Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Extrapolation-Based Prediction-Correction Methods for Time-varying Convex Optimization

Nicola Bastianello[a,*], Ruggero Carli[b], Andrea Simonetto[c]

[a] School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden
[b] Department of Information Engineering (DEI), University of Padova, Italy
[c] UMA, ENSTA Paris, Institut Polytechnique de Paris, Palaiseau 91120, France

ARTICLE INFO

Article history:
Received 20 December 2022
Revised 4 April 2023
Accepted 29 April 2023
Available online 1 May 2023

Keywords:
Online optimization
Prediction-correction
Operator theory
Graph signal processing

ABSTRACT

In this paper, we focus on the solution of online optimization problems that arise often in signal processing and machine learning, in which we have access to streaming sources of data. We discuss algorithms for online optimization based on the prediction-correction paradigm, both in the primal and dual space. In particular, we leverage the typical regularized least-squares structure appearing in many signal processing problems to propose a novel and tailored prediction strategy, which we call extrapolation-based. By using tools from operator theory, we then analyze the convergence of the proposed methods as applied both to primal and dual problems, deriving an explicit bound for the tracking error, that is, the distance from the time-varying optimal solution. We further discuss the empirical performance of the algorithm when applied to signal processing, machine learning, and robotics problems.

1. Introduction

Continuously varying optimization programs have appeared as a natural extension of time-invariant ones when the cost function, the constraints, or both, depend on a time parameter and change continuously in time. This setting captures relevant problems in the data streaming era, see *e.g.* [13,42] and references therein.

We focus here on linearly constrained regularized least-squares problems of the form

$$\mathsf{P}(t)\!: \boldsymbol{x}^*(t), \boldsymbol{y}^*(t) = \arg\min\nolimits_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{y}\in\mathbb{R}^m} \underbrace{\frac{1}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{d}(t)\|^2 + f_0(\boldsymbol{x})}_{=:f(\boldsymbol{x};t)} + h(\boldsymbol{y}), \tag{1}$$

$$\text{s.t. } \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} = \boldsymbol{c}, \tag{2}$$

where, $t \in \mathbb{R}_+$ is non-negative, continuous, and is used to index time; $f : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is a *smooth strongly convex function* uniformly in time; in addition, $h : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ is a closed convex and proper function, matrices $\boldsymbol{D} \in \mathbb{R}^{q\times n}, \boldsymbol{A} \in \mathbb{R}^{p\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{p\times m}$, and vector $\boldsymbol{c} \in \mathbb{R}^p$. Finally $\boldsymbol{d}(t) : \mathbb{R}_+ \to \mathbb{R}^q$ is a vector function of time, describing the data.

Problem $\mathsf{P}(t)$ is typical in signal processing, and depending on the specific values for the matrices and vectors, it could yield a streaming, i.e., time-varying, LASSO, Group-LASSO, the elastic net, as well as various regularized least-squares problems. The structure of Problem $\mathsf{P}(t)$ is so typical that we specifically use it to devise novel algorithms for its resolution. In particular, we use the fact that the Hessian of f is constant in time.

For handy notation, when $\boldsymbol{x} = \boldsymbol{y}$, we introduce the primal problem,

$$\mathsf{P}_\mathsf{p}(t): \quad \boldsymbol{x}^*(t) = \arg\min\nolimits_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}; t) + g(\boldsymbol{x}), \tag{3}$$

with $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ a closed convex and proper function and f defined as before.

Solving any of the two problems means determining, at each time t, the optimizers $\boldsymbol{x}^*(t)$ or $(\boldsymbol{x}^*(t), \boldsymbol{y}^*(t))$, and therefore, computing the optimizers' trajectory (*i.e.*, the optimizers' evolution in time), up to some arbitrary but fixed accuracy. We notice here that problems $\mathsf{P}(t), \mathsf{P}_\mathsf{p}(t)$ are not available a priori, but they are revealed as time evolves: *e.g.*, problem $\mathsf{P}(t')$ will be revealed at $t = t'$ and known for all $t \geq t'$. In this context, we are interested in modeling how the problems $\mathsf{P}(t), \mathsf{P}_\mathsf{p}(t)$ evolve in time.

We will look at primal and dual first-order methods. Problem (3) is the online version of a composite optimization problem (*i.e.*, of the form $f + g$) and we will consider primal first-

[*] Corresponding author.
E-mail addresses: nicolba@kth.se (N. Bastianello), carlirug@dei.unipd.it (R. Carli), andrea.simonetto@ensta-paris.fr (A. Simonetto).

https://doi.org/10.1016/j.sigpro.2023.109089
0165-1684/

order methods. Note here that g could be the indicator function of a closed convex set, thereby enabling modeling constrained optimization problems varying with time. Problem (1) is the online version of the alternating direction method of multipliers (ADMM) setting, and we will consider dual first-order methods. The idea is to present in a unified way a broad class of online optimization algorithms that can tackle instances of problems (1)-(3). Further note that the two problems could be transformed into each other, if one so wishes, but we prefer to treat them separately to encompass both primal and dual methods.

The focus is on discrete-time settings as in [15,41,47]. In this context, we will use sampling arguments to reinterpret Sections 1 and (3) as a sequence of time-invariant problems. In particular, focusing here only on (3) for simplicity, upon sampling the objective function $f(\boldsymbol{x}; t) + g(\boldsymbol{x})$ at time instants t_k, $k = 0, 1, 2, \ldots$, where the sampling period $T_s := t_k - t_{k-1}$ can be chosen arbitrarily small, one can solve the sequence of time-invariant problems

$$\mathsf{P}_{\mathrm{p}}(t_k): \quad \boldsymbol{x}^*(t_k) = \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}; t_k) + g(\boldsymbol{x}), \qquad k \in \mathbb{N}. \tag{4}$$

By decreasing T_s, an arbitrary accuracy may be achieved when approximating problem (3) with (4). In this context, we will hereafter assume that T_s is a small constant and $T_s < 1$. However, solving (4) for each sampling time t_k may not be computationally affordable in many application domains, even for moderate-size problems. We therefore consider here approximating the discretized optimizers' trajectory $\{\boldsymbol{x}^*(t_k)\}_{k\in\mathbb{N}}$ by using first-order methods. In particular, we will focus on prediction-correction methods [15,30,41,47]. This methodology arises from non-stationary optimization [27,31], parametric programming [15,19,22,24,35,47], and continuation methods in numerical mathematics [1].

This paper extends the current state-of-the-art methods, *e.g.*, [40,41], by offering the following contributions.

1. We provide novel prediction-correction methods in both primal space and dual space for online optimization with constraints. In doing so, we show how existing prediction-correction online algorithms can be generalized with the help of operator theoretical tools. In particular, the abstract methodology we discuss includes special cases such as the ones based on (projected) gradient method [41], proximal point, forward-backward splitting, Peaceman-Rachford splitting [7], as well as the ones based on dual ascent [40]. Moreover, the proposed algorithms includes new online algorithms based on the method of multipliers, dual forward-backward splitting, and ADMM. With our methodology, we obtain unified results, and a general error bound (Proposition 1), which allows one to plug any prediction strategy they are working with and obtain the corresponding asymptotic error.
2. By leveraging the structure of our signal processing problem, we propose and theoretically characterize a prediction strategy which applies extrapolation on a set of past cost functions collected by the online algorithm[1]. The number of historical costs used can be tuned in order to increase accuracy. Differently from the Taylor expansion-based prediction strategy of *e.g.* [43], the prediction can be computed without needing to compute derivatives of the cost. Under suitable assumptions, we analyze the convergence of the resulting online algorithm, in particular by deriving an upper bound to the asymptotic tracking error (*i.e.* the distance from the optimal trajectory $\{\boldsymbol{x}^*(t_k)\}_{k\in\mathbb{N}}$).
3. We further apply the proposed extrapolation prediction strategy to problems with linear constraints such as Problem (1), for which we prove convergence within a bounded neighborhood of the optimal trajectories $\{\boldsymbol{x}^*(t_k), \boldsymbol{y}^*(t_k)\}_{k\in\mathbb{N}}$.

1.1. Related work

Time-varying, streaming, and online problems have a long tradition in signal processing and machine learning. The recent surveys [13,42] cover some key references. From the signal processing literature, we can cite here early algorithms for recursive least-squares and compressive sensing [2,11,45,46], as well as for dynamic filtering [3,4,12]. These signal processing problems are special cases of problem (4), and the algorithms proposed in this paper can then be applied to solve them.

More recently, the works [23,26,41,43] are in line with what we present here, in the sense that they depict time-varying optimization solutions where given a new problem at time t, one attempts at finding an approximate optimizer of it. In this sense, past data help warm starting the algorithm at time t, but do not influence the new sampled problem. In this paper, we take the same approach as these previous works, but propose a novel warm-starting strategy, and theoretically analyze its performance for the different class of problems (1).

This line of research is also related to online convex optimization (OCO) [14,20,39], which was formulated to analyze learning from streaming data. However, differently from our approach, in OCO the set-up is adversarial, in the sense that only information observed up to time t_k can be used to compute the decision to be applied at time t_{k+1}[2]. Once the decision is applied the learner gains access to the new cost function and incurs a regret; importantly, the cost function may be chosen adversarially to maximize this regret.

Finally, we mention the related approach of streaming optimization discussed in [21]. Similarly to the approach in this paper, a new cost function is revealed at each time; with the difference that also a new optimization *variable* is added, and the goal is to solve the overall problem being pieced together over time. In our approach, we focus on a time-varying cost function with a fixed size unknown variable, and assume that the cost function observed at time t_k provides all the information required to compute (in principle) the optimal solution.

Organization. In Section 2, we introduce the necessary background. In Sections 3 and 4, we present the proposed prediction-correction methodology and the novel extrapolation-based prediction approach, and analyze its performance. Section 5 describes the dual version of the proposed approach. Section 6 concludes with several numerical examples.

2. Mathematical background

2.1. Notation

Vectors are written as $\boldsymbol{x} \in \mathbb{R}^n$ and matrices as $\boldsymbol{A} \in \mathbb{R}^{p\times n}$. We denote by $\lambda_{\mathrm{M}}(\boldsymbol{A})$ and $\lambda_{\mathrm{m}}(\boldsymbol{A})$ the largest and smallest eigenvalues of a square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. We use $\|\cdot\|$ to denote the Euclidean norm in the vector space, as well as the respective induced norms for matrices. In particular, given a matrix $\boldsymbol{A} \in \mathbb{R}^{p\times n}$, we have $\|\boldsymbol{A}\| = \sigma_{\mathrm{M}}(\boldsymbol{A}) = \sqrt{\lambda_{\mathrm{M}}(\boldsymbol{A}^\top\boldsymbol{A})}$, where σ_{M} denotes the largest singular value. The gradient of a differentiable function $f(\boldsymbol{x}; t): \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ with respect to $\boldsymbol{x}$ at the point $(\boldsymbol{x}, t)$ is denoted as $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}; t) \in \mathbb{R}^n$, and $\nabla_{\boldsymbol{xx}} f(\boldsymbol{x}; t) \in \mathbb{R}^{n\times n}$ denotes the Hessian of $f(\boldsymbol{x}; t)$ w.r.t. $\boldsymbol{x}$. The notations $\frac{\partial^{(l)}}{\partial t^{(l)}} \nabla_{\boldsymbol{x}} f(\boldsymbol{x}; t) = \nabla_{t\cdots t\boldsymbol{x}} f(\boldsymbol{x}; t)$ denote the l-th derivative w.r.t. t of the gradient. We indicate the inner product of vectors belonging to $\mathbb{R}^n$ as $\langle \boldsymbol{v}, \boldsymbol{u}\rangle := \boldsymbol{v}^\top\boldsymbol{u}$, for all $\boldsymbol{v} \in \mathbb{R}^n, \boldsymbol{u} \in \mathbb{R}^n$, where $(\cdot)^\top$ means

[1] While extrapolation is a known technique in numerical mathematics [33,34], here we fully characterize its theoretical asymptotic error, and we use it in a constrained setting.

[2] Please refer to Remark 2 for further discussions.

transpose. We denote by $\circ$ the composition operation. We use $\{\boldsymbol{x}^\ell\}_{\ell\in\mathbb{N}}$ to indicate sequences of vectors indexed by non-negative integers, for which we define linear convergence as follows (see [32] for details).

Definition 1 (Linear convergence)**.** Let $\{\boldsymbol{x}^\ell\}_{\ell\in\mathbb{N}}$ and $\{\boldsymbol{y}^\ell\}_{\ell\in\mathbb{N}}$ be sequences in $\mathbb{R}^n$, and consider the points $\boldsymbol{x}^*, \boldsymbol{y}^* \in \mathbb{R}^n$. We say that $\{\boldsymbol{x}^\ell\}_{\ell\in\mathbb{N}}$ converges *Q-linearly* to $\boldsymbol{x}^*$ if there exists $\lambda \in (0,1)$ such that: $\left\|\boldsymbol{x}^{\ell+1} - \boldsymbol{x}^*\right\| \le \lambda \left\|\boldsymbol{x}^\ell - \boldsymbol{x}^*\right\|$, $\forall \ell \in \mathbb{N}$.

We say that $\{\boldsymbol{y}^\ell\}_{\ell\in\mathbb{N}}$ converges *R-linearly* to $\boldsymbol{y}^*$ if there exists a Q-linearly convergent sequence $\{\boldsymbol{x}^\ell\}_{\ell\in\mathbb{N}}$ and $C > 0$ such that: $\left\|\boldsymbol{y}^\ell - \boldsymbol{y}^*\right\| \le C \left\|\boldsymbol{x}^\ell - \boldsymbol{x}^*\right\|$, $\forall \ell \in \mathbb{N}$.

2.2. Convex analysis

A function $f : \mathbb{R}^n \to \mathbb{R}$ is *μ-strongly convex*, $\mu > 0$, iff $f(\boldsymbol{x}) - \frac{\mu}{2}\|\boldsymbol{x}\|^2$ is convex. It is said to be *L-smooth* iff $\nabla f(\boldsymbol{x})$ is L-Lipschitz continuous or, equivalently, iff $f(\boldsymbol{x}) - \frac{L}{2}\|\boldsymbol{x}\|^2$ is concave. We denote by $\mathcal{S}_{\mu,L}(\mathbb{R}^n)$ the class of twice differentiable, μ-strongly convex, and L-smooth functions, and $\kappa := L/\mu$ will denote the condition number of such functions. An extended real line function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is *closed* if its epigraph $\mathrm{epi}(f) = \{(\boldsymbol{x}, a) \in \mathbb{R}^{n+1} \mid x \in \mathrm{dom}(f),\ f(\boldsymbol{x}) \le a\}$ is closed. It is *proper* if it does not attain $-\infty$. We denote by $\Gamma_0(\mathbb{R}^n)$ the class of closed, convex and proper functions $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. Notice that functions in $\Gamma_0(\mathbb{R}^n)$ need not be smooth. Given a function $f \in \Gamma_0(\mathbb{R}^n)$, we define its *convex conjugate* as the function $f^* : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ such that $f^*(\boldsymbol{w}) = \sup_{\boldsymbol{x}\in\mathbb{R}^n} \{\langle \boldsymbol{w}, \boldsymbol{x}\rangle - f(\boldsymbol{x})\}$. The convex conjugate of a function $f \in \Gamma_0(\mathbb{R}^n)$ belongs to $\Gamma_0(\mathbb{R}^n)$ as well, and if $f \in \mathcal{S}_{\mu,L}(\mathbb{R}^n)$ then $f^* \in \mathcal{S}_{1/L,1/\mu}(\mathbb{R}^n)$, [37, Chapter 12.H].

The *subdifferential* of a convex function $f \in \Gamma_0(\mathbb{R}^n)$ is defined as the set-valued operator $\partial f : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ such that:

$$\boldsymbol{x} \mapsto \{\boldsymbol{z} \in \mathbb{R}^n \mid \forall \boldsymbol{y} \in \mathbb{R}^n : \ \langle \boldsymbol{y} - \boldsymbol{x}, \boldsymbol{z}\rangle + f(\boldsymbol{x}) \le f(\boldsymbol{y})\},$$

and we denote by $\tilde{\nabla} f(\boldsymbol{x}) \in \partial f(\boldsymbol{x})$ the subgradients. The subdifferential of a convex function is *monotone*, that is, for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$: $0 \le \langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{u} - \boldsymbol{v}\rangle$ where $\boldsymbol{u} \in \partial g(\boldsymbol{x}), \boldsymbol{v} \in \partial g(\boldsymbol{y})$.

2.3. Operator theory

We briefly review some notions and results in operator theory, and we refer to [9,38] for a thorough treatment.

Definition 2. An operator $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ is:

- *λ-Lipschitz*, with $\lambda > 0$, iff $\|\mathcal{T}\boldsymbol{x} - \mathcal{T}\boldsymbol{y}\| \le \lambda\|\boldsymbol{x} - \boldsymbol{y}\|$ for any two $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$; it is *non-expansive* iff $\lambda \in (0,1]$ and *λ-contractive* iff $\lambda \in (0,1)$;
- *β-strongly monotone*, with $\beta > 0$, iff $\beta\|\boldsymbol{x} - \boldsymbol{y}\|^2 \le \langle \boldsymbol{x} - \boldsymbol{y}, \mathcal{T}\boldsymbol{x} - \mathcal{T}\boldsymbol{y}\rangle$, for any two $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$.

By using the Cauchy-Schwarz inequality, a β-strongly monotone operator can be shown to satisfy:

$$\beta\|\boldsymbol{x} - \boldsymbol{y}\| \le \|\mathcal{T}\boldsymbol{x} - \mathcal{T}\boldsymbol{y}\|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n. \tag{5}$$

Definition 3. Let $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ be an operator, a point $\boldsymbol{x}^* \in \mathbb{R}^n$ is a *fixed point* for $\mathcal{T}$ iff $\boldsymbol{x}^* = \mathcal{T}\boldsymbol{x}^*$.

By the Banach-Picard theorem [9, Theorem 1.51], contractive operators have a unique fixed point.

2.4. Operator theory for convex optimization

Operator theory can be employed to solve convex optimization problems; the main idea is to translate a minimization problem into the problem of finding the fixed points of a suitable operator.

Let $f \in \mathcal{S}_{\mu,L}(\mathbb{R}^n)$ and $g \in \Gamma_0(\mathbb{R}^n)$ and consider the optimization problem

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} \{f(\boldsymbol{x}) + g(\boldsymbol{x})\}. \tag{6}$$

Let $\mathcal{T} : \mathbb{R}^p \to \mathbb{R}^p$ and $\mathcal{X} : \mathbb{R}^p \to \mathbb{R}^n$ be two operators. Let $\mathcal{T}$ be λ-contractive and such that its fixed point $\boldsymbol{z}^*$ yields the solution to (6) through the operator $\boldsymbol{x}^* = \mathcal{X}\boldsymbol{z}^*$. Let the operator $\mathcal{X}$ be χ-Lipschitz.

We employ then the *Banach-Picard fixed point algorithm*, defined as the update:

$$\boldsymbol{z}^{\ell+1} = \mathcal{T}\boldsymbol{z}^\ell, \quad \ell \in \mathbb{N}. \tag{7}$$

By the contractiveness of $\mathcal{T}$, the Q-linear convergence to the fixed point is guaranteed [9, Theorem 1.51]:

$$\left\|\boldsymbol{z}^{\ell+1} - \boldsymbol{z}^*\right\| \le \lambda\left\|\boldsymbol{z}^\ell - \boldsymbol{z}^*\right\| \le \lambda^{\ell+1}\left\|\boldsymbol{z}^0 - \boldsymbol{z}^*\right\|, \tag{8}$$

as well as R-linear convergence of $\{\boldsymbol{x}^\ell\}_{\ell\in\mathbb{N}}$ obtained through $\boldsymbol{x}^\ell = \mathcal{X}\boldsymbol{z}^\ell$ as

$$\left\|\boldsymbol{x}^{\ell+1} - \boldsymbol{x}^*\right\| \le \chi\lambda^{\ell+1}\left\|\boldsymbol{z}^0 - \boldsymbol{z}^*\right\|. \tag{9}$$

The following lemma further characterizes the convergence in terms of $\boldsymbol{x}$.

Lemma 1. *Let $\mathcal{X}$ be β-strongly monotone. Then, convergence of the sequence $\{\boldsymbol{x}^\ell\}_{\ell\in\mathbb{N}}$ with $\boldsymbol{x}^\ell = \mathcal{X}\boldsymbol{z}^\ell$ is characterized by the following inequalities:*

$$\left\|\boldsymbol{x}^\ell - \boldsymbol{x}^*\right\| \le \zeta(\ell)\left\|\boldsymbol{x}^0 - \boldsymbol{x}^*\right\|, \qquad \left\|\boldsymbol{x}^\ell - \boldsymbol{x}^0\right\| \le \xi(\ell)\left\|\boldsymbol{x}^0 - \boldsymbol{x}^*\right\| \tag{10}$$

where

$$\zeta(\ell) := \begin{cases} 1, & \text{for } \ell = 0, \\ \frac{\chi}{\beta}\lambda^\ell, & \text{otherwise} \end{cases} \quad \text{and} \quad \xi(\ell) := \begin{cases} 0, & \text{for } \ell = 0, \\ 1 + \frac{\chi}{\beta}\lambda^\ell, & \text{otherwise} \end{cases}. \tag{11}$$

Proof. In the case in which $\ell = 0$, then we have $\left\|\boldsymbol{x}^\ell - \boldsymbol{x}^*\right\| = \left\|\boldsymbol{x}^0 - \boldsymbol{x}^*\right\|$ and $\left\|\boldsymbol{x}^\ell - \boldsymbol{x}^0\right\| = \left\|\boldsymbol{x}^0 - \boldsymbol{x}^0\right\| = 0$, which give the first cases in the definitions of $\zeta(\ell)$ and $\xi(\ell)$. If $\ell > 0$, then by β-strong monotonicity of $\mathcal{X}$, Eq. (5), we have

$$\left\|\boldsymbol{z}^0 - \boldsymbol{z}^*\right\| \le \frac{1}{\beta}\left\|\mathcal{X}\boldsymbol{z}^0 - \mathcal{X}\boldsymbol{z}^*\right\| = \frac{1}{\beta}\left\|\boldsymbol{x}^0 - \boldsymbol{x}^*\right\|. \tag{12}$$

Combining (9) with (12) then yields the second case in the definition of $\zeta(\ell)$. Moreover, using the triangle inequality we have: $\left\|\boldsymbol{x}^\ell - \boldsymbol{x}^0\right\| \le \left\|\boldsymbol{x}^\ell - \boldsymbol{x}^*\right\| + \left\|\boldsymbol{x}^0 - \boldsymbol{x}^*\right\|$, and the second case in the definition of $\xi(\ell)$ follows by (9) and (12). □

Examples of $\mathcal{T}$ and $\mathcal{X}$ operators for primal problems are reported in Example 1 below, while Example 2 discusses the dual solver Alternating direction method of multipliers (ADMM). We now give a formal definition of operator theoretical solver, which will be needed for our developments.

Definition 4 (Operator theoretical solver)**.** Let $\mathcal{T} : \mathbb{R}^p \to \mathbb{R}^p$ and $\mathcal{X} : \mathbb{R}^p \to \mathbb{R}^n$ be two operators, respectively, λ-contractive for $\mathcal{T}$ and χ-Lipschitz and β-strongly monotone for $\mathcal{X}$, such that the solution $\boldsymbol{x}^*$ of (6) can be computed as $\boldsymbol{x}^* = \mathcal{X}\boldsymbol{z}^*$ with $\boldsymbol{z}^*$ being the fixed point of $\mathcal{T}$. Suppose that a recursive method, *e.g.* the Banach-Picard in (7), is available to compute the fixed point $\boldsymbol{z}^*$. Then we call this recursive method an *operator theoretical solver* for problem (6), and call each recursive update of the method a *step* of the solver. We also use the short-hand notation $\mathrm{O}(\lambda, \chi, \beta)$ to indicate such a solver, for which the contraction rates in Lemma 1 are valid.

Example 1 (Operator theoretical solvers)**.** Problem (6) can be solved by applying one of the following *splitting algorithms*:

- *Forward-backward splitting (FBS)* (or *proximal gradient method*): we choose $\mathcal{T} = \text{prox}_{\rho g} \circ (\mathcal{I} - \rho \nabla_{\boldsymbol{x}} f)$, which is contractive for $\rho < 2/L$ and has $\mathcal{X} = \mathcal{I}$; the algorithm is characterized by [44]:

$$\boldsymbol{y}^\ell = \boldsymbol{x}^\ell - \rho \nabla_{\boldsymbol{x}} f(\boldsymbol{x}^\ell), \qquad \boldsymbol{x}^{\ell+1} = \text{prox}_{\rho g}(\boldsymbol{y}^\ell), \quad \ell \in \mathbb{N}. \tag{13}$$

- *Peaceman-Rachford splitting (PRS)*: we choose $\mathcal{T} = \text{refl}_{\rho g} \circ \text{refl}_{\rho f}$, which is contractive for any $\rho > 0$ and has $\mathcal{X} = \text{prox}_{\rho f}$; the algorithm's updates are [18]:

$$\boldsymbol{x}^\ell = \text{prox}_{\rho f}(\boldsymbol{z}^\ell), \qquad \boldsymbol{y}^\ell = \text{prox}_{\rho g}(2\boldsymbol{x}^\ell - \boldsymbol{z}^\ell), \qquad \boldsymbol{z}^{\ell+1} = \boldsymbol{z}^\ell + (\boldsymbol{y}^\ell - \boldsymbol{x}^\ell) \tag{14}$$

  and, from the fixed point $\boldsymbol{z}^*$ of $\mathcal{T}$ we compute the solution $\boldsymbol{x}^*$ through $\mathcal{X} = \text{prox}_{\rho f}$.

If problem (6) does not have a non-smooth term ($g(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathbb{R}^n$), then FBS and PRS reduce to the *gradient descent method* [44] and *proximal point algorithm* (PPA) [36], respectively.

3. Prediction-Correction algorithms

We start by describing in this section the proposed prediction-correction methodology, referring to problem (4):

$$\boldsymbol{x}_k^* = \arg\min_{\boldsymbol{x} \in \mathbb{R}^n} \{f(\boldsymbol{x}; t_k) + g(\boldsymbol{x})\}, \quad k \in \mathbb{N} \tag{15}$$

where hereafter $\boldsymbol{x}_k^* := \boldsymbol{x}^*(t_k)$. Notice that the *size n* of the problem does not change over time, only the cost function f. As said, problem (15) can model a wide range of both constrained and unconstrained optimization problems, in which a smooth term f is (possibly) summed to a non-smooth term g. For example, we may have that g is the indicator function of a constraint set, or a non-smooth function promoting some structural properties (such as an ℓ_1 norm enforcing sparsity).

Remark 1 (Explicit. *v.* implicit time-dependence) Notice that in many data-driven applications, the costs would not depend explicitly on time; rather, they would depend on time-varying data, and hence only implicitly on time. Nonetheless, the model we employ is general enough to account also for implicit time-dependence.

3.1. Methodology

Suppose that an operator theoretical solver for problem (15) is available. The prediction-correction scheme is characterized by the following two steps:

- *Prediction*: at time t_k, we approximate the as yet unobserved cost $f(\boldsymbol{x}; t_{k+1})$ using the past observations; let $\hat{f}_{k+1}(\boldsymbol{x})$ be such approximation, then we solve the problem

$$\hat{\boldsymbol{x}}_{k+1}^* = \arg\min_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ \hat{f}_{k+1}(\boldsymbol{x}) + g(\boldsymbol{x}) \right\} \tag{16}$$

  with initial condition $\boldsymbol{x}_k$, which yields the prediction $\hat{\boldsymbol{x}}_{k+1}^*$. In practice, it is possible to compute only an approximation of $\hat{\boldsymbol{x}}_{k+1}^*$, denoted by $\hat{\boldsymbol{x}}_{k+1}$, by applying N_P steps of the solver.
- *Correction*: when, at time t_{k+1}, the cost $f_{k+1}(\boldsymbol{x}) := f(\boldsymbol{x}; t_{k+1})$ is made available, we can correct the prediction computed at the previous step by solving:

$$\boldsymbol{x}_{k+1}^* = \arg\min_{\boldsymbol{x} \in \mathbb{R}^n} \{f_{k+1}(\boldsymbol{x}) + g(\boldsymbol{x})\} \tag{17}$$

  with initial condition equal to $\hat{\boldsymbol{x}}_{k+1}$. We will denote by $\boldsymbol{x}_{k+1}$ the (possibly approximate) correction computed by applying N_C steps of the solver.

Fig. 1 depicts the flow of the prediction-correction scheme, in which information observed up to time t_k is used to compute the prediction $\hat{\boldsymbol{x}}_{k+1}$. In turn, the prediction serves as a warm-starting condition for the correction problem, characterized by the cost observed at time t_{k+1}.

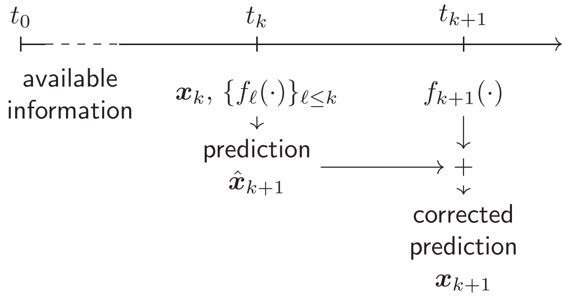

Fig. 1. The prediction-correction scheme.

3.1.1. Solvers

As described above, the proposed methodology requires that an operator theoretical solver for the prediction and correction steps be available. In particular, there are λ-contractive operators $\hat{\mathcal{T}}_{k+1}, \mathcal{T}_{k+1} : \mathbb{R}^p \to \mathbb{R}^p$ with fixed points $\hat{\boldsymbol{z}}_{k+1}^*, \boldsymbol{z}_{k+1}^*$, and χ-Lipschitz, β-strongly monotone operators $\hat{\mathcal{X}}_{k+1}, \mathcal{X}_{k+1} : \mathbb{R}^p \to \mathbb{R}^n$, such that

$$\hat{\boldsymbol{x}}_{k+1}^* = \hat{\mathcal{X}}_{k+1} \hat{\boldsymbol{z}}_{k+1}^* \quad \text{and} \quad \boldsymbol{x}_{k+1}^* = \mathcal{X}_{k+1} \boldsymbol{z}_{k+1}^*.$$

For simplicity, we assume that the convergence rate of the prediction and correction solvers are the same, and we denote them by $\text{O}(\lambda, \chi, \beta)$. Therefore the contraction functions ζ and ξ in Lemma 1 are the same in both cases.

There is a broad range of solvers that can be used within the proposed methodology, depending on the structure of problem (15). For example, if $g \equiv 0$, then *gradient method* and *proximal point algorithms* are suitable solvers, while if $g \not\equiv 0$ then *forward-backward*[3] and *Peaceman-Rachford* splitting can be used.

3.2. Prediction methods

The most straightforward prediction method is the choice $\hat{f}_{k+1}(\boldsymbol{x}) = f_k(\boldsymbol{x})$ which simply employs the last observed cost as a prediction of the next. However, as we will discuss in the following, using a more sophisticated prediction strategy can lead to better performance.

In particular, we look at extrapolation-based prediction. First of all we briefly review a numerical technique for polynomial interpolation [34], which we then leverage to design a novel prediction strategy.

3.2.1. Polynomial interpolation

Let $\varphi : \mathbb{R} \to \mathbb{R}$ be a function that we want to interpolate from the pairs $\{(t_i, \varphi_i)\}_{i=1}^I$ where $\varphi_i := \varphi(t_i)$, and with $t_i \neq t_j$ for any $i \neq j$. The interpolated function is then defined as [34, Theorem 8.1]:

$$\hat{\varphi}(t) = \sum_{i=1}^I \varphi_i \ell_i(t), \qquad \text{with} \qquad \ell_i(t) = \prod_{\substack{1 \leq j \leq I \\ j \neq i}} \frac{t - t_j}{t_i - t_j}. \tag{18}$$

The interpolation error can be characterized by [34, Theorem 8.2]:

$$\varphi(t) - \hat{\varphi}(t) = \frac{\varphi^{(I)}(\upsilon)}{I!} \omega_I(t) \qquad \text{with} \qquad \omega_I(t) = \prod_{i=1}^I (t - t_i) \tag{19}$$

and where υ is a scalar in the smallest interval that contains t and $\{t_i\}_{i=1}^I$.

Since in the following we are interested in evaluating the interpolated function at a point t that lies outside the interval $[t_1, t_I]$, we will refer to the resulting function as *extrapolation*.

[3] Also called *proximal gradient method*.

3.2.2. Extrapolation-based prediction

Let us now apply the polynomial interpolation technique (18) to the function $f(\boldsymbol{x};t):\mathbb{R}^n\times\mathbb{R}_+\to\mathbb{R}^n$ w.r.t. the scalar variable $t\in\mathbb{R}_+$. In particular, we compute the predicted function $\hat{f}_{k+1}$ from the set of past functions $\{f_i(\boldsymbol{x})\}_{i=k-I+1}^{k}$. Since the sampling times are multiples of T_s, it is easy to see that the coefficients in (18) become:

$$\ell_i(t_{k+1})=\prod_{\substack{k-I+1\leq j\leq k\\ j\neq i}}\frac{t_{k+1}-t_j}{t_i-t_j}=\prod_{\substack{1\leq h\leq I\\ h\neq i-(k+1)}}\frac{h}{i-(k+1)-h}=(-1)^{i-(k+1)}\binom{I}{i-(k+1)}$$

and letting $\ell_i:=\ell_i(t_{k+1})$ the prediction is thus given by

$$\hat{f}_{k+1}(\boldsymbol{x})=\sum_{i=k-I+1}^{k}\ell_i f_i(\boldsymbol{x}),\quad\forall\boldsymbol{x}\in\mathbb{R}^n. \tag{20}$$

In general, however, the predicted cost $\hat{f}_{k+1}$ may not be strongly convex – as a matter of fact, it can even fail to be convex.

However, and crucially, since for our $f(\boldsymbol{x};t)$ the Hessian is time-independent, then $\nabla_{\boldsymbol{xx}}f(\boldsymbol{x};t)=\nabla_{\boldsymbol{xx}}f(\boldsymbol{x})$ for any $(\boldsymbol{x};t)\in\mathbb{R}^n\times\mathbb{R}_+$, which implies that

$$\nabla_{\boldsymbol{xx}}\hat{f}_{k+1}(\boldsymbol{x})=(\sum_{i=k-I+1}^{k}\ell_i)\nabla_{\boldsymbol{xx}}f(\boldsymbol{x})=\nabla_{\boldsymbol{xx}}f(\boldsymbol{x}), \tag{21}$$

having used the fact that $\sum_{i=k-I+1}^{k}\ell_i=1$. Therefore $\hat{f}_{k+1}$ inherits the same strong convexity and smoothness properties of the original cost.

This property is inherent to our regularized least-squares structure and typical in signal processing, and very useful for good prediction.

Remark 2 (Alternative prediction strategies)**.** We mention here two alternative prediction strategies that have been proposed in the literature. The simpler one, widely used in the context of *online learning* [39], is the choice of $\hat{f}_{k+1}=f_k$. This strategy, hereafter called "one-step-back" prediction, is particularly suited to adversarial environments, where the future cost f_{k+1} is chosen by the adversary and the best decision we can make is based on f_k. Alternatively, under the assumption that the gradient of f_k is differentiable in time we can choose the *Taylor expansion*-based prediction $\nabla_{\boldsymbol{x}}\hat{f}_{k+1}(\boldsymbol{x})=\nabla_{\boldsymbol{x}}f_k(\boldsymbol{x}_k)+T_\mathrm{s}\nabla_{t\boldsymbol{x}}f_k(\boldsymbol{x}_k)+\nabla_{\boldsymbol{xx}}f_k(\boldsymbol{x}_k)(\boldsymbol{x}-\boldsymbol{x}_k)$ [43].

Remark 3 (Computational comparison)**.** From Remark 2, the Taylor-based prediction is $\nabla_{\boldsymbol{x}}\hat{f}_{k+1}(\boldsymbol{x})=\nabla_{\boldsymbol{x}}f_k(\boldsymbol{x}_k)+T_\mathrm{s}\nabla_{t\boldsymbol{x}}f_k(\boldsymbol{x}_k)+\nabla_{\boldsymbol{xx}}f_k(\boldsymbol{x}_k)(\boldsymbol{x}-\boldsymbol{x}_k)$. This means that to compute it, we need to evaluate the gradient, the Hessian, and the time-derivative of the gradient. On the other hand, the extrapolation-based prediction only requires the computation of gradients from the I past costs that are stored – this means that we only need access to an *oracle* of the gradients, and building a prediction has a much lower cost. Finally, we remark that the computationally cheaper approach is the one-step-back prediction $\hat{f}_{k+1}=f_k$, which requires accessing the oracle of only one past cost. Nonetheless, as the theoretical and numerical results will show, the more refined extrapolation-based prediction achieves much smaller tracking error than using $\hat{f}_{k+1}=f_k$, thus justifying its higher computational burden.

4. Primal online algorithms

We are now ready to present our main convergence results. We start by formally stating the required assumptions, and then we provide bounds to the tracking error achieved by the proposed prediction-correction method.

4.1. Assumptions

Assumption 1. *(i)* The cost function $f:\mathbb{R}^n\times\mathbb{R}_+\to\mathbb{R}$ belongs to $\mathcal{S}_{\mu,L}(\mathbb{R}^n)$ uniformly in t. *(ii)* The function $g:\mathbb{R}^n\to\mathbb{R}\cup\{+\infty\}$ either belongs to $\Gamma_0(\mathbb{R}^n)$, or $g(\cdot)\equiv 0$. *(iii)* The solution to (15) is finite for any $k\in\mathbb{N}$.

Assumption 1*(i)* guarantees that problem (15) is strongly convex and has a unique solution for each time instance. Uniqueness of the solution implies that the solution trajectory is also unique.

Assumption 2. The gradient of function f has bounded time derivative, that is, there exists $C_0>0$ such that $\|\nabla_{t\boldsymbol{x}}f(\boldsymbol{x};t)\|\leq C_0$ for any $\boldsymbol{x}\in\mathbb{R}^n$, $t\in\mathbb{R}_+$.

By imposing Assumption 2 we ensure that the solution trajectory is Lipschitz in time, as we will see, and therefore prediction-type methods would work well.

Assumption 3. The function f has a static Hessian, that is, $\nabla_{\boldsymbol{xx}}f(\boldsymbol{x};t)=\nabla_{\boldsymbol{xx}}f(\boldsymbol{x})$ for any $(\boldsymbol{x};t)\in\mathbb{R}^n\times\mathbb{R}_+$. For the chosen extrapolation order $I\in\mathbb{N}$, $I\geq 2$, there exists $C(I)>0$ such that:

$$\left\|\frac{\partial^{(I)}}{\partial t^{(I)}}\nabla_{\boldsymbol{x}}f(\boldsymbol{x}^*_{k+1};\tau)\right\|\leq C(I),\quad\tau\in[t_{k+1-I},t_{k+1}]. \tag{22}$$

As mentioned in Section 3.2.2 a static Hessian guarantees that the extrapolation-based prediction is strongly convex. The bound on the I-th time-derivative of the gradient will instead serve to quantify the quality of the prediction, by comparing $\hat{\boldsymbol{x}}^*_{k+1}$ and the true optimum $\boldsymbol{x}^*_{k+1}$.

4.2. Convergence

We start by presenting a general bound (meta)-proposition, which can be used to derive the asymptotic error for a large variety of prediction strategies, and it is of independent interest.

Proposition 1 (General error bound)**.** *Let Assumption 1 hold and consider any prediction strategy that uses the same functional class as the original problem (15). Let $\sigma_k,\tau_k\in[0,+\infty)$ be such that for any $k\in\mathbb{N}$:*

$$\left\|\boldsymbol{x}^*_{k+1}-\boldsymbol{x}^*_k\right\|\leq\sigma_k\quad\text{and}\quad\left\|\hat{\boldsymbol{x}}^*_{k+1}-\boldsymbol{x}^*_{k+1}\right\|\leq\tau_k. \tag{23}$$

Then the error incurred by a prediction-correction method that uses the solver $\mathsf{O}(\lambda,\chi,\beta)$ is upper bounded by:

$$\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*_{k+1}\right\|\leq\zeta(N_\mathrm{C})\Big(\zeta(N_\mathrm{P})\left\|\boldsymbol{x}_k-\boldsymbol{x}^*_k\right\|+\zeta(N_\mathrm{P})\sigma_k+\xi(N_\mathrm{P})\tau_k\Big), \tag{24}$$

with functions ζ and ξ defined in Lemma 1.

Proof. See Appendix A.1. □

We are now ready to bound σ_k and τ_k for our prediction strategy. First, we present a useful lemma that, employing the assumptions in Section 4.1, bounds the distance between consecutive points in the optimal trajectory $\{\boldsymbol{x}^*_k\}_{k\in\mathbb{N}}$.

Lemma 2. *Let Assumptions 1 and 2 hold, then the distance between the optimizers of problems (15) at t_k and t_{k+1} is bounded by:*

$$\left\|\boldsymbol{x}^*_{k+1}-\boldsymbol{x}^*_k\right\|\leq C_0T_\mathrm{s}/\mu. \tag{25}$$

Proof. See Appendix A.3. □

The second step is to provide a bound on the distance between the optimizer of the prediction problem and the actual optimizer $\boldsymbol{x}^*_{k+1}$, i.e., a τ_k for our prediction strategy.

Lemma 3. *Let Assumptions 1 and 3 hold. Using the extrapolation-based prediction (20) of order I for f yields the following prediction error:*

$$\left\|\hat{\boldsymbol{x}}^*_{k+1}-\boldsymbol{x}^*_{k+1}\right\|\leq C(I)T_\mathrm{s}^I/\mu. \tag{26}$$

Proof. See Appendix A.4. □

With these lemmas in place we can now characterize the convergence when the extrapolation-based prediction (20) is employed.

Theorem 1. *Consider Problem (15). Consider the prediction-correction algorithm with the extrapolation-based prediction strategy (20) of order $I \in \mathbb{N}$, $I \geq 2$, for f. Let Assumptions 1 to 3 hold. Consider the operator theoretic solver $\mathsf{O}(\lambda, \beta, \chi)$ to solve both the prediction and correction problems with contraction rates ζ and ξ given in Lemma 1. Choose the prediction and correction horizons N_P and N_C such that*

$$\zeta(N_\mathrm{C})\zeta(N_\mathrm{P}) < 1.$$

Then the trajectory $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ generated by the prediction-correction algorithm converges Q-linearly with rate $\zeta(N_\mathrm{C})\zeta(N_\mathrm{P})$ to a neighborhood of the optimal trajectory $\{\boldsymbol{x}_k^*\}_{k\in\mathbb{N}}$, whose radius is upper bounded as

$$\limsup_{k\to\infty} \left\|\boldsymbol{x}_k - \boldsymbol{x}_k^*\right\| = \frac{\zeta(N_\mathrm{C})}{\mu} \frac{[\zeta(N_\mathrm{P})C_0 T_\mathrm{s} + C(I)\xi(N_\mathrm{P})T_\mathrm{s}^I]}{1 - \zeta(N_\mathrm{C})\zeta(N_\mathrm{P})}. \tag{27}$$

Proof. See Appendix A.5. □

Remark 4 (. N_C and N_P choice) Notice that if the operator $\mathcal{X}$ converting between $\boldsymbol{z}$ and the primal variable $\boldsymbol{x}$ is the identity (which is the case *e.g.* for gradient and proximal gradient methods), then $\zeta(N_\mathrm{C})\zeta(N_\mathrm{P}) < 1$ is automatically satisfied whenever at least one of N_C or N_P is non-zero.

5. Dual online algorithms

We now propose a dual version to the prediction-correction methodology, that allows us to solve linearly constrained online problems. We apply the extrapolation-based prediction to this class of problems and study the convergence of the resulting method.

5.1. Problem formulation

We are interested in solving the following online convex optimization problem with linear constraints, cf. (1):

$$\boldsymbol{x}^*(t), \boldsymbol{y}^*(t) = \arg\min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{y}\in\mathbb{R}^m} f(\boldsymbol{x}; t) + h(\boldsymbol{y}), \quad \text{s.t. } \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} = \boldsymbol{c}, \tag{28}$$

where $\boldsymbol{A} \in \mathbb{R}^{p\times n}$, $\boldsymbol{B} \in \mathbb{R}^{p\times m}$ and $\boldsymbol{c} \in \mathbb{R}^p$. The following assumption will hold throughout this section, and we will further use Assumptions 2 and 3 for f.

Assumption 4. *(i)* The cost function f belongs to $\mathcal{S}_{\mu,L}(\mathbb{R}^n)$ uniformly in time and satisfies Assumption 2. *(ii)* The cost h either belongs to $\Gamma_0(\mathbb{R}^m)$ or $h(\cdot) \equiv 0$ with $\boldsymbol{B} = \boldsymbol{0}$. *(iii)* The matrix $\boldsymbol{A} \in \mathbb{R}^{p\times n}$ is full row rank and the vector $\boldsymbol{c}$ can be written as the sum of two vectors $\boldsymbol{c}' \in \mathrm{im}(\boldsymbol{A})$ and $\boldsymbol{c}'' \in \mathrm{im}(\boldsymbol{B})$[4]

The Fenchel dual of (28) is

$$\boldsymbol{w}^*(t) = \arg\min_{\boldsymbol{w}\in\mathbb{R}^p} \left\{d^f(\boldsymbol{w}; t) + d^h(\boldsymbol{w})\right\} \tag{29}$$

where $d^f(\boldsymbol{w}; t) = f^*(\boldsymbol{A}^\top \boldsymbol{w}; t) - \langle \boldsymbol{w}, \boldsymbol{c}\rangle$ and $d^h(\boldsymbol{w}) = h^*(\boldsymbol{B}^\top \boldsymbol{w})$. Problem (29) conforms to the class of problems that can be solved with the prediction-correction splitting methods of Section 3. Indeed, by Assumption 4 we can see that $d^f \in \mathcal{S}_{\bar{\mu},\bar{L}}(\mathbb{R}^p)$, with $\bar{\mu} := \lambda_\mathrm{m}(\boldsymbol{A}\boldsymbol{A}^\top)/L$ and $\bar{L} := \lambda_\mathrm{M}(\boldsymbol{A}\boldsymbol{A}^\top)/\mu$ [18, Prop. 4]; and $d^h \in \Gamma_0(\mathbb{R}^p)$ [9, Cor. 13.38]. We further know that the gradient of d^f is characterized by [17]:

$$\nabla_{\boldsymbol{w}} d^f(\boldsymbol{w}; t) = \boldsymbol{A}\bar{\boldsymbol{x}}(\boldsymbol{w}, t) - \boldsymbol{c}, \quad \bar{\boldsymbol{x}}(\boldsymbol{w}, t) : = \arg\min_{\boldsymbol{x}} \left\{f(\boldsymbol{x}; t) - \langle \boldsymbol{A}^\top \boldsymbol{w}, \boldsymbol{x}\rangle\right\}. \tag{30}$$

Finally, assuming that there exists $C_0 \geq 0$ such that $\|\nabla_{t\boldsymbol{x}} f(\boldsymbol{x}; t)\| \leq C_0$, then we can prove that also the gradient of the dual cost d^f has bounded rate of change.

Lemma 4. *Let Assumption 4 hold for the primal problem (28). Then d^f is such that, for any $\boldsymbol{x} \in \mathbb{R}^n$ and $t \in \mathbb{R}_+$:*

$$\left\|\nabla_{t\boldsymbol{w}} d^f(\boldsymbol{w}; t)\right\| \leq \|\boldsymbol{A}\| C_0/\mu =: \bar{C}_0.$$

Proof. See Appendix B.1. □

Remark 5 (Full rank. $\boldsymbol{A}$) The assumption that $\boldsymbol{A}$ be full row rank is necessary to guarantee that $d^f \in \mathcal{S}_{\bar{\mu},\bar{L}}(\mathbb{R}^p)$. However, when problem (31) reduces to $\min_{\boldsymbol{x}} f(\boldsymbol{x})$ s.t. $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{c}$, this assumption *can be relaxed*. In this case we are able to prove that the dual function d^f is *strongly convex in the subspace of the image of $\boldsymbol{A}$, i.e.*, $\mathrm{im}(\boldsymbol{A})$. Therefore, if $\mathrm{im}(\boldsymbol{A})$ is an invariant set for the trajectory generated by the solver, the solver is contractive and the convergence analysis of this paper applies to show linear convergence. The solvers dual ascent and method of multipliers indeed satisfy these conditions, see [40] for more details.

5.2. Dual prediction-correction methodology

Applying the same approach of Section 3, we are interested in solving (28) sampled at times t_k, $k \in \mathbb{N}$:

$$\boldsymbol{x}_k^*, \boldsymbol{y}_k^* = \arg\min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{y}\in\mathbb{R}^m} \{f(\boldsymbol{x}; t_k) + h(\boldsymbol{y})\} \quad \text{s.t. } \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} = \boldsymbol{c} \tag{31}$$

where $\boldsymbol{x}_k^* = \boldsymbol{x}^*(t_k)$, $\boldsymbol{y}_k^* = \boldsymbol{y}^*(t_k)$. The sequence of dual problems is then

$$\boldsymbol{w}_k^* = \arg\min_{\boldsymbol{w}\in\mathbb{R}^p} \left\{d^f(\boldsymbol{w}; t_k) + d^h(\boldsymbol{w})\right\} \tag{32}$$

with $k \in \mathbb{N}$, $\boldsymbol{w}_k^* := \boldsymbol{w}^*(t_k)$. As mentioned above, (32) can be solved by the prediction-correction methods described in Section 3. The goal then is to design a suitable prediction strategy.

The idea is to apply the extrapolation-based prediction of Section 3 to the primal cost function f_k, hence choosing $\hat{f}_{k+1}(\boldsymbol{x}) = \sum_{i=1}^I \ell_i f_{k+1-i}(\boldsymbol{x})$. The corresponding dual prediction problem then is

$$\hat{\boldsymbol{w}}_{k+1}^* = \arg\min_{\boldsymbol{w}\in\mathbb{R}^p} \left\{\hat{d}_{k+1}^f(\boldsymbol{w}) + \hat{d}^h(\boldsymbol{w})\right\} \tag{33}$$

with $\hat{d}_{k+1}^f(\boldsymbol{w}) = \hat{f}_{k+1}^*(\boldsymbol{A}^\top \boldsymbol{w}) - \langle \boldsymbol{w}, \boldsymbol{c}\rangle$.

5.3. Convergence analysis

The following result characterizes the convergence in terms of the primal and dual variables.

Theorem 2. *Consider the problem (31). Apply the prediction-correction method defined in Section 3 to the dual problem (32), with extrapolation-based prediction applied to f. Let $\mathsf{O}(\lambda, \beta, \chi)$ be a suitable dual solver with contraction rates $\bar{\zeta}$ and $\bar{\xi}$ given in Lemma 1 for $d^f \in \mathcal{S}_{\bar{\mu},\bar{L}}(\mathbb{R}^p)$. Let Assumption 4 hold.*

Choose the prediction and correction horizons such that

$$\bar{\zeta}(N_\mathrm{C})\bar{\zeta}(N_\mathrm{P}) < 1.$$

Then the dual trajectory $\{\boldsymbol{w}_k\}_{k\in\mathbb{N}}$ generated by the dual prediction-correction method converges to a neighborhood of the optimal trajectory $\{\boldsymbol{w}_k^*\}_{k\in\mathbb{N}}$, whose radius is upper bounded as

$$\limsup_{k\to\infty} \left\|\boldsymbol{w}_k - \boldsymbol{w}_k^*\right\| = \frac{\bar{\zeta}(N_\mathrm{C})}{\bar{\mu}} \frac{[\bar{\zeta}(N_\mathrm{P})\bar{C}_0 T_\mathrm{s} + \bar{\xi}(N_\mathrm{P})(\|\boldsymbol{A}\|/\mu)C(I)T_\mathrm{s}^I]}{1 - \bar{\zeta}(N_\mathrm{C})\bar{\zeta}(N_\mathrm{P})}.$$

[4] This assumption ensures that the problem does indeed have a solution; otherwise, it would not be possible to satisfy the linear constraints.

Moreover, the primal trajectories $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$, $\{\boldsymbol{B}\boldsymbol{y}_k\}_{k\in\mathbb{N}}$ *converge to a neighborhood of the optimal trajectories* $\{\boldsymbol{x}_k^*\}_{k\in\mathbb{N}}$ $\{\boldsymbol{B}\boldsymbol{y}_k^*\}_{k\in\mathbb{N}}$, *whose radii are upper bounded as*

$$\limsup_{k\to\infty}\left\|\boldsymbol{x}_k-\boldsymbol{x}_k^*\right\| = (\|\boldsymbol{A}\|/\mu)\limsup_{k\to\infty}\left\|\boldsymbol{w}_k-\boldsymbol{w}_k^*\right\|,$$

$$\limsup_{k\to\infty}\left\|\boldsymbol{B}(\boldsymbol{y}_k-\boldsymbol{y}_k^*)\right\| = \|\boldsymbol{B}\|(\|\boldsymbol{A}\|^2/\mu+1/\rho)\limsup_{k\to\infty}\left\|\boldsymbol{w}_k-\boldsymbol{w}_k^*\right\|.$$

Proof. See Appendix B.2. □

We conclude this section showing an example of online algorithm that results when applying the prediction-correction approach in the dual space.

Example 2 (Prediction-correction ADMM)**.** The well known alternating direction method of multipliers (ADMM) applied to $\min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x})+h(\boldsymbol{y})$, s.t. $\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{y}=\boldsymbol{c}$ is characterized by the updates [6, eq. (8)][5]

$$\boldsymbol{x}^\ell = \arg\min_{\boldsymbol{x}\in\mathbb{R}^n}\left\{f(\boldsymbol{x})+\frac{\rho}{2}\left\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{z}^\ell/\rho-\boldsymbol{c}\right\|^2\right\},\quad \boldsymbol{w}^\ell=\boldsymbol{z}^\ell-\rho(\boldsymbol{A}\boldsymbol{x}^\ell-\boldsymbol{c})$$

$$\boldsymbol{y}^\ell = \arg\min_{\boldsymbol{y}\in\mathbb{R}^m}\left\{h(\boldsymbol{y})+\frac{\rho}{2}\left\|\boldsymbol{B}\boldsymbol{y}-(2\boldsymbol{w}^\ell-\boldsymbol{z}^\ell)/\rho\right\|^2\right\},\ \boldsymbol{u}^\ell=2\boldsymbol{w}^\ell-\boldsymbol{z}^\ell-\rho\boldsymbol{B}\boldsymbol{y}^\ell,$$

$$\boldsymbol{z}^{\ell+1}=\boldsymbol{z}^\ell+2(\boldsymbol{u}^\ell-\boldsymbol{w}^\ell),\quad \ell\in\mathbb{N}, \tag{34}$$

where only the primal costs f and h play an explicit role, and where $\boldsymbol{w}^\ell$ is the vector of dual variables of the problem. Therefore, when applying ADMM as a solver for both the prediction and correction problems, we apply (34) replacing f with $\hat{f}_{k+1}(\boldsymbol{x})=\sum_{i=1}^{I}\ell_i f_{k+1-i}(\boldsymbol{x})$ and $f_{k+1}(\boldsymbol{x})$, respectively.

6. Numerical results

In this section, we present extensive numerical results to showcase the performance of the proposed prediction strategy. In particular, We apply our algorithm to three synthetic benchmarks, stemming from time-varying regularized least-squares, time-varying online learning with ADMM, and online robotic tracking, as well as one real-data benchmark stemming from online graph signal processing.

We compare our prediction strategy to other available ones, namely one-step-back prediction [39], Taylor-based prediction using either the exact computations for the time-derivatives and backward finite difference [41,43], the simplified prediction strategy of [25], and two ZeaD prediction formulas [33]. Other prediction strategies do exist, but they would typically involve more complex computations or more memory (i.e., longer time horizons). While we do not compare all the methods in all the examples, the interested reader is referred to the `tvopt` Python package[6] [5] which provides all the tools for more extensive comparisons.

6.1. Time-varying regularized least-squares

We consider the composite problem (cf[43].):

$$f(\boldsymbol{x};t)=\|\boldsymbol{x}-\boldsymbol{b}(t)\|^2/2+\epsilon\log(1+\exp(\langle\boldsymbol{1}_n,\boldsymbol{x}\rangle))\ \text{and}\ g(\boldsymbol{x})=\nu\|\boldsymbol{x}\|_1 \tag{35}$$

with $n=20$, and where $\boldsymbol{b}(t)\in\mathbb{R}^n$ is a signal with sinusoidal components (with angular velocity $\omega=0.02\pi$ and randomly generated phases), $\epsilon=0.75$, $\nu=0.5$. The function f has $\mu=1$, $L=1+\epsilon N/4$, $C_0=\omega$. The prediction and correction problems are solved using the proximal gradient method (a.k.a. forward-backward splitting) [9] with step-size $\alpha=2/(L+\mu)$, and $N_\text{P}=[5,20,40]$, $N_\text{C}=5$. We run the simulations for three values of the sampling time $T_\text{s}=[0.002,0.02,0.2]$.

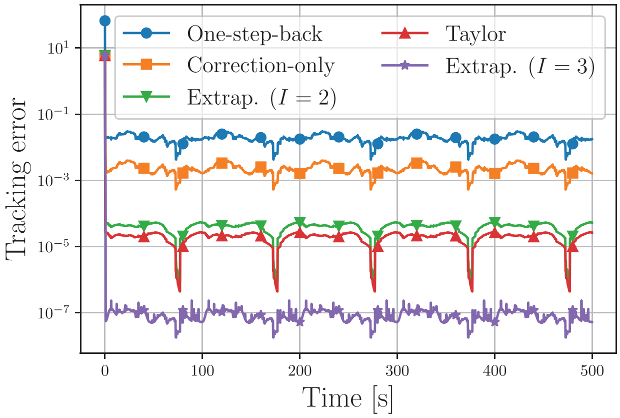

Fig. 2. Tracking error comparison with $T_\text{s}=0.2$ and $N_\text{P}=20$.

Table 1
Comparison of asymptotic errors observed in the numerical simulations for (35).

Method	$T_\text{s}=0.2$		
	$N_\text{P}=5$	$N_\text{P}=20$	$N_\text{P}=40$
One-step-back	3.39×10^{-2}	3.02×10^{-2}	3.02×10^{-2}
Correction-only	3.96×10^{-3}	3.96×10^{-3}	3.96×10^{-3}
Taylor	4.21×10^{-4}	5.45×10^{-5}	5.45×10^{-5}
Extrapolation I=2	4.19×10^{-4}	2.72×10^{-5}	2.72×10^{-5}
Extrapolation I=3	4.18×10^{-4}	2.35×10^{-7}	5.25×10^{-7}
		$T_\text{s}=0.02$	
	$N_\text{P}=5$	$N_\text{P}=20$	$N_\text{P}=40$
One-step-back	3.42×10^{-3}	3.02×10^{-3}	3.02×10^{-3}
Correction-only	4.02×10^{-4}	4.02×10^{-4}	4.02×10^{-4}
Taylor	4.28×10^{-5}	6.67×10^{-7}	6.63×10^{-7}
Extrapolation I=2	4.26×10^{-5}	3.39×10^{-7}	3.31×10^{-7}
Extrapolation I=3	4.24×10^{-5}	6.82×10^{-8}	5.48×10^{-10}
		$T_\text{s}=0.002$	
	$N_\text{P}=5$	$N_\text{P}=20$	$N_\text{P}=40$
One-step-back	3.15×10^{-4}	2.78×10^{-4}	2.78×10^{-4}
Correction-only	3.71×10^{-5}	3.71×10^{-5}	3.71×10^{-5}
Taylor	3.91×10^{-6}	9.46×10^{-9}	5.62×10^{-9}
Extrapolation I=2	3.91×10^{-6}	7.55×10^{-9}	2.81×10^{-9}
Extrapolation I=3	3.91×10^{-6}	6.33×10^{-9}	1.67×10^{-12}

We compare the proposed extrapolation-based prediction with order $I=2$ (*i.e.* $\hat{f}_{k+1}=2f_k-f_{k-1}$) and order $I=3$ (*i.e.* $\hat{f}_{k+1}=3f_k-3f_{k-1}+f_{k-2}$) against the following methods:

- "One-step-back": the predictive online gradient characterized by $\boldsymbol{x}_{k+1}=\boldsymbol{x}_k-\alpha\nabla f_k(\boldsymbol{x}_k)$ [39, p. 132];
- "Correction-only": the online gradient characterized by[7] $\boldsymbol{x}_{k+1}=\boldsymbol{x}_k-\alpha\nabla f_{k+1}(\boldsymbol{x}_k)$ [13, eq. (2)];
- "Taylor" (of order 2): the prediction-correction method using the Taylor expansion-based prediction (see Remark 2) [43].

In Fig. 2 we report a first comparison in terms of the tracking error evolution for the different methods, with $T_\text{s}=0.2$ and $N_\text{P}=20$ (for the methods using prediction). As we can see, an extrapolation of the third order outperforms all other methods, while in general the prediction-correction methods outperform the one-step-back and correction-only approaches.

These observations hold up in Table 1, which reports the asymptotic tracking error for the different strategies. Combining prediction and correction achieves better results; moreover, the

[5] See also the arXiv version of [6] which reports the derivation of eq. (8) in Appendix A https://arxiv.org/abs/1901.09252.

[6] Code available here https://github.com/nicola-bastianello/tvopt.

[7] Note how it differs from the "one-step-back" since the gradient of f_{k+1} is used instead of the gradient of f_k.

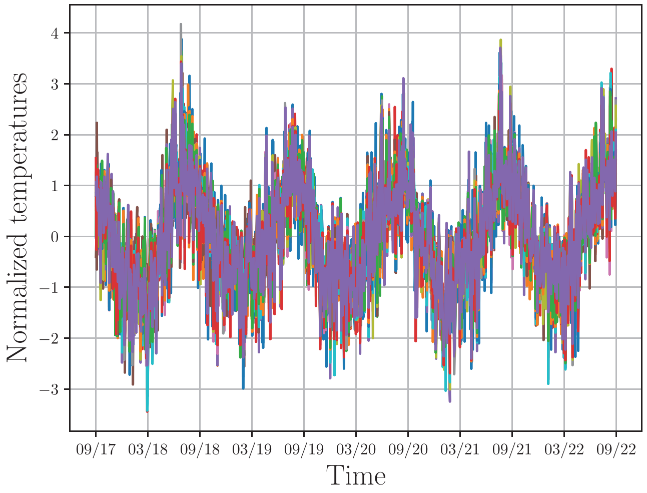

Fig. 3. Normalized temperatures over the range Sep. 2017 to Sep. 2022.

larger the sampling time is, the larger the asymptotic error, which is in accordance with the theory. Also we observe how extrapolation with $I=3$ can boost performance, especially when N_{P} is large, with respect to Taylor – this is due to the fact that in Theorem 1 the asymptotic error depends on T_s^I. We also recall, by Remark 3, that the computational complexity of building a Taylor-based prediction exceeds that of the extrapolation-based prediction, since the former needs access to second order derivatives while the latter only to the gradient, so in practice it may not be reasonable to go beyond a Taylor prediction of order 2.

6.2. Online graph signal processing

As a second example, we consider an online graph signal processing problem, in which the goal is to learn the time-varying topology of a graph from signals observed at the nodes [28]. Formally, we want to reconstruct the topologies of the graphs in the sequence $\{\mathcal{G}_k=(\mathcal{V},\mathcal{E}_k,\boldsymbol{S}_k)\}_{k\in\mathbb{N}}$, where $\mathcal{V}=1,\dots,N$ is the set of nodes, $\mathcal{E}_k$ are the edges at time k, and $\boldsymbol{S}_k\in\mathbb{R}^{N\times N}$ is the graph shift operator that represents the topology, which we need to reconstruct. By employing the smoothness-based model of [28, sec. IV.C], learning the time-varying topology requires that we solve the online problem $\min_{\boldsymbol{S}\in\mathbb{R}^{N\times N}} f_k(\boldsymbol{S})+g(\boldsymbol{S})$, where

$$f_k(\boldsymbol{S})=\operatorname{tr}\left(\operatorname{diag}(\boldsymbol{S}\boldsymbol{1})\hat{\boldsymbol{\Sigma}}_k\right)-\operatorname{tr}\left(\boldsymbol{S}\hat{\boldsymbol{\Sigma}}_k\right)+\frac{\lambda_1}{4}\|\boldsymbol{S}\|_F^2-\lambda_2\boldsymbol{1}^\top\log(\boldsymbol{S}\boldsymbol{1})$$

with $\hat{\boldsymbol{\Sigma}}=\frac{1}{K}\boldsymbol{X}_k\boldsymbol{X}_k^\top$ and $\boldsymbol{X}_k\in\mathbb{R}^{N\times K}$ stacks K samples from the nodes' signals at time k. Moreover, $g(\boldsymbol{S})=\iota_{\mathcal{S}}(\boldsymbol{S})$ is the indicator function of the set $\mathcal{S}$ of non-negative symmetric matrices with zero diagonal[8]

We use the dataset of hourly temperature measurements at 25 weather stations across Ireland[9], collected from Sep. 2017 to Sep. 2022. Fig. 3 depicts the normalized temperatures observed at the different stations over this range. We follow the set-up of [28, sec. VI.B], using the proximal gradient method as solver, with $N_{\mathrm{P}}=N_{\mathrm{C}}=1$. In Table 1 we compare the proposed prediction-correction method using an extrapolation-based strategy (with $I=2$ and $I=3$) with a correction-only approach and with [28], which employs a Taylor expansion-based prediction.

As we can see in Table 2, introducing a prediction improves the performance over a correction-only approach. And, while the Taylor expansion-based method has very similar performance to the extrapolation with $I=2$, the use of an additional past cost, with $I=3$, in turn improves performance. Notice that the use of higher order extrapolation does not yield the same drastic improvement as in the synthetic problem of the previous section, since it is affected by the noise in the real data and the $C(I)$'s can be rather large for I greater than 2 or 3.

6.3. Online learning with ADMM

Consider now the online linear regression problem $\min_{\boldsymbol{x}}\frac{1}{2}\sum_{i=1}^N\left\|\boldsymbol{A}^i\boldsymbol{x}-\boldsymbol{b}_k^i\right\|^2+\nu\|\boldsymbol{x}\|_1$, where each agent $i\in\{1,\dots,N\}$ stores the time-varying data set $(\boldsymbol{A}^i,\boldsymbol{b}_k^i)$, $\boldsymbol{A}^i\in\mathbb{R}^{m_i\times n}$, $\boldsymbol{b}_k^i\in\mathbb{R}^{m_i}$. Following a cloud-based learning approach, the goal is to solve this problem by relying on a central coordinator that receives and aggregates the results of local computations, without accessing the local data. Specifically, we reformulate the problem as (cf. [10, section 8.2])

$$\min_{\{\boldsymbol{x}^i\}_{i=1}^N,\boldsymbol{y}}\frac{1}{2}\sum_{i=1}^N\left\|\boldsymbol{A}^i\boldsymbol{x}^i-\boldsymbol{b}_k^i\right\|^2+\nu\|\boldsymbol{y}\|_1$$
$$\text{s.t. } \boldsymbol{x}^i=\boldsymbol{y},\quad i=1,\dots,N$$

where the N agents are tasked with processing the local data $(\boldsymbol{A}^i,\boldsymbol{b}_k^i)$ in order to update $\boldsymbol{x}^i$, and the central coordinator has the role of averaging $\boldsymbol{x}^i$ and enforcing sparsity with the ℓ_1-norm. This reformulation of the problem conforms to (31) and hence we can apply the prediction-correction ADMM discussed in Example 2.

The numerical results described below were derived as follows. The local matrices $\boldsymbol{A}^i$ were randomly generated so that $f_k^i(\boldsymbol{x}^i):=(1/2)\left\|\boldsymbol{A}^i\boldsymbol{x}^i-\boldsymbol{b}_k^i\right\|^2\in\mathcal{S}_{\mu,L}(\mathbb{R}^n)$, and $\boldsymbol{b}_k^i=\boldsymbol{A}^i\bar{\boldsymbol{x}}_k+\boldsymbol{e}_k^i$ where one third of $\bar{\boldsymbol{x}}_k$'s components are zero and the remaining change in a sinusoidal way, $\boldsymbol{e}_k^i$ is random normal noise with either medium variance 0.2, or low variance 0.002. We compared the performance of the one-step-back ADMM, the correction-only ADMM, and the prediction-correction ADMM. For the latter we use extrapolation of order $I=2,3$, as well as Taylor predictions based on backward finite-difference [43]. In particular, in this problem setting, extrapolation reads:

$$I=2:\nabla\hat{f}_{k+1}^i(\boldsymbol{x})=\boldsymbol{A}^i\boldsymbol{x}^i+2\boldsymbol{b}_k^i-\boldsymbol{b}_{k-1}^i, \tag{36}$$

$$I=3:\nabla\hat{f}_{k+1}^i(\boldsymbol{x})=\boldsymbol{A}^i\boldsymbol{x}^i+3\boldsymbol{b}_k^i-3\boldsymbol{b}_{k-1}^i+\boldsymbol{b}_{k-2}^i. \tag{37}$$

For Taylor with $O(T_s^2)$ and $O(T_s^3)$ backward finite-difference,

$$I=2:\nabla\hat{f}_{k+1}^i(\boldsymbol{x})=\boldsymbol{A}^i\boldsymbol{x}^i+2\boldsymbol{b}_k^i-\boldsymbol{b}_{k-1}^i, \tag{38}$$

$$I=3:\nabla\hat{f}_{k+1}^i(\boldsymbol{x})=\boldsymbol{A}^i\boldsymbol{x}^i+3\boldsymbol{b}_k^i-3\boldsymbol{b}_{k-1}^i+\boldsymbol{b}_{k-2}^i. \tag{39}$$

Note that Taylor with backward finite-difference is the same here as extrapolation, but this is not true in general. Finally, we report ZeaD [33] prediction results with $\zeta=1$ and $\zeta=2$:

$$\text{ZeaD}, I{=}3:\nabla\hat{f}_{k+1}(\boldsymbol{x}){=}\boldsymbol{A}^i\boldsymbol{x}^i{+}\frac{2\zeta+3}{2}\boldsymbol{b}_k^i{-}2\zeta\,\boldsymbol{b}_{k-1}^i{+}\frac{2\zeta-1}{2}\boldsymbol{b}_{k-2}^i. \tag{40}$$

Table 3 reports the asymptotic error of the compared approaches for different numbers of agents, each endowed with an equal number of data points from a total of $m=250$ ($m_i=m/N$), with the setting $P=10, C=2$. Similarly to the results of the previous sections, we observe that prediction-correction is in general better than prediction or correction alone. Different prediction strategies work better in different noise and number of agent settings. In this example, extrapolations of order 2 and 3 behave in par with the others, and sometimes marginally better. Additionally,

[8] In practice, we solve a *vectorized* version of this problem, which corresponds to a composite problem of the form (3), see [28, eq. (32)] for the details.

[9] https://www.met.ie/climate/available-data/historical-data

Table 2
Comparison of asymptotic errors observed in the numerical simulations for real weather data.

Method	Min	Mean ± Std	Max
Correction-only	5.293×10^{-3}	$7.393 \times 10^{-2} \pm 4.811 \times 10^{-3}$	3.616×10^{-1}
Taylor [28]	3.264×10^{-3}	$4.478 \times 10^{-2} \pm 2.945 \times 10^{-2}$	$2.490 \times 10^{-}$
Extrapolation ($I = 2$)	3.263×10^{-3}	$4.478 \times 10^{-2} \pm 2.945 \times 10^{-2}$	2.490×10^{-1}
Extrapolation ($I = 3$)	2.812×10^{-3}	$4.161 \times 10^{-2} \pm 2.691 \times 10^{-2}$	2.156×10^{-1}

Table 3
Comparison of asymptotic errors with respect to the true signal, for the online linear regression problem solved using ADMM, for different numbers of agents N. * this is equivalent to Taylor of the same order with backward finite-difference.

	$\sigma = 0.2$			$\sigma = 0.002$		
Method	$N = 1$	5	10	$N = 1$	5	10
One-step-back	1.37	0.50	0.28	1.06	0.38	0.36
Correction-only	**1.29**	0.63	0.20	1.07	0.50	0.45
Extrapolation* ($I = 2$)	1.37	**0.34**	**0.12**	**0.93**	0.16	0.13
Extrapolation* ($I = 3$)	1.37	0.42	0.15	**0.93**	**0.15**	**0.12**
ZeaD ($\zeta = 1, I = 3$)	1.37	0.38	0.15	**0.93**	0.16	0.13
ZeaD ($\zeta = 2, I = 3$)	1.35	0.48	0.18	**0.93**	**0.15**	**0.12**

we notice that a larger number of agents taking part in the solution of the problem can lead to small improvements in the asymptotic error. This is partly explained by observing that the costs f_k^i have a lower value of C_0 (the bound on the gradient's variation over time) than the cost defined on the whole data set, which leads to a lower asymptotic bound according to Theorem 2.

6.4. Online robotics

As a fourth example, we rework here the robotic setting considered in [7,14] In particular, we consider a number $N = 10$ of mobile robots that follow a leader robot while it moves in a 2D space. The problem can be formulated as,

$$\min_{\boldsymbol{x} \in \mathbb{R}^{2(N+1)}} f(\boldsymbol{x}; t_k) := \sum_{i=1}^{N} \frac{1}{2}\left(z_k^i - \boldsymbol{v}_i^\top \boldsymbol{x}^0\right)^2 + \frac{\lambda}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2, \tag{41}$$

$$\text{subject to } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{42}$$

where $\boldsymbol{x}^i \in \mathbb{R}^2$ is the position of robot $i = 0, \ldots, N$, where 0 represents the leader. The above problem amounts at estimating the position of the leader robot $\boldsymbol{x}^0$ based on local measurements z_k^i and a linear model $\boldsymbol{v}_i^\top$, with a suitable ℓ_2 regularization. In addition, the followers move as to maintain a rigid formation as imposed by the constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. All the details are given in [7].

We solve the above problem with a proximal gradient, by projecting over the constraint, employing several prediction and correction methods. In Fig. 4, we report the asymptotical tracking error varying the sampling time for a correction-only method, a simplified prediction [25], two extrapolation-based predictions of 2nd and 3rd order, respectively, as well as a ZeaD prediction of third order with ($\zeta = 1$). In all cases, the number of proximal gradients are $P = 20$ for prediction and $C = 5$ for correction.

As one can appreciate, the extrapolation methods achieve the theoretical order of $O(T_s^2)$ and $O(T_s^3)$ and do very well with respect to other prediction methods (simplified of second order, and ZeaD of third order), further advocating for this prediction modality.

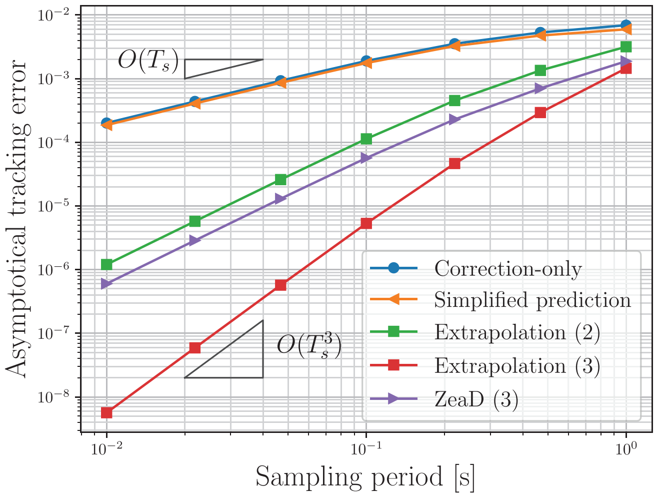

Fig. 4. Comparison of several methods to solve an online robotics problem in terms of the asymptotical tracking error *vs.* the sampling time.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Nicola Bastianello: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Ruggero Carli:** Supervision. **Andrea Simonetto:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing, Supervision.

Data availability

The data used is from a database available online

Appendix A. Proofs of Section 4

A1. Proof of Proposition 1

Consider a prediction-correction strategy where we apply N_P and N_C steps during prediction and correction, respectively. By Lemma 1, the following holds:

$$\left\|\hat{\boldsymbol{x}}_{k+1} - \hat{\boldsymbol{x}}_{k+1}^*\right\| \le \zeta(N_\mathrm{P})\left\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k+1}^*\right\| \tag{A.1a}$$

$$\left\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_{k+1}^*\right\| \le \zeta(N_\mathrm{C})\left\|\hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_{k+1}^*\right\|. \tag{A.1b}$$

The goal now is to bound the prediction error $\left\|\hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_{k+1}^*\right\|$. If $N_\mathrm{P} = 0$ then no prediction steps are applied, and thus, using the triangle inequality, we can write:

$$\begin{aligned}\left\|\hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_{k+1}^*\right\| = \left\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}^*\right\| \le \left\|\boldsymbol{x}_k - \boldsymbol{x}_k^*\right\| + \left\|\boldsymbol{x}_{k+1}^* - \boldsymbol{x}_k^*\right\| \le \left\|\boldsymbol{x}_k - \boldsymbol{x}_k^*\right\| + \sigma_k \\ = \zeta(N_\mathrm{P})\left\|\boldsymbol{x}_k - \boldsymbol{x}_k^*\right\| + \zeta(N_\mathrm{P})\sigma_k + \xi(N_\mathrm{P})\tau_k\end{aligned}$$

where we used the facts that $\zeta(N_\mathrm{P}) = 1$ and $\xi(N_\mathrm{P}) = 0$ if $N_\mathrm{P} = 0$ to derive the last equality (cf. Lemma 1). Consider now the case of $N_\mathrm{P} > 0$. By the triangle inequality and the early termination inequality (A.1a), the following chain of inequalities holds:

$$\left\|\hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_{k+1}^*\right\| \le \left\|\hat{\boldsymbol{x}}_{k+1} - \hat{\boldsymbol{x}}_{k+1}^*\right\| + \left\|\hat{\boldsymbol{x}}_{k+1}^* - \boldsymbol{x}_{k+1}^*\right\| \le \zeta(N_\mathrm{P})\left\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k+1}^*\right\| + \tau_k$$

$$\begin{aligned}
&\le \zeta(N_{\mathrm{P}})\Big(\left\|\boldsymbol{x}_k - \boldsymbol{x}_k^*\right\| + \left\|\boldsymbol{x}_k^* - \boldsymbol{x}_{k+1}^*\right\| + \left\|\boldsymbol{x}_{k+1}^* - \hat{\boldsymbol{x}}_{k+1}^*\right\|\Big) + \tau_k \\
&\le \zeta(N_{\mathrm{P}})\left\|\boldsymbol{x}_k - \boldsymbol{x}_k^*\right\| + \zeta(N_{\mathrm{P}})\sigma_k + (1 + \zeta(N_{\mathrm{P}}))\tau_k \\
&= \zeta(N_{\mathrm{P}})\left\|\boldsymbol{x}_k - \boldsymbol{x}_k^*\right\| + \zeta(N_{\mathrm{P}})\sigma_k + \xi(N_{\mathrm{P}})\tau_k
\end{aligned}$$

where the last equality follows by the fact that $\xi(N_{\mathrm{P}}) = 1 + \zeta(N_{\mathrm{P}})$ if $N_{\mathrm{P}} > 0$ (cf. Lemma 1). Therefore for any $N_{\mathrm{P}} \ge 0$ we can bound the prediction error as

$$\left\|\hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_{k+1}^*\right\| \le \zeta(N_{\mathrm{P}})\left\|\boldsymbol{x}_k - \boldsymbol{x}_k^*\right\| + \zeta(N_{\mathrm{P}})\sigma_k + \xi(N_{\mathrm{P}})\tau_k \tag{A.2}$$

and combining (A.2) with (A.1b) for the correction step yields

$$\left\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_{k+1}^*\right\| \le \zeta(N_{\mathrm{C}})\zeta(N_{\mathrm{P}})\left\|\boldsymbol{x}_k - \boldsymbol{x}_k^*\right\| + \zeta(N_{\mathrm{C}})(\zeta(N_{\mathrm{P}})\sigma_k + \xi(N_{\mathrm{P}})\tau_k), \tag{A.3}$$

from which the thesis. □

A2. A supporting result

Theorem 3. *Let $f \in \mathcal{S}_{\mu,L}(\mathbb{R}^n)$ and $g \in \Gamma_0(\mathbb{R}^n)$, then the solution mapping $S(\boldsymbol{p}) = \{\boldsymbol{y} \mid \nabla_{\boldsymbol{x}} f(\boldsymbol{y}) + \partial g(\boldsymbol{y}) \ni \boldsymbol{p}\}$ of the parameterized generalized equation $\nabla_{\boldsymbol{x}} f(\boldsymbol{y}) + \partial g(\boldsymbol{y}) \ni \boldsymbol{p}$ is single-valued and μ^{-1}-Lipschitz continuous.*

Proof. The proof follows from [29, Theorem 1]. □

A3. Proof of Lemma 2

The following proof is an extension of [16, Theorem 2F.10] when $\nabla_{t\boldsymbol{x}} f$ exists everywhere. First, we define the auxiliary functions: $\Psi(\boldsymbol{y}) = \nabla_{\boldsymbol{x}} f_{k+1}(\boldsymbol{y}) + \partial g(\boldsymbol{y})$ and $\psi(\boldsymbol{y}) = \nabla_{\boldsymbol{x}} f_k(\boldsymbol{y}) - \nabla_{\boldsymbol{x}} f_{k+1}(\boldsymbol{y})$ and, by the fact that $\nabla_{\boldsymbol{x}} f_k(\boldsymbol{x}_k^*) + \partial g(\boldsymbol{x}_k^*) \ni \boldsymbol{0}$ and $\nabla_{\boldsymbol{x}} f_{k+1}(\boldsymbol{x}_{k+1}^*) + \partial g(\boldsymbol{x}_{k+1}^*) \ni \boldsymbol{0}$, we have $(\psi + \Psi)(\boldsymbol{x}_{k+1}^*) \ni \psi(\boldsymbol{x}_{k+1}^*)$. We define now the function $F(\boldsymbol{y}) = (\psi + \Psi)(\boldsymbol{y})$ and consider the parametric generalized equation $F(\boldsymbol{y}) + \boldsymbol{p} \ni \boldsymbol{0}$. Under Assumptions 1 and 2, Theorem 3 implies that the solution mapping $\boldsymbol{p} \mapsto \boldsymbol{y}(\boldsymbol{p})$ for this generalized equation is everywhere single valued and Lipschitz continuous with constant μ^{-1}, *i.e.* $\left\|\boldsymbol{y}(\boldsymbol{p}) - \boldsymbol{y}(\boldsymbol{p}')\right\| \le \left\|\boldsymbol{p} - \boldsymbol{p}'\right\|/\mu$. Therefore, setting $\boldsymbol{p} = \boldsymbol{0}$ and $\boldsymbol{p}' = -\psi(\boldsymbol{x}_{k+1}^*)$, implies

$$\left\|\boldsymbol{x}_{k+1}^* - \boldsymbol{x}_k^*\right\| \le \left\|\psi(\boldsymbol{x}_{k+1}^*)\right\|/\mu \le C_0 T_{\mathrm{s}}/\mu.$$

where we used the fact that: $\left\|\psi(\boldsymbol{x}_{k+1}^*)\right\| = \left\|\nabla_{\boldsymbol{x}} f_k(\boldsymbol{x}_{k+1}^*) - \nabla_{\boldsymbol{x}} f_{k+1}(\boldsymbol{x}_{k+1}^*)\right\| \le C_0 T_{\mathrm{s}}$, see [41, eq. (59)]. □

A4. Proof of Lemma 3

Define the functions $\Psi(\boldsymbol{y}) = \nabla_{\boldsymbol{x}} f_{k+1}(\boldsymbol{y}) + \partial g(\boldsymbol{y})$ and $\psi(\boldsymbol{y}) = \nabla_{\boldsymbol{x}} \hat{f}_{k+1}(\boldsymbol{y}) - \nabla_{\boldsymbol{x}} f_{k+1}(\boldsymbol{y})$; by the optimality conditions of the correction and prediction problems we have that $(\Psi + \psi)(\boldsymbol{x}_{k+1}^*) \ni \psi(\boldsymbol{x}_{k+1}^*)$ and $(\Psi + \psi)(\hat{\boldsymbol{x}}_{k+1}^*) \ni \boldsymbol{0}$. Then, applying Theorem 3 to the parametrized generalized equation $(\Psi + \psi)(\boldsymbol{y}) \ni \boldsymbol{p}$ we have the following bound $\left\|\hat{\boldsymbol{x}}_{k+1}^* - \boldsymbol{x}_{k+1}^*\right\| \le \left\|\psi(\boldsymbol{x}_{k+1}^*)\right\|/\mu$. By the interpolation error formula (19), we have the bound:

$$\begin{aligned}
\left\|\psi(\boldsymbol{x}_{k+1}^*)\right\| \quad &= \left\|\nabla_{\boldsymbol{x}} \hat{f}_{k+1}(\boldsymbol{x}_{k+1}^*) - \nabla_{\boldsymbol{x}} f_{k+1}(\boldsymbol{x}_{k+1}^*)\right\| \\
&\le \left\|\tfrac{1}{I!}\tfrac{\partial^{(I)}}{\partial t^{(I)}}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_{k+1}^*; \tau)\omega_I(t_{k+1})\right\| \le C(I)T_{\mathrm{s}}^I
\end{aligned} \tag{A.4}$$

where $\tau \in [t_{k-1}, t_{k+1}]$, and we used the facts that $\omega_I(t_{k+1}) = \prod_{i=1}^{I}(t_{k+1} - t_{k+1-i}) = I!T_{\mathrm{s}}^I$ (cf. (19)) and (22) to derive the last inequality. □

A5. Proof of Theorem 1

By Lemmas 2 and 3 we know that there exist $\sigma, \tau \in [0, +\infty)$ such that $\left\|\boldsymbol{x}_{k+1}^* - \boldsymbol{x}_k^*\right\| \le \sigma$ and $\left\|\hat{\boldsymbol{x}}_{k+1}^* - \boldsymbol{x}_{k+1}^*\right\| \le \tau$. As such, Proposition 1 holds. We can then use Equation (A.3) with our bounds for σ, τ.

If then, we choose N_{P} and N_{C} such that $\zeta(N_{\mathrm{P}})\zeta(N_{\mathrm{C}}) < 1$, then the error converges and using the geometric series the thesis of Theorem 1 follows. □

Appendix B. Proofs of Section 5

B1. Proof of Lemma 4

Notice that $\bar{\boldsymbol{x}}(\boldsymbol{w}, t)$ is the unique solution to the equation $\psi(\boldsymbol{x}; \boldsymbol{w}, t) := \nabla_{\boldsymbol{x}} f(\boldsymbol{x}; t) - \boldsymbol{A}^\top \boldsymbol{w} = \boldsymbol{0}$, where $\psi(\boldsymbol{x}; \boldsymbol{w}, t)$ is differentiable in $\boldsymbol{x}$ with $\nabla_{\boldsymbol{x}}\psi(\boldsymbol{x}; \boldsymbol{w}, t) = \nabla_{\boldsymbol{xx}} f(\boldsymbol{x}; t)$ non-singular. Now set $\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}}(\boldsymbol{w}, t)$ to simplify the notation. Fixing $\boldsymbol{w}$ and applying [16, Theorem 1.B.1] w.r.t. t gives $\partial\bar{\boldsymbol{x}}/\partial t = -[\nabla_{\boldsymbol{x}}\psi(\bar{\boldsymbol{x}}; \boldsymbol{w}, t)]^{-1}\nabla_t\psi(\bar{\boldsymbol{x}}; \boldsymbol{w}, t) = -\nabla_{\boldsymbol{xx}} f(\bar{\boldsymbol{x}}; t)^{-1}\nabla_{t\boldsymbol{x}} f(\bar{\boldsymbol{x}}; t)$, and as a consequence, we have

$$\nabla_{t\boldsymbol{w}} d^f(\boldsymbol{w}; t) = -\boldsymbol{A}\nabla_{\boldsymbol{xx}} f(\bar{\boldsymbol{x}}(\boldsymbol{w}, t); t)^{-1}\nabla_{t\boldsymbol{x}} f(\bar{\boldsymbol{x}}(\boldsymbol{w}, t); t). \tag{B.1}$$

Using (B.1) and the sub-multiplicativity of the norm we have

$$\begin{aligned}
\left\|\nabla_{t\boldsymbol{w}} d^f(\boldsymbol{w}; t)\right\| &= \left\|\boldsymbol{A}\nabla_{\boldsymbol{xx}} f(\bar{\boldsymbol{x}}(\boldsymbol{w}, t); t)^{-1}\nabla_{t\boldsymbol{x}} f(\bar{\boldsymbol{x}}(\boldsymbol{w}, t); t)\right\| \\
&\le \|\boldsymbol{A}\|\left\|\nabla_{\boldsymbol{xx}} f(\bar{\boldsymbol{x}}(\boldsymbol{w}, t); t)^{-1}\right\|\|\nabla_{t\boldsymbol{x}} f(\bar{\boldsymbol{x}}(\boldsymbol{w}, t); t)\| \\
&\le \|\boldsymbol{A}\|C_0/\mu
\end{aligned}$$

where the last inequality holds by Assumption 4 (i). □

B2. Proof of Theorem 2

As observed in Section 5.1, under Assumption 4 the dual cost d_k^f is $\bar{\mu}$-strongly convex and $\bar{L}$-smooth, and $d_k^h \in \Gamma_0(\mathbb{R}^p)$. Therefore, we can follow the same derivation in Appendix A.5 to show that (A.3) holds for the dual problem, with

$$\left\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_{k+1}^*\right\| \le \zeta(N_{\mathrm{C}})\zeta(N_{\mathrm{P}})\left\|\boldsymbol{w}_k - \boldsymbol{w}_k^*\right\| + \zeta(N_{\mathrm{C}})(\zeta(N_{\mathrm{P}})\bar{\sigma} + \xi(N_{\mathrm{P}})\bar{\tau}). \tag{B.2}$$

The goal now is to provide a bound to both $\bar{\sigma}$ and $\bar{\tau}$. First, since Lemma 4 holds, we can apply Lemma 2 to prove that

$$\left\|\boldsymbol{w}_{k+1}^* - \boldsymbol{w}_k^*\right\| \le \bar{C}_0 T_{\mathrm{s}}/\bar{\mu} =: \bar{\sigma}.$$

To bound $\bar{\tau}$, following the derivation in Appendix A.3 we can see that

$$\left\|\hat{\boldsymbol{w}}_{k+1}^* - \boldsymbol{w}_{k+1}^*\right\| \le \left\|\psi(\boldsymbol{w}_{k+1}^*)\right\|/\bar{\mu}$$

where $\psi(\boldsymbol{w}) = \nabla_{\boldsymbol{w}}\hat{d}_{k+1}^f(\boldsymbol{w}) - \nabla_{\boldsymbol{w}} d_{k+1}^f(\boldsymbol{w})$. Using (30) we further know that $\nabla_{\boldsymbol{w}} d_{k+1}^f(\boldsymbol{w}_{k+1}^*) = \boldsymbol{A}\bar{\boldsymbol{x}} - \boldsymbol{c}$ and $\nabla_{\boldsymbol{w}}\hat{d}_{k+1}^f(\boldsymbol{w}_{k+1}^*) = \boldsymbol{A}\bar{\bar{\boldsymbol{x}}} - \boldsymbol{c}$, with $\bar{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} F_{k+1}(\boldsymbol{x})$ and $\bar{\bar{\boldsymbol{x}}} = \arg\min_{\boldsymbol{x}}\hat{F}_{k+1}(\boldsymbol{x})$, having defined

$$F_{k+1}(\boldsymbol{x}) = f_{k+1}(\boldsymbol{x}) - \langle\boldsymbol{A}^\top\boldsymbol{w}_{k+1}^*, \boldsymbol{x}\rangle,$$

$$\hat{F}_{k+1}(\boldsymbol{x}) = \hat{f}_{k+1}(\boldsymbol{x}) - \langle\boldsymbol{A}^\top\boldsymbol{w}_{k+1}^*, \boldsymbol{x}\rangle = \sum_{i=1}^{I}\ell_i F_{k+1-i}(\boldsymbol{x}).$$

Using the sub-multiplicativity of the norm we have $\left\|\psi(\boldsymbol{w}_{k+1}^*)\right\| \le \|\boldsymbol{A}\|\left\|\bar{\boldsymbol{x}} - \bar{\bar{\boldsymbol{x}}}\right\|$, and we need to bound $\left\|\bar{\boldsymbol{x}} - \bar{\bar{\boldsymbol{x}}}\right\|$.

Defining $\Gamma(\boldsymbol{y}) := \nabla_{\boldsymbol{x}} F_{k+1}(\boldsymbol{y})$ and $\gamma(\boldsymbol{y}) := \nabla_{\boldsymbol{x}}\hat{F}_{k+1}(\boldsymbol{y}) - \nabla_{\boldsymbol{x}} F_{k+1}(\boldsymbol{y})$, we can see that $\bar{\boldsymbol{x}}$ and $\bar{\bar{\boldsymbol{x}}}$ are the solutions of the generalized equation $(\Gamma + \gamma)(\boldsymbol{y}) = \boldsymbol{p}$ when $\boldsymbol{p} = \boldsymbol{0}$ and $\boldsymbol{p} = \gamma(\bar{\bar{\boldsymbol{x}}})$. Therefore, applying the inverse function theorem [16, Theorem 1A.1] we have $\left\|\bar{\boldsymbol{x}} - \bar{\bar{\boldsymbol{x}}}\right\| \le \left\|\gamma(\bar{\bar{\boldsymbol{x}}})\right\|/\mu$. Finally, we have

$$\left\|\gamma(\bar{\bar{\boldsymbol{x}}})\right\| = \left\|\nabla_{\boldsymbol{x}}\hat{F}_{k+1}(\bar{\boldsymbol{x}}) - \nabla_{\boldsymbol{x}} F_{k+1}(\bar{\boldsymbol{x}})\right\| = \left\|\nabla_{\boldsymbol{x}}\hat{f}_{k+1}(\bar{\boldsymbol{x}}) - \nabla_{\boldsymbol{x}} f_{k+1}(\bar{\boldsymbol{x}})\right\| \le C(I)T_{\mathrm{s}}^I$$

where the inequality holds by (A.4).

Putting everything together yields the prediction error bound

$$\left\| \hat{\boldsymbol{w}}_{k+1}^* - \boldsymbol{w}_{k+1}^* \right\| \leq \|\boldsymbol{A}\| C(I) T_s^I / (\mu \bar{\mu}) =: \bar{\tau}$$

and substituting into (B.2) yields the dual convergence bound.

The primal convergence bound can then be derived as a consequence of (B.2) by using the fact that [8, Lemma A.1]

$$\begin{aligned} \left\| \boldsymbol{x}_{k+1} - \boldsymbol{x}_{k+1}^* \right\| &\leq \|\boldsymbol{A}\|/\mu \left\| \boldsymbol{w}_{k+1} - \boldsymbol{w}_{k+1}^* \right\|, \\ \left\| \boldsymbol{B}(\boldsymbol{y}_{k+1} - \boldsymbol{y}_{k+1}^*) \right\| &\leq \|\boldsymbol{B}\| (1/\rho + \|\boldsymbol{A}\|^2/\mu) \left\| \boldsymbol{w}_{k+1} - \boldsymbol{w}_{k+1}^* \right\| \ \text{(in case } h \not\equiv 0\text{)}. \end{aligned}$$

□

References

[1] E.L. Allgower, K. Georg, Numerical continuation methods: An introduction, Springer-Verlag, 1990.

[2] D. Angelosante, J.A. Bazerque, G.B. Giannakis, Online adaptive estimation of sparse signals: where RLS meets the ℓ_1-norm, IEEE Trans. Signal Process. 58 (2010) 3436–3447.

[3] M.S. Asif, J. Romberg, Sparse recovery of streaming signals using ℓ_1-homotopy, IEEE Trans. Signal Process. 62 (2014) 4209–4223.

[4] A. Balavoine, J. Romberg, C. Rozell, Discrete and continuous iterative soft thresholding with a dynamic input, IEEE Trans. Signal Process. 63 (2015) 3165–3176.

[5] N. Bastianello, Tvopt: a python framework for time-varying optimization, in: 2021 60th IEEE Conference on Decision and Control (CDC), 2021, pp. 227–232.

[6] N. Bastianello, R. Carli, L. Schenato, M. Todescato, Asynchronous distributed optimization over lossy networks via relaxed ADMM: stability and linear convergence, IEEE Trans. Automat. Contr. 66 (2021) 2620–2635.

[7] N. Bastianello, A. Simonetto, R. Carli, Prediction-correction splittings for nonsmooth time-varying optimization, in: 2019 18th European Control Conference (ECC), IEEE, Naples, Italy, 2019, pp. 1963–1968.

[8] N. Bastianello, A. Simonetto, R. Carli, Primal and dual prediction-correction methods for time-varying convex optimization, arXiv:2004.11709 [cs, math] (2020). http://arxiv.org/abs/2004.11709.

[9] H.H. Bauschke, P.L. Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Spaces, 2 edition, Springer, Cham, 2017. CMS books in mathematics.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trend. Mach. Learn. 3 (2010) 1–122, doi:10.1561/2200000016.

[11] F.S. Cattivelli, C.G. Lopes, A.H. Sayed, Diffusion recursive least-squares for distributed estimation over adaptive networks, IEEE Trans. Signal Process. 56 (2008) 1865–1877.

[12] A.S. Charles, A. Balavoine, C.J. Rozell, Dynamic filtering of time-varying sparse signals via ℓ_1 minimization, IEEE Trans. Signal Process. 64 (2016) 5644–5656.

[13] E. Dall'Anese, A. Simonetto, S. Becker, L. Madden, Optimization and learning with information streams: time-varying algorithms and applications, IEEE Signal Process. Mag. 37 (2020) 71–83.

[14] R. Dixit, A.S. Bedi, R. Tripathi, K. Rajawat, Online learning with inexact proximal online gradient descent algorithms, IEEE Trans. Signal Process. 67 (2019) 1338–1352.

[15] A.L. Dončev, M.I. Krastanov, R.T. Rockafellar, V.M. Veliov, An euler–newton continuation method for tracking solution trajectories of parametric variational inequalities, SIAM J. Control Optim. 51 (2013) 1823–1840.

[16] A.L. Dončev, R.T. Rockafellar, Implicit Functions and Solution Mappings: A view from Variational Analysis, 2 edition, Springer, New York, NY Heidelberg Dordrecht, 2014. Springer series in operations research and financial engineering.

[17] P. Giselsson, S. Boyd, Metric selection in fast dual forward-backward splitting, Automatica 62 (2015) 1–10.

[18] P. Giselsson, S. Boyd, Linear convergence and metric selection for douglas-rachford splitting and ADMM, IEEE Trans. Automat. Contr. 62 (2017) 532–544.

[19] J. Guddat, F. Guerra Vazquez and H. T. Jongen, Parametric Optimization: Singularities, Pathfollowing and Jumps, John Wiley & Sons, Chichester, UK, 1990.

[20] E.C. Hall, R.M. Willett, Online convex optimization in dynamic environments, IEEE J. Sel. Top. Signal Process. 9 (2015) 647–662.

[21] T.H. Hamam, J. Romberg, Streaming solutions for time-varying optimization problems, IEEE Trans. Signal Process. 70 (2022) 3582–3597.

[22] J.H. Hours, C.N. Jones, A parametric nonconvex decomposition algorithm for real-time and distributed NMPC, IEEE Trans. Automat. Contr. 61 (2016) 287–302.

[23] F.Y. Jakubiec, A. Ribeiro, D-MAP: distributed maximum a posteriori probability estimation of dynamic systems, IEEE Trans. Signal Process. 61 (2013) 450–466.

[24] V. Kungurtsev, J. Jäschke, A prediction-correction path-following algorithm for dual-degenerate parametric optimization problems, SIAM J. Optim. 27 (2017) 538–564.

[25] Z. Lin, F. Chen, L. Xiang, G. Guo, A simplified prediction-correction algorithm for time-varying convex optimization, in: 2019 Chinese Control Conference (CCC), 2019, pp. 1989–1994.

[26] Q. Ling, A. Ribeiro, Decentralized dynamic optimization through the alternating direction method of multipliers, IEEE Trans. Signal Process. 62 (2014) 1185–1197.

[27] J.J. Moreau, Evolution problem associated with a moving convex set in a hilbert space, J. Differ. Equ. 26 (1977) 347–374.

[28] A. Natali, E. Isufi, M. Coutino, G. Leus, Learning time-varying graphs from online data, IEEE Open J. Signal Process. 3 (2022) 212–228.

[29] Y. Nesterov, Smooth minimization of non-smooth functions, Math. Program. 103 (2005) 127–152.

[30] S. Paternain, M. Morari, A. Ribeiro, A prediction-correction algorithm for real–time model predictive control, arXiv preprint arXiv:1911.10051 (2019).

[31] B.T. Polyak, Introduction to Optimization, Optimization Software, Inc., 1987.

[32] F. Potra, On q-order and r-order of convergence, J. Optim. Theory Appl. 63 (1989) 415–431.

[33] Z. Qi, Y. Zhang, New models for future problems solving by using ZND method, correction strategy and extrapolation formulas, IEEE Access 7 (2019) 84536–84544.

[34] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Texts in Applied Mathematics, 2nd ed, Springer, Berlin ; New York, 2007. Number 37

[35] S.M. Robinson, Strongly regular generalized equations, Math. Oper. Res. 5 (1980) 43–62.

[36] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Control Optim. 14 (1976) 877–898.

[37] R.T. Rockafellar, R.J.B. Wets, Variational Analysis, Die Grundlehren der Mathematischen Wissenschaften in Einzeldarstellungen, 3, Springer, Dordrecht, 2009. Number 317

[38] E.K. Ryu, S. Boyd, A primer on monotone operator methods, Appl. Comput. Math. 15 (2016) 3–43.

[39] S. Shalev-Shwartz, Online learning and online convex optimization, Found. Trend. Mach. Learn. 4 (2011) 107–194.

[40] A. Simonetto, Dual prediction–correction methods for linearly constrained time-varying convex programs, IEEE Trans. Automat. Contr. 64 (2019) 3355–3361.

[41] A. Simonetto, E. Dall'Anese, Prediction-correction algorithms for time-varying constrained optimization, IEEE Trans. Signal Process. 65 (2017) 5481–5494.

[42] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, G.B. Giannakis, Time-varying convex optimization: time-structured algorithms and applications, Proc. IEEE (to appear) (2020).

[43] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, A. Ribeiro, A class of prediction–correction methods for time-varying convex optimization, IEEE Trans. Signal Process. 64 (2016) 4576–4591.

[44] A. Taylor, Convex Interpolation and Performance Estimation of First-Order Methods for Convex Optimization, 2017 Ph.D. thesis. Ph.D. thesis. Université catholique de Louvain

[45] N. Vaswani, J. Zhan, Recursive recovery of sparse signal sequences from compressive measurements: a review, IEEE Trans. Signal Process. 64 (2016) 3523–3549.

[46] Y. Yang, M. Zhang, M. Pesavento, D.P. Palomar, An online parallel and distributed algorithm for recursive estimation of sparse signals, IEEE Trans. Signal Inf. Process. Netw. 2 (2016) 290–305.

[47] V.M. Zavala, M. Anitescu, Real-time nonlinear optimization as a generalized equation, SIAM J. Control Optim. 48 (2010) 5444–5467.