Martinez.

[26] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[27] R. H. Byrd, J. Nocedal, and R. B. Schnabel. Representations of quasi-newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(1-3):129–156, 1994.

[28] C. Castaing, T. X. Dúc Hā, and M. Valadier. Evolution equations governed by the sweeping process. *Set-Valued Analysis*, 1(2):109–139, 1993.

[29] J. Chen and V. K. N. Lau. Convergence analysis of saddle point problems in time varying wireless systems — Control theoretical approach. *IEEE Transactions on Signal Processing*, 60(1):443–452, 2012.

[30] N. Chen, A. Agarwal, A. Wierman, S. Barman, and L. L. H. Andrew. Online convex optimization using predictions. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 191–204, Portland, OR, USA, June 2015.

[31] A. Cherukuri, B. Gharesifard, and J. Cortés. Saddle-point dynamics: conditions for asymptotic stability of saddle points. *SIAM Journal on Control and Optimization*, 55(1):486–511, 2017.

[32] A. Cherukuri, E. Mallada, S. Low, and J. Cortés. The role of convexity in saddle-point dynamics: Lyapunov function and robustness. *IEEE Transactions on Automatic Control*, 63(8):2449–2464, 2018.

[33] K. Christakou, J.-Y. LeBoudec, M. Paolone, and D.-C. Tomozei. Efficient computation of sensitivity coefficients of node voltages and line currents in unbalanced radial electrical distribution networks. *IEEE Transactions on Smart Grid*, 4(2):741–750, 2013.

[34] M. Colombino, E. Dall'Anese, and A. Bernstein. Online optimization as a feedback controller: Stability and tracking, 2018, arXiv:1805.09877.

[35] E. Dall'Anese and A. Simonetto. Optimal power flow pursuit. *IEEE Transactions on Smart Grid*, 9(2):942–952, 2018.

[36] E. Dall'Anese, S. S. Guggilam, A. Simonetto, Y. C. Chen, and S. V. Dhople. Optimal regulation of virtual power plants. *IEEE Transactions on Power Systems*, 33(2):1868–1881, 2018.

[37] M. Diehl, H. G. Bock, and J. P. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on control and optimization*, 43(5):1714–1736, 2005.

[38] R. Divi and H. K. Kesavan. A shifted penalty function approach for optimal load-flow. *IEEE Transactions on Power Apparatus and Systems*, PAS-101(9):3502–3512, 1982.

[39] A. L. Dontchev, M. I. Krastanov, R. T. Rockafellar, and V. M. Veliov. An Euler–Newton continuation method for tracking solution trajectories of parametric variational inequalities. *SIAM Journal on Control and Optimization*, 51(3):1823–1840, 2013.

[40] A. L. Dontchev and R. T. Rockafellar. *Implicit Functions and Solution Mappings*. Springer Science+Business Media New York, 2nd edition, 2014.

[41] K. Dvijotham, S. Low, and M. Chertkov. Convexity of energy-like functions: Theoretical results and applications to power system operations, 2015, arXiv:1501.04052.

[42] K. Dvijotham, S. Low, and M. Chertkov. Solving the power flow equations: A monotone operator approach, 2015, arXiv:1506.08472.

[43] M. Fazlyab, C. Nowzari, G. J. Pappas, A. Ribeiro, and V. M. Preciado. Self-triggered time-varying convex optimization. In *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, pages 3090–3097, Las Vegas, NV, USA, Dec. 2016.

[44] G. B. Folland. *Real Analysis: Modern Techniques and Their Applications*. John Wiley & Sons, 2nd edition, 1999.

[45] S. Frank, I. Steponavice, and S. Rebennack. Optimal power flow: a bibliographic survey I – formulations and deterministic methods. *Energy Systems*, 3(3):221–258, 2012.

[46] S. Frank, I. Steponavice, and S. Rebennack. Optimal power flow: a bibliographic survey II – non-deterministic and hybrid methods. *Energy Systems*, 3(3):259–289, 2012.

[47] L. Gan and S. H. Low. An online gradient algorithm for optimal power flow on radial networks. *IEEE Journal on Selected Areas in Communications*, 34(3):625–638, 2016.

[48] X. Gao, X. Li, and S. Zhang. Online learning with non-convex losses and non-stationary regret. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 235–243, Playa Blanca, Lanzarote, Spain, Apr. 2018.

[49] J. Guddat, F. G. Vazquez, and H. T. Jongen. *Parametric Optimization: Singularities, Pathfollowing and Jumps*. Springer Fachmedien Wiesbaden, 1990.

[50] D. Hajinezhad and M. Hong. Perturbed proximal primal-dual algorithm for nonconvex nonsmooth optimization. [Online] Available at: `http://people.ece.umn.edu/~mhong/PProx_PDA.pdf`, 2017.

[51] E. C. Hall and R. M. Willett. Online convex optimization in dynamic environments. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):647–662, 2015.

[52] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler. Projected gradient descent on riemannian manifolds with applications to online power system optimization. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 225–232, Monticello, IL, USA, Sept. 2016.

[53] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler. Generic existence of unique Lagrange multipliers in AC optimal power flow. *IEEE Control Systems Letters*, 2(4):791–796, 2018.

[54] A. Hauswirth, I. Subotic, S. Bolognani, G. Hug, and F. Dörfler. Time-varying projected dynamical systems with applications to feedback optimization of power systems. In *Proceedings of the 57th IEEE Conference on Decision and Control (CDC)*, pages 3258–3263, Miami Beach, FL, USA, Dec. 2018.

[55] A. Hauswirth, A. Zanardi, S. Bolognani, F. Dörfler, and G. Hug. Online optimization in closed loop on the power flow manifold. In *2017 IEEE Manchester PowerTech*, pages 1–6, Manchester, UK, June 2017.

[56] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan. Online optimization: Competing with dynamic comparators. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 398–406, San Diego, CA, USA, May 2015.

[57] J. Jost. *Postmodern Analysis*. Springer-Verlag Berlin Heidelberg, 3rd edition, 2005.

[58] M. B. Khuzani and N. Li. Distributed regularized primal-dual method: Convergence analysis and trade-offs, 2016, arXiv:1609.08262.

[59] A. J. Korsak. On the question of uniqueness of stable load-flow solutions. *IEEE Transactions on Power Apparatus and Systems*, PAS-91(3):1093–1100, 1972.

[60] J. Koshal, A. Nedić, and U. V. Shanbhag. Multiuser optimization: Distributed algorithms and error analysis. *SIAM Journal on Optimization*, 21(3):1046–1081, 2011.

[61] L. S. P. Lawrence, Z. E. Nelson, E. Mallada, and J. W. Simpson-Porco. Optimal steady-state control for linear time-invariant systems. In *Proceedings of the 57th IEEE Conference on Decision and Control (CDC)*, pages 3251–3257, Miami Beach, FL, USA, Dec. 2018.

[62] Y. Li, G. Qu, and N. Li. Using predictions in online optimization with switching costs: A fast algorithm and a fundamental limit. In *2018 Annual American Control Conference*, pages 3008–3013, Milwaukee, WI, USA, June 2018.

[63] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. *IEEE/ACM Transactions on Networking*, 21(5):1378–1391, 2013.

[64] Q. Ling and A. Ribeiro. Decentralized dynamic optimization through the alternating direction method of multipliers. *IEEE Transactions on Signal Processing*, 62(5):1185–1197, 2014.

[65] S. H. Low. Convex relaxation of optimal power flow — Part I: Formulations and equivalence. *IEEE Transactions on Control of Network Systems*, 1(1):15–27, 2014.

[66] S. H. Low. Convex relaxation of optimal power flow — Part II: Exactness. *IEEE Transactions on Control of Network Systems*, 1(2):177–189, 2014.

[67] S. H. Low and D. E. Lapsley. Optimization flow control — I: Basic algorithm and convergence. *IEEE/ACM Transactions on on Networking*, 7(6):861–874, 1999.

[68] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, pages 7195–7201, Las Vegas, NV, USA, Dec. 2016.

[69] J. L. Morales and J. Nocedal. Remark on "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization". *ACM Transactions on Mathematical Software*, 38(1):7, 2011.

[70] J. R. Munkres. *Topology*. Prentice Hall, 2nd edition, 2000.

[71] A. Nagurney and D. Zhang. *Projected Dynamical Systems and Variational Inequalities with Applications*. Springer Science+Business Media New York, 1996.

[72] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Science+Business Media, 2nd edition, 2006.

[73] J. S. Pang. Solution differentiability and continuation of newton's method for variational inequality problems over polyhedral sets. *Journal of Optimization Theory and Applications*, 66(1):121–135, 1990.

[74] N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[75] R. R. Phelps. *Convex Functions, Monotone Operators and Differentiability*. Springer-Verlag Berlin Heidelberg, 2nd edition, 1993.

[76] R. Plastock. Homeomorphisms between Banach spaces. *Transactions of the American Mathematical Society*, 200:169–183, 1974.

[77] A. Y. Popkov. Gradient methods for nonstationary unconstrained optimization problems. *Automation and Remote Control*, 66(6):883–891, 2005.

[78] G. Qu and N. Li. On the exponential stability of primal-dual gradient dynamics. *IEEE Control Systems Letters*, 3(1):43–48, 2019. Full version available at `https://arxiv.org/abs/1803.01825`.

[79] S. Rahili and W. Ren. Distributed continuous-time convex optimization with time-varying cost functions. *IEEE Transactions on Automatic Control*, 62(4):1590–1605, 2017.

[80] J. Rakowska, R. T. Haftka, and L. T. Watson. An active set algorithm for tracing parametrized optima. *Structural optimization*, 3(1):29–44, 1991.

[81] S. M. Robinson. Strongly regular generalized equations. *Mathematics of Operations Research*, 5(1):43–62, 1980.

[82] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[83] W. Rudin. *Functional Analysis*. McGraw-Hill, 2nd edition, 1991.

[84] A. Simonetto. Time-varying convex optimization via time-varying averaged operators, 2017, arXiv:1704.07338.

[85] A. Simonetto and E. Dall'Anese. Prediction-correction algorithms for time-varying constrained optimization. *IEEE Transactions on Signal Processing*, 65(20):5481–5494, 2017.

[86] A. Simonetto and G. Leus. Distributed asynchronous time-varying constrained optimization. In *Conference Record of the Forty-Eighth Asilomar Conference on Signals, Systems & Computers*, pages 2142–2146, Pacific Grove, CA, USA, Nov. 2014.

[87] A. Simonetto and G. Leus. Double smoothing for time-varying distributed multiuser optimization. In *Proceedings of the 2nd IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 852–856, Atlanta, GA, USA, Dec. 2014.

[88] Y. Tang, E. Dall'Anese, A. Berstein, and S. Low. Running primal-dual gradient method for time-varying nonconvex problems, 2018, arXiv:1812.00613.

[89] Y. Tang, E. Dall'Anese, A. Berstein, and S. H. Low. A feedback-based regularized primal-dual gradient method for time-varying nonconvex optimization. In *Proceedings of the 57th IEEE Conference on Decision and Control (CDC)*, pages 3244–3250, Miami Beach, FL, USA, Dec. 2018.

[90] Y. Tang, K. Dvijotham, and S. Low. Real-time optimal power flow. *IEEE Transactions on Smart Grid*, 8(6):2963–2973, 2017.

[91] Y. Tang and S. Low. Distributed algorithm for time-varying optimal power flow. In *Proceedings of the 56th IEEE Conference on Decision and Control (CDC)*, pages 3264–3270, Melbourne, VIC, Australia, Dec. 2017.

[92] G. Wachsmuth. On LICQ and the uniqueness of Lagrange multipliers. *Operations Research Letters*, 41(1):78–80, 2013.

[93] J. Wang and N. Elia. A control perspective for centralized and distributed convex optimization. In *Proceedings of the 50th IEEE Conference on Decision and Control (CDC)*, pages 3800–3805, Orlando, FL, USA, Dec. 2011.

[94] D. Willett and J. S. W. Wong. On the discrete analogues of some generalizations of Gronwall's inequality. *Monatshefte für Mathematik*, 69(4):362–367, 1965.

[95] C. Xi and U. A. Khan. Distributed dynamic optimization over directed graphs. In *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, pages 245–250, Las Vegas, NV, USA, Dec. 2016.

[96] T. Yang, L. Zhang, R. Jin, and J. Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, USA, June 2016.

[97] V. M. Zavala and M. Anitescu. Real-time nonlinear optimization as a generalized equation. *SIAM Journal on Control and Optimization*, 48(8):5444–5467, 2010.

[98] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, 2011.

[99] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, ICML-2003, pages 928–935, Washington DC, USA, Aug. 2003.