Automatica 185 (2026) 112787

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Online learning of nonlinear parametric models under non-smooth regularization using EKF and ADMM☆

Lapo Frascati [a,*], Alberto Bemporad [b]

[a] ODYS S.r.l., 20159, Milan, Italy
[b] IMT School for Advanced Studies Lucca, 55100, Lucca, Italy

ARTICLE INFO

Article history:
Received 26 February 2025
Received in revised form 22 July 2025
Accepted 7 November 2025
Available online 19 December 2025

Keywords:
Kalman filtering
Non-smooth regularization
Online learning
Parameter estimation
Adaptive control
Neural networks

ABSTRACT

This paper proposes a novel combination of extended Kalman filtering (EKF) with the alternating direction method of multipliers (ADMM) for learning parametric nonlinear models online under non-smooth regularization terms, including ℓ_1 and ℓ_0 penalties and bound constraints on model parameters. For the case of linear time-varying models and non-smooth convex regularization terms, we provide a sublinear regret bound that ensures the proper behavior of the online learning strategy. The approach is computationally efficient for a wide range of regularization terms, which makes it appealing for its use in embedded control applications for online model adaptation. We show the performance of the proposed method in three simulation examples, highlighting its effectiveness compared to other batch and online algorithms.

1. Introduction

Online learning of nonlinear parametric models is of paramount importance in several domains, including model-based adaptive control and real-time estimation of unmeasured variables. Typically, parametric models derived from physics (Schoukens & Ljung, 2019) or black-box (Pillonetto et al., 2025) structures are identified offline on training data, then directly deployed and used without any further updates. On the other hand, further adapting the model online can significantly improve its predictive capabilities (Akpan & Hassapis, 2011), especially when the phenomenon we are modeling changes over time, and allows for smaller model structures that adapt to varying operating conditions, unlike single, overall models trained offline to cover all conditions.

A vast literature exists for online learning (Hoi, Sahoo, Lu, & Zhao, 2021), and several approaches using first or second order gradient information are suitable for real-time model adaptation. Examples of algorithms exploiting first-order information are online gradient descent (OGD) and its regularized alternatives, i.e., follow the regularized leader (FTRL) and online mirror descent (OMD) (McMahan, 2017), which are usually fast at execution but slow at convergence. Second-order gradient information can be used to improve the convergence speed, such as in the online Newton step (ONS), or in the well known extended Kalman filter (EKF) (Kalman, 1960) which has been proved to be very effective in online model adaptation by treating the parameters as constant states to be estimated (Abulikemu & Changliu, 2020; Bemporad, 2021).

While FTRL and OMD, as well as modified versions of the standard EKF (Bemporad, 2021), can effectively deal with *smooth* regularization terms, *non-smooth regularizers* are often required when learning models. Non-smooth ℓ_0 or ℓ_1 penalties are used to induce sparsity in the model (Van de Geer, 2010), and group-Lasso terms to remove entire parts of the model, such as neurons of a neural network (Scardapane, Comminiello, Hussain, & Uncini, 2017). Indeed, obtaining compact models is particularly important for embedded applications, such as model predictive control, where the real-time numerical complexity of the controller depends directly on the complexity of the prediction model. In addition, indicator functions of feasible sets are another example of non-smooth penalties that are used to impose constraints on model coefficients, such as known bounds on certain unknown physical parameters (Boyd, Parikh, Chu, Peleato, & Eckstein, 2011).

Several approaches have been proposed for online learning under non-smooth regularization. Online ADMM (alternating direction method of multipliers) (Wang & Banjaree, 2012) allows handling quite general non-smooth regularizers, while EKF can be modified to deal with ℓ_1-regularization by treating it as a special limit case of a smooth regularizer (Bemporad, 2021). While the main limitation of online ADMM is its convergence

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Matthias A. Muller under the direction of Editor Alessandro Chiuso.

* Corresponding author.
E-mail addresses: lapo.frascati@odys.it (L. Frascati), alberto.bemporad@imtlucca.it (A. Bemporad).

https://doi.org/10.1016/j.automatica.2025.112787

speed, the main limitation of EKF is that it is not directly suitable for dealing with general non-smooth regularization terms, such as ℓ_0 regularization, group-Lasso penalties, and indicator functions of feasible sets. The main contribution of this work is to show interesting connections and similarities between these two approaches, developing an extension of EKF that inherits the advantages from both, i.e., the convergence speed of EKF and the capability to handle general non-smooth regularization terms of online-ADMM.

The proposed method consists of a simple and computationally efficient modification of the EKF algorithm by intertwining updates based on online measurements and output prediction errors with updates related to ADMM iterations. This modification allows EKF to deal with a broad class of non-smooth regularization terms for which ADMM is applicable, including ℓ_0/ℓ_1 penalties, group-Lasso and indicator functions of simple sets. For linear time-varying models and convex regularization terms, we provide a sublinear regret bound that proves the proper behavior of the resulting online learning strategy. The proposed method is computationally efficient and numerically robust, making it especially appealing for embedded adaptive control applications.

The rest of the paper is organized as follows. Section 2 gives a quick introduction to the use of EKF for online model learning, setting the background for the proposed ADMM+EKF approach described in Section 3. In Section 4, we prove a sublinear regret bound for the proposed approach in the convex linear case. Simulation results are shown in Section 5 and conclusions are drawn in Section 6.

1.1. Notation

Given a matrix $A \in \mathbb{R}^{m\times n}$, $A_{i:}$ denotes the ith row of A, $A_{:j}$ its jth column, A_{ij} its (i,j)th entry. Given a vector $v \in \mathbb{R}^n$ we denote by v^m a measurement of v, by $\|v\|_1 = \sum_{i=1}^n |v_i|$ the 1-norm of v and by $\|v\|_0$ its 0-norm, which is defined as the number of non-zero elements in the vector. Given a symmetric positive semidefinite matrix $P = P^T \succeq 0$, $P \in \mathbb{R}^{n\times n}$, we denote by $\|v\|_P^2$ the quadratic form $v^T P v$. Further, we denote by $\hat{v}_{i|j}$ the estimate of vector v at instant i given all information up to instant j, and by $P_{i|j}$ the corresponding covariance matrix. $v \sim \mathcal{N}(\mu, P)$ and $v \sim \mathcal{U}_{[v_{min}, v_{max}]}$ denote that v was randomly generated from a normal distribution with mean μ and covariance P or from a uniform distribution in the interval $[v_{min}, v_{max}]$, respectively.

2. EKF for online model learning

Given a dataset (z_k^m, y_k^m), $z \in \mathbb{R}^{n_z}$, $y \in \mathbb{R}^{n_y}$, $k = 0, 1, \ldots, N-1$, our goal is to recursively estimate a nonlinear parametric model

$$y = h(k, z; x) \tag{1}$$

which describes the (possibly time-varying) relationship between the input signal z_k^m and the output signal y_k^m. In (1), $x \in \mathbb{R}^{n_x}$ is the vector of parameters to be learned, such as the weights of a feedforward neural network mapping z into y, or the coefficients of a nonlinear autoregressive model, with y representing the current output and z a vector of past inputs and outputs of a dynamical system. In order to estimate x and capture its possible time-varying nature, we consider the nonlinear dynamical model

$$x_{k+1} = x_k + q_k, \quad y_k^{nl} = h_k(x_k) + r_k \tag{2}$$

where $h_k(\cdot) = h(k, z_k; \cdot)$ and we assume $h_k : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ differentiable for all k, $y_k^{nl} \in \mathbb{R}^{n_y}$, and $r_k \sim \mathcal{N}(0, R_k)$, $q_k \sim \mathcal{N}(0, Q_k)$ are the measurement and process noise that we introduce to model, respectively, measurement errors and variations of the model parameters over time, with corresponding covariance matrices $R_k = R_k' \succ 0$, $Q_k = Q_k' \succ 0$. By linearizing model (2) around a value $\bar{x}_k$ of the parameter vector, i.e., by approximating $h_k(x_k) \approx h_k(\bar{x}_k) + C_k(x_k - \bar{x}_k)$, $C_{k,i:} = \nabla_x h_{ki}(\bar{x}_k)'$, $i = 1, \ldots, n_y$, we obtain the linear time-varying model

$$x_{k+1} = x_k + q_k, \quad y_k = C_k x_k + r_k \tag{3}$$

with $y_k = y_k^{nl} - h_k(\bar{x}_k) + C_k \bar{x}_k$. The classical Kalman filter (Kalman, 1960) can be used to estimate the state in (3), i.e., to learn the parameters x_k recursively. Given $\hat{x}_{0|-1}, P_{0|-1}$ we perform the following iterations for $k = 0, \ldots, N-1$:

$$\begin{aligned} P_{k|k}^{-1} &= P_{k|k-1}^{-1} + C_k^T R_k^{-1} C_k \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + P_{k|k} C_k^T R_k^{-1} (y_k^m - C_k \hat{x}_{k|k-1}) \\ P_{k+1|k}^{-1} &= (Q_k + P_{k|k})^{-1} \\ \hat{x}_{k+1|k} &= \hat{x}_{k|k} \end{aligned} \tag{4}$$

with $\bar{x}_k = \hat{x}_{k|k-1}$. The first two updates in (4) are usually referred to as the correction step and the last two as the prediction step. Note that (4) is an EKF for model (2), since C_k is the Jacobian of the output function at $\hat{x}_{k|k-1}$ and the output prediction error used in the correction step is $e_k = y_k^m - C_k \hat{x}_{k|k-1} = y_k^{nl,m} - h_k(\hat{x}_{k|k-1})$.

As shown in Jeffrey, Preston, and Jeremy (2012), the state estimates $\hat{x}_{k|k}, \hat{x}_{k+1|k}$ generated by the Kalman filter (4) are part of the optimizer of the following optimization problem

$$\begin{aligned} \hat{x}_{0|k}, \ldots, \hat{x}_{k|k}, \hat{x}_{k+1|k} = \underset{x_0, \ldots, x_k, x_{k+1}}{\arg\min} & \left\| x_0 - \hat{x}_{0|-1} \right\|_{P_{0|-1}^{-1}}^2 \\ + \sum_{i=0}^{k} & \left\| y_i^m - C_i x_i \right\|_{R_i^{-1}}^2 + \left\| x_{i+1} - x_i \right\|_{Q_i^{-1}}^2 . \end{aligned} \tag{5}$$

Problem (5) can be solved recursively at each step k by minimizing the following cost functions:

$$\hat{x}_{k|k} = \underset{x_k}{\arg\min} \left\| x_k - \hat{x}_{k|k-1} \right\|_{P_{k|k-1}^{-1}}^2 + \left\| y_k^m - C_k x_k \right\|_{R_k^{-1}}^2 \tag{6a}$$

$$\begin{aligned} \hat{x}_{k|k}, \hat{x}_{k+1|k} = \underset{x_k, x_{k+1}}{\arg\min} & \left\| x_k - \hat{x}_{k|k} \right\|_{P_{k|k}^{-1}}^2 \\ & + \left\| x_{k+1} - x_k \right\|_{Q_k^{-1}}^2 \end{aligned} \tag{6b}$$

where $\hat{x}_{k|k}, \hat{x}_{k+1|k}, P_{k|k}$, and $P_{k+1|k}$ are the state estimates and covariance matrices computed as in (4).

3. EKF under non-smooth regularization

In order to regularize the model, we modify the classical iterations (4) by changing the minimization in (6a) to

$$\begin{aligned} \hat{x}_{k|k} = \underset{x_k}{\arg\min} & \frac{1}{2} \left\| x_k - \hat{x}_{k|k-1} \right\|_{P_{k|k-1}^{-1}}^2 \\ & + \frac{1}{2} \left\| y_k^m - C_k x_k \right\|_{R_k^{-1}}^2 + g(x_k) \end{aligned} \tag{7}$$

where $g(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R} \cup \{+\infty\}$ is a possibly non-smooth and non-convex regularization term. By defining $\mathcal{S} = \{(x_k, v) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} : x_k = v\}$, (7) can be equivalently reformulated as the following constrained optimization problem

$$\begin{aligned} \hat{x}_{k|k}, v^\star = \underset{(x_k, v) \in \mathcal{S}}{\arg\min} & \frac{1}{2} \left\| x_k - \hat{x}_{k|k-1} \right\|_{P_{k|k-1}^{-1}}^2 \\ & + \frac{1}{2} \left\| y_k^m - C_k x_k \right\|_{R_k^{-1}}^2 + g(v) \end{aligned} \tag{8}$$

which can be solved by executing the following scaled ADMM iterations (Boyd et al., 2011):

$$\begin{aligned} \hat{x}_{k|k}^{t+1} = \underset{x_k}{\arg\min} & \left\| x_k - \hat{x}_{k|k-1} \right\|_{P_{k|k-1}^{-1}}^2 \\ & + \left\| y_k^m - C_k x_k \right\|_{R_k^{-1}}^2 + \rho \left\| x_k - v^t + w^t \right\|_2^2 \end{aligned} \tag{9a}$$

$$\nu^{t+1} = \arg\min_{\nu} g(\nu) + \frac{\rho}{2}\left\|\nu - \hat{x}_{k|k}^{t+1} - w^{l}\right\|_2^2$$
$$= \operatorname{prox}_{\frac{g}{\rho}}(\hat{x}_{k|k}^{t+1} + w^{t}) \quad (9b)$$
$$w^{t+1} = w^{t} + \hat{x}_{k|k}^{t+1} - \nu^{t+1} \quad (9c)$$

for $t = 0, \ldots, n_a - 1$, where $\rho > 0$ is a hyper-parameter to be calibrated and "prox" is the proximal operator (Parikh & Boyd, 2013). As shown in Boyd et al. (2011), in the convex case, the ADMM iterations (9a)–(9c) converge to the optimizer of (8) as $n_a \to \infty$, and often converge to a solution of acceptable accuracy within a few tens of iterations. Iteration (9c) is straightforward to compute; iteration (9b) can be solved explicitly and efficiently with complexity $\mathcal{O}(n_x)$ for a wide range of non-smooth and non-convex regularization functions g, such as $g(x) = \|x\|_0$, $g(x) = \|x\|_1$, and the indicator function $g(x) = 0$ if $x_{\min} \le x \le x_{\max}$ or $+\infty$ otherwise (Parikh & Boyd, 2013). Iteration (9a) can be rewritten as

$$\hat{x}_{k|k}^{t+1} = \arg\min_{x_k} \left\|x_k - \hat{x}_{k|k-1}\right\|_{P_{k|k-1}^{-1}}^2 + \left\|\overline{y}_k^m - \overline{C}_k x_k\right\|_{\overline{R}_k^{-1}}^2 \quad (10)$$

where $\overline{y}_k^m = [(y_k^m)' \ (\nu^t - w^t)']'$, $\overline{C}_k = [C_k' \ I]'$, and $\overline{R}_k = \begin{bmatrix} R_k & 0 \\ 0 & \rho^{-1} I \end{bmatrix}$. Therefore, iteration (9a) can be performed directly in the correction step of the EKF by including n_x additional "fake" state measurements $\nu^t - w^t$ with covariance matrix $\rho^{-1} I$.

Algorithm 1 summarizes the proposed extension of EKF with ADMM iterations (EKF-ADMM). The algorithm returns the estimate $\hat{x}_{k|k}$ of the parameter vector x obtained after processing N measurements. It also returns the last value of ν, which could be used as an alternative estimate of x too; for example, in case g is the indicator function of a constraint set, ν would be guaranteed to be feasible. Note that the dual vector w is not reset at each EKF iteration k; it is used as a warm start for the next n_a ADMM iterations at step $k+1$, as the solutions $\hat{x}_{k|k}$ at consecutive time instants k are usually similar.

3.1. Computational complexity

Given the block-diagonal structure of the measurement noise covariance matrix $\overline{R}_k$, (10) can be rewritten as

$$\hat{x}_{k|k}^{t+1} = \arg\min_{x_k} \left\|x_k - \hat{x}_{k|k-1}\right\|_{P_{k|k-1}^{-1}}^2 + \left\|y_k^m - C_k x_k\right\|_{R_k^{-1}}^2 + \rho \sum_{i=1}^{n_x} \left\|x_{k,i} - \nu_i^t + w_i^t\right\|_2^2 \quad (11)$$

which highlights the separation of the contributions of the true measurements y_k^m and of the fake regularization measurements $\nu^t - w^t$ in the correction step; moreover, we can process the measurements $\nu_i^t - w_i^t$ separately one by one. This allows designing a computationally more efficient and numerically robust version of the proposed EKF-ADMM algorithm, as the correction due to the true measurements y_k^m can be performed only once, instead of n_a times, as it does not change with t. Moreover, there is no need for any matrix inversion when processing the fake measurements, as by processing them one by one, the matrix inversion required to compute the Kalman gain becomes a simple division, since each measurement is just a scalar value. Assuming a complexity $\mathcal{O}(n_x)$ for evaluating the proximal operator, EKF-ADMM has complexity $\mathcal{O}(n_x^3 + n_a n_x^2)$, which is the same order of the full EKF for general state estimation. Moreover, EKF-ADMM has the same number of Jacobian matrices evaluations than the classical EKF, which is usually the most time-consuming part in case x represents the weights and bias terms of a neural network model to learn. Summarizing, the proposed approach is computationally efficient and, if the Kalman filter is implemented using numerically robust factored or square-root modifications (Thornton & Bierman, 1975), the method is appealing for embedded applications.

Algorithm 1 EKF-ADMM

Input: $\hat{x}_{0|-1}$, $P_{0|-1}^{-1}$, $\nu = \hat{x}_{0|-1}$, $w = 0$, $\rho > 0$
  for $k = 0, \ldots, N-1$ **do**
    $K_k = P_{k|k-1}\overline{C}_k^T(\overline{R}_k + \overline{C}_k P_{k|k-1}\overline{C}_k^T)^{-1}$
    for $t = 0, \ldots, n_a - 1$ **do**
      $\hat{x}_{k|k} \leftarrow \hat{x}_{k|k-1} + K_k\left(\begin{pmatrix} y_k^m \\ \nu - w \end{pmatrix} - \overline{C}_k \hat{x}_{k|k-1}\right)$
      $\nu \leftarrow \operatorname{prox}_{\frac{g}{\rho}}(\hat{x}_{k|k} + w)$
      $w \leftarrow w + \hat{x}_{k|k} - \nu$
    end for
    $P_{k|k} = (I - K_k\overline{C}_k)P_{k|k-1}$
    $\hat{x}_{k+1|k} = \hat{x}_{k|k}$
    $P_{k+1|k} = P_{k|k} + Q_k$
  end for
  return $\hat{x}_{N-1|N-1}$, ν

4. Regret analysis

We investigate the theoretical properties of EKF-ADMM for linear time-varying models, i.e., models of the form

$$y_k = h_k(x_k) = C_k x_k \quad (12)$$

where C_k are now given time-varying matrices for $k = 0, 1, \ldots, N-1$, and convex regularization terms g. In particular, we want to evaluate the ability of the algorithm to solve the optimization problem $\min_x \sum_{k=0}^{N-1}(f_k(x) + g(x))$ online, where $f_k(x) = \frac{1}{2}\left\|y_k^m - C_k x\right\|_{R_k^{-1}}^2$, via the following two regret functions $R_f(N) = \sum_{k=0}^{N-1}(f_k(x_k) + g(\nu_k)) - \min_{x,\nu\in\mathcal{S}} \sum_{k=0}^{N-1}(f_k(x) + g(\nu))$ and $R_c(N) = \sum_{k=0}^{N-1}\|x_{k+1} - \nu_k\|^2$, where, to simplify the notation, we have defined $x_k = \hat{x}_{k|k-1}$, $P_k = P_{k|k-1}$, $\forall k = 0, 1, \ldots, N-1$. Notice that $R_f(N)$ quantifies the loss we suffer by learning the model online instead of solving it in a batch way given all N measurements, while $R_c(N)$ quantifies the violation of the constraint $x = \nu$. To ensure a proper behavior of EKF-ADMM, we want to prove a sublinear regret bound for both, i.e., $R_f(N) \le \mathcal{O}(\sqrt{N})$ and $R_c(N) \le \mathcal{O}(\sqrt{N})$ (Wang & Banjaree, 2012).

EKF-ADMM is a generalization of the online ADMM method proposed in Wang and Banjaree (2012), in which a sublinear regret bound is derived for the case $n_a = 1$ and $P_k^{-1} = P^{-1} \succ 0$, $\forall k$, while, more recently, in Zhang, Xiao, Wu, and Zhang (2024) a sublinear regret bound has been derived for the case $n_a = 1$ and $P_k^{-1} \ge P_{k+1}^{-1}$, $\forall k$. Here we will provide a sublinear regret in the case $n_a = 1$ and $P_k^{-1} = P_{k+1}^{-1}, \forall k \ge k_n \ll N$, which is a reasonable assumption as the EKF covariance matrix, when estimating the parameters of a model, usually has a transient and then reaches a steady-state value. By assuming $n_a = 1$ and $P_k^{-1} = P_{k+1}^{-1}, \forall k \ge k_n \ll N$, Algorithm 1 can be equivalently rewritten as in Algorithm 2.

The following Theorem 4.1 is an extension of Wang and Banjaree (2012, Theorem 4), and provides conditions for sublinear regret bounds of Algorithm 2 in the case of a linear time-varying model (12) and convex regularization function g.

Theorem 4.1. *Let $\{x_k, \nu_k, w_k\}_{k=0}^{N-1}$ be the sequence generated by Algorithm 2 and let $x^\star$, $\nu^\star$ be the best solution in hindsight, i.e. $x^\star$, $\nu^\star = \arg\min_{x,\nu\in\mathcal{S}} \sum_{k=0}^{N-1}(f_k(x)+g(\nu))$. Let the following assumptions hold:*

A1. $\exists \alpha, G_f, D_x, D_\nu, F > 0$ such that $\forall k = 0, \ldots, N-1$:

(a) $\|x - y\|_{P_k^{-1}}^2 \ge \alpha \|x - y\|_2^2$, $\forall x, y$

(b) $\|\nabla f_k(x_k)\|_2^2 = \left\|C_k^T R_k^{-1}(C_k x_k - y_k^m)\right\|_2^2 \le G_f^2$

Algorithm 2 EKF-ADMM ($n_a = 1$, frozen P)

Input: $x_0, P_0^{-1}, \nu_1 = x_0, w_0 = 0, \rho, \eta > 0, k_n \geq 0$
for $k = 0, \dots, N-1$ **do**

$$x_{k+1} \leftarrow \operatorname*{argmin}_x \frac{1}{2} \left\| y_k^m - C_k x \right\|^2_{R_k^{-1}} + w_k^T (x - \nu_k) + \\ + \frac{\rho}{2} \left\| x - \nu_k \right\|^2 + \frac{\eta}{2} \left\| x - x_k \right\|^2_{P_k^{-1}}$$

$$\nu_{k+1} \leftarrow \operatorname*{argmin}_\nu g(\nu) + w_k^T (x_{k+1} - \nu) + \\ + \frac{\rho}{2} \left\| x_{k+1} - \nu \right\|^2 = \operatorname{prox}_{\frac{g}{\rho}} (x_{k+1} + \frac{w_k}{\rho})$$

$$w_{k+1} \leftarrow w_k + \rho(x_{k+1} - \nu_{k+1})$$

if $k < k_n$ **then**

$$P_{k+1}^{-1} \leftarrow (Q_k + (P_k^{-1} + \bar{C}_k^T \bar{R}_k^{-1} \bar{C}_k)^{-1})^{-1}$$

else

$$P_{k+1}^{-1} \leftarrow P_k^{-1}$$

end
end for
return x_N, ν_N

(c) $\frac{1}{2} \left\| x^\star \right\|^2_{P_k^{-1}} \leq D_x^2$ *and* $\left\| \nu^\star \right\|_2^2 \leq D_\nu$

(d) $f_k(x_{k+1}) + g(\nu_{k+1}) - (f_k(x^\star) + g(\nu^\star)) \geq -F$

A2. $\exists M_{k_n} \geq 0$ *such that* $\frac{1}{2} \sum_{k=1}^{k_n} \left\| x^\star - x_k \right\|^2_{(P_k^{-1} - P_{k-1}^{-1})} \leq M_{k_n}$

A3. To ease the notation, $x_0 = 0, g(0) = 0$ *and* $g(\nu) \geq 0$.

Then, if $\eta = \frac{G_f \sqrt{N}}{D_x \sqrt{2\alpha}}$ *and* $\rho = \sqrt{N}$, *the following sublinear regret bounds are guaranteed:*

$$R_f(N) \leq \frac{\sqrt{N} D_\nu}{2} + \frac{G_f D_x \sqrt{N}}{\sqrt{2\alpha}} + \frac{G_f \sqrt{N} (D_x^2 + M_{k_n})}{D_x \sqrt{2\alpha}} \tag{13a}$$

$$R_c(N) \leq 2F\sqrt{N} + D_\nu + \frac{2G_f}{D_x \sqrt{2\alpha}} (D_x^2 + M_{k_n}). \tag{13b}$$

Proof. See Appendix. □

Corollary 4.2. *Consider the linear time-invariant case* $C_k \equiv C_0$, $\forall k \geq 0$. *If the steady-state Kalman filter is used, then Theorem 4.1 holds with* $M_{k_n} = 0$.

In general, as proved in Zhang et al. (2024), Theorem 4.1 holds with $M_{k_n} = 0$ whenever $P_k^{-1} \succeq P_{k+1}^{-1}, \forall k$. Intuitively, this means that for online model adaptation we need to limit the importance of the previous samples to promptly adapt the model to changes and therefore bound the regret function. This can be accomplished, for example, using the EKF with a proper forgetting factor (Abulikemu & Changliu, 2020).

5. Simulation results

We evaluate the performance of the proposed EKF-ADMM algorithm on three different examples: online LASSO (Ranstam & Cook, 2018), online training of a neural network on data from a static model under ℓ_1 regularization or bound constraints, online adaptation of a neural network on data from a time-varying model under ℓ_0 regularization.

5.1. Online LASSO

Consider the LASSO problem $\min_x \sum_{k=0}^{N-1} (\frac{1}{2} \left\| y_k^m - C_k x \right\|^2_{R^{-1}} + \lambda \left\| x \right\|_1)$, where $x \in \mathbb{R}^3$ is the parameter vector, $C_k \in \mathbb{R}^{2\times 3}$ are randomly generated matrices with $C_{k,ij} \sim \mathcal{N}(0, 1)$, $y_k^m = C_k x_{\text{true}} + r_k \in \mathbb{R}^2$ is the vector of measurements and r_k is random measurement noise, $r_{k,i} \sim \mathcal{N}(0, 10^{-2})$. We will evaluate the behavior of the regret functions $R_f(N)$ and $R_c(N)$ as $N \to \infty$ when using Algorithm 2. The following settings are used: $P_{0|-1} = I, Q_k = 10^{-6} I, R = 10^{-3} I, k_n = 10^3, \rho = 10^4 \sqrt{N}$ and $\eta = 10^{-6} \sqrt{N}$, where $P_{0|-1}, Q_k$ and R are manually tuned to maximize the Kalman filter performance without regularization and ρ, η are chosen according to the expressions provided in Theorem 4.1. Results for different values of λ are shown in Fig. 1. In this case, Theorem 4.1 holds and, as expected, both the regrets $R_f(N)$ and $R_c(N)$ decrease as the number N of samples increases.

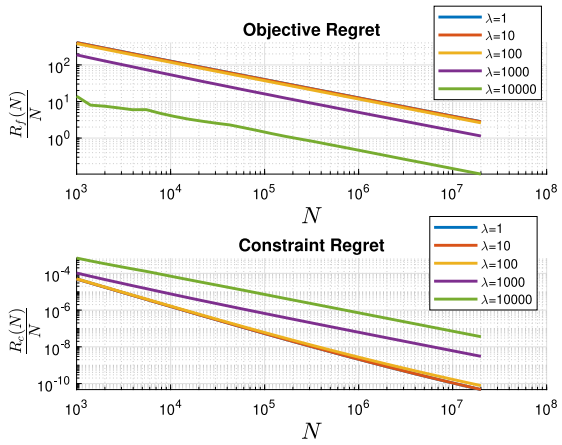

Fig. 1. Objective and constraint regret for online LASSO.

5.2. Online learning of a static model

Consider the dataset generated by the static nonlinear model $y_k^m = \frac{z_{k,1}^2 - e^{\frac{z_{k,2}}{10}}}{3 + |z_{k,1} + z_{k,2}|} + r_k$. We want to train online a neural network $h_k(x)$ with 2 layers, 8 neurons in each layer, and tanh activation function, with $n_x = 105$ trainable weights in total. The training is performed on $N = 10^5$ randomly generated data points, where $z_{k,i} \sim \mathcal{U}_{[-10,10]}$ and $r_k \sim \mathcal{N}(0, 10^{-2})$. Let $\{x_k\}_{k=0}^{N-1}$ be the sequence of weights generated by Algorithm 1: we evaluate the online adaptation performance by means of the regret function $R_f(N) = \sum_{k=0}^{N-1} (f_k(x_k) + g(x_k)) - \min_x \sum_{k=0}^{N-1} (f_k(x) + g(x))$, where $f_k(x) = \frac{1}{2} \left\| y_k^m - h_k(x) \right\|^2$, and the quality of a given solution x using the performance indices $\text{Loss}(x) = \frac{1}{N} \sum_{k=0}^{N-1} (f_k(x) + g(x))$, $\text{Mse}(x) = \frac{1}{N} \sum_{k=0}^{N-1} f_k$, $\text{Reg}(x) = \frac{1}{N} \sum_{k=0}^{N-1} g(x)$ and $\text{Cv}(x) = \left\| x - \Pi_{\mathcal{C}}(x) \right\|_2^2$, where, given a constraint set $\mathcal{C} \subseteq \mathbb{R}^{n_x}$, $\Pi_{\mathcal{C}}(x)$ is the projection of the point x onto $\mathcal{C}$. The training is performed in MATLAB R2022a on an Intel Core i7 12700H CPU with 16 GB of RAM, using the library CasADi (Andersson, Gillis, Horn, Rawlings, & Diehl, 2019) to compute the required Jacobian matrices via automatic differentiation. All results are averaged over 20 runs starting from different initial conditions, that were randomly generated using the well known Xavier's initialization strategy.

5.2.1. ℓ_1 regularization

We train the neural model under the regularization function $g(x) = \lambda \left\| x \right\|_1$, with $\lambda = 10^{-4}$. We selected the following hyperparameters: $\rho = 10\lambda, n_a = 1, Q_k = 10^{-4} I, R_k = I$ and $P_{0|0} = 100 I$. We compare the results to different online optimization alternatives: online ADMM (Wang & Banjaree, 2012) with constant matrix $P = 10^{-2} I$ (online-ADMM), EKF-ADMM with time-varying $\rho_k = 10^{\frac{k}{N} - 2} \lambda$ (EKF-ADMMtv), EKF with ℓ_1-regularization (Bemporad, 2021) (EKF-ℓ_1) and SMIDAS (Shalev-Shwartz & Tewari,

Table 1
Online learning a static model of (1) with ℓ_1 regularization: mean (standard deviation) Loss, Mse, sparsity ratio and execution time obtained over 20 runs.

	Loss (10^{-3})	Mse (10^{-3})	Sparsity (%)	Time [s]
LBFGS (Bemporad, 2025)	5.40 (0.72)	1.03 (0.19)	80.66 (5.32)	80.51 (2.42)
NAILM (Bemporad, 2023)	5.24 (0.48)	1.06 (0.15)	63.85 (5.00)	235.41 (52.78)
EKF-ADMM	5.99 (0.68)	1.44 (0.17)	45.28 (4.98)	58.27 (1.41)
EKF-ADMMtv	5.27 (0.46)	1.29 (0.42)	57.00 (7.95)	55.90 (1.41)
online-ADMM (Wang & Banjaree, 2012)	10.38 (1.7)	4.68 (1.8)	3.62 (2.21)	530.69 (29.09)
EKF-ℓ_1 (Bemporad, 2021)	5.47 (0.67)	1.42 (0.26)	56.42 (7.67)	12.46 (0.27)
SMIDAS (Shalev-Shwartz & Tewari, 2009)	89.93 (36.3)	84.67 (36.6)	61.71 (9.98)	4.11 (1.11)

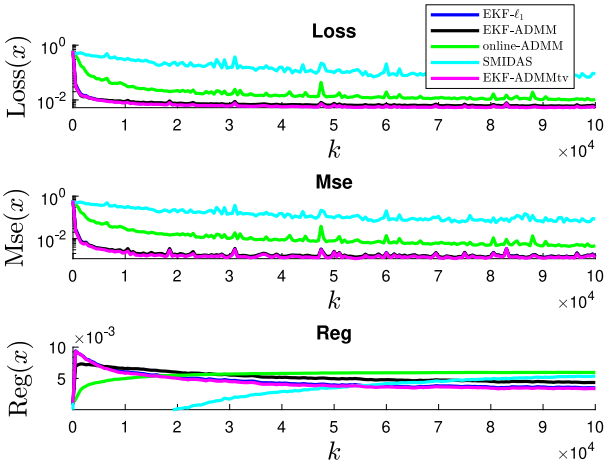

Fig. 2. Online learning with ℓ_1 regularization: Loss, Mse and Reg averaged over 20 runs.

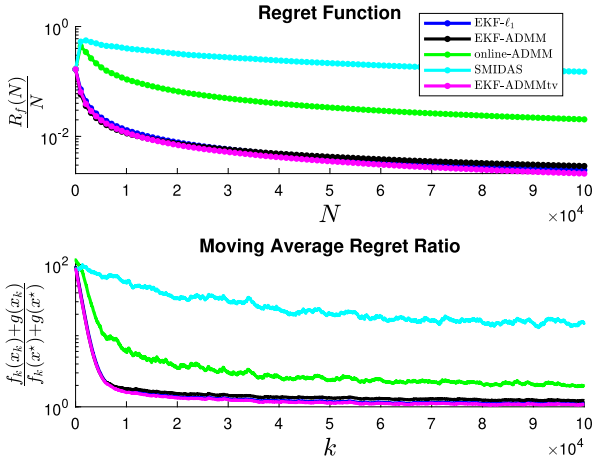

Fig. 3. Online learning with ℓ_1 regularization: regret and sample regret averaged over 20 runs.

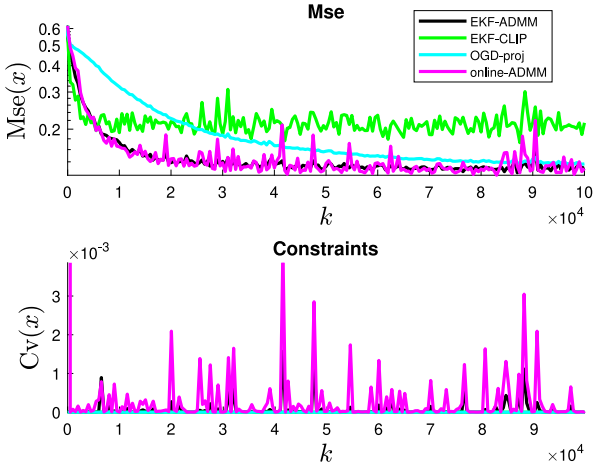

Fig. 4. Online learning with bounds: Mse and constraints violation averaged over 20 runs.

2009) with learning rate $\eta = 5 \cdot 10^{-2}$. Notice that $P_{0|0}, R_k$ and Q_k are chosen by manually tuning EKF-ℓ_1 and then used for all other approaches, while all the remaining method-specific hyperparameters have been manually tuned to maximize their performance. The reason for choosing a time-varying ρ_k is that fake measurements are usually not accurate initially, so that it is better to start with a higher value of ρ_k^{-1} and then decrease it progressively. In addition, we compare with two offline batch algorithms: NAILM (Bemporad, 2023) and LBFGS (Bemporad, 2025), the latter using the Python library `jax-sysid`. Note that all the online algorithms consume the dataset only once (1 epoch), except NAILM and LBFGS that run over 150 and 5000 epochs respectively. The results obtained at the end of the training phase are reported in Table 1.

Among the online approaches, EKF-ADMMtv provides the lowest loss: this is also true during the training phase, as shown in Fig. 2. The online learning performance of EKF-ADMM can be also evaluated by looking at the regret function in Fig. 3, where it is also apparent that, compared to the other approaches, the proposed algorithm improves the solution quality faster.

5.2.2. Bound constraints

Let us now repeat the training under the bound constraints imposed by the regularization function $g(x) = 0$ if $x \in \mathcal{C}$ and $g(x) = +\infty$ otherwise, where $\mathcal{C} = \{x \in \mathbb{R}^{n_x} : |x_i| \leq 0.5\}$. We use the hyper-parameters $\rho = 1, n_a = 5, Q_k = 10^{-4}I, P_{0|0} = 100I$, and $R_k = I$. In this example, we also compare with a simple clipping step of the Kalman filter (EKF-CLIP) and an online projected gradient method (OGD-proj) (Hoi et al., 2021) with learning rate $\eta = 10^{-3}$. Matrices $P_{0|0}, R_k$ and Q_k are the same as for the ℓ_1 example and are in common for all the Kalman-like approaches, while all the other hyperparameters have been manually re-tuned to optimize performance in the bounds constraints example. Results obtained at the end of the training phase are reported in Table 2. Among the online approaches, considering the final Mse, Cv and execution time, EKF-ADMM provides the best quality solution, while OGD-proj attains similar performance but at a slower pace. Fig. 4 shows the performance of the solution during the training phase.

5.3. Online learning of a time-varying model

We test now the ability of EKF-ADMM to adapt the same neural network model, under ℓ_0 regularization, when the data-generating system switches as follows: $y_k^m = \frac{z_{k,1}^2 - e^{\frac{z_{k,2}}{10}}}{3+|z_{k,1}+z_{k,2}|} + r_k$ if $k \leq$

Table 2
Online learning a static model (1) with bound constraints: mean (standard deviation) Mse, constraints violation, and execution time obtained over 20 runs.

	Mse	Cv (10^{-6})	Time [s]
LBFGS (Bemporad, 2025)	0.122 (0.011)	0 (0)	75.87 (5.58)
NAILM (Bemporad, 2023)	0.137 (0.013)	0.38 (0.71)	101.82 (3.88)
EKF-ADMM	0.131 (0.011)	10.76 (4.93)	70.46 (3.45)
online-ADMM (Wang & Banjaree, 2012)	0.129 (0.010)	90.77 (59.37)	610.73 (9.82)
EKF-CLIP	0.214 (0.048)	0 (0)	11.89 (0.12)
OGD-proj (Hoi et al., 2021)	0.137 (0.013)	0 (0)	2.56 (0.15)

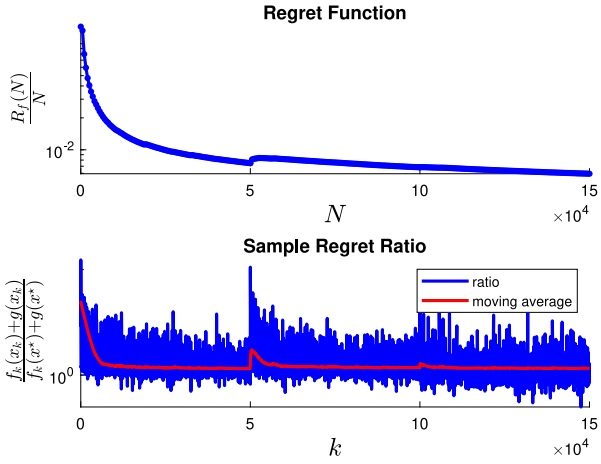

Fig. 5. Online learning with ℓ_0 regularization: regret and sample regret averaged over 20 runs.

$\frac{N}{3}$, $y_k^m = \frac{z_{k,1}^2 - e^{\frac{z_{k,2}}{2}}}{3+|z_{k,1}+z_{k,2}|} + r_k$ if $\frac{N}{3} < k \leq \frac{2N}{3}$ and $y_k^m = \frac{0.3\cdot z_{k,1}^2 - e^{\frac{z_{k,2}}{2}}}{3+|z_{k,1}+z_{k,2}|} + r_k$ if $\frac{2N}{3} < k$, with $N = 1.5 \cdot 10^5$, $z_{k,i} \sim \mathcal{U}_{[-10,10]}$ and $r_k \sim \mathcal{N}(0, 10^{-2})$. We evaluate the regret function $R_f(N) = \sum_{k=0}^{N-1}(f_k(x_k) + g(x_k)) - \min_{z_1,z_2,z_3} \sum_{i=1}^{3} r_i(z_i)$, with $r_i(z_i) = \sum_{k=(i-1)\frac{N}{3}}^{i\frac{N}{3}}(f_k(z_i)+g(z_i))$, where $\{x_k\}_{k=0}^{N-1}$ is the sequence generated by Algorithm 1. The regularization term is $g(x) = \lambda \|w\|_0$, with $\lambda = 10^{-4}$, and we use the EKF-ADMM hyper-parameters $\rho = 10^3 \cdot \lambda$, $p_a = 1$, $Q_k = 10^{-4}I$, and $P_{0|0} = 100I$, which were manually tuned by optimizing the EKF performance. Since the model is now time-varying, we will also use an EKF implementation with forgetting factor $\alpha = 0.9$ (Abulikemu & Changliu, 2020). The resulting regret function is shown in Fig. 5. It is apparent that EKF-ADMM can effectively track changes of the underlying data-generating system.

We have noticed that the choice of ρ is crucial to obtain good adaptation performance. While for some values of ρ the convergence is slow and not satisfactory, with limited effort, in all the reported examples, we could find appropriate values that gave good performance. We remark that, in the general nonlinear case, this might not always be the case, as both the EKF and ADMM are not guaranteed to converge, which might complicate tuning ρ. This could limit the use of ADMM iterations within the EKF framework in certain challenging applications.

6. Conclusions

We have proposed a novel algorithm for online learning of nonlinear parametric models under non-smooth regularization using a combination of EKF and ADMM, for which we derived a sublinear regret bound for the convex linear time-varying case. The approach is computationally cheap and is suitable for factorized or square-root implementations that can make it numerically robust, and is therefore very appealing for embedded applications of adaptive control, such as adaptive model predictive control. The effectiveness of the approach has been evaluated in three numerical examples. Future investigations will focus on extending the approach to the recursive identification of parametric nonlinear state-space dynamical models from input/output data under non-smooth regularization, in which both the hidden states and the parameters are jointly estimated.

Acknowledgments

The research work of Lapo Frascati has been financially supported by ODYS S.r.l. The research work of Alberto Bemporad has been funded by the European Union (ERC Advanced Research Grant COMPACT, No. 101141351). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

Appendix. Proof of Theorem 4.1

Starting from $R_f(N)$, since x_{k+1}, ν_{k+1} are the optimal solutions of the first two optimization problems in Algorithm 2 and since $w_k = w_{k+1} - \rho(x_{k+1} - \nu_{k+1})$ we have that $\nabla f_k(x_{k+1}) = -(w_{k+1} - \rho(\nu_k - \nu_{k+1})) - \eta(P_k^{-1}x_{k+1} - P_k^{-1}x_k)$ and $w_{k+1} \in \partial g(\nu_{k+1})$, where ∂g is the subgradient of g. Due to the convexity of $f_k(\cdot)$ and $g(\cdot)$, $f_k(x_{k+1}) - f_k(x^\star) \leq \nabla f_k(x_{k+1})^T(x_{k+1} - x^\star) = -w_{k+1}^T(x_{k+1} - \nu^\star) + \frac{\rho}{2}(\|\nu^\star - \nu_k\|^2 - \|\nu^\star - \nu_{k+1}\|^2 + \|x_{k+1} - \nu_{k+1}\|^2 - \|x_{k+1} - \nu_k\|^2) + \frac{\eta}{2}(\|x^\star - x_k\|^2_{P_k^{-1}} - \|x^\star - x_{k+1}\|^2_{P_k^{-1}} - \|x_{k+1} - x_k\|^2_{P_k^{-1}})$, where the equality derives from a succession of square completions and by recalling that $x^\star = \nu^\star$, and $g(\nu_{k+1}) - g(\nu^\star) \leq w_{k+1}^T(\nu_{k+1} - \nu^\star)$. Summing the two inequalities together and noticing that $-w_{k+1}^T(x_{k+1} - \nu_{k+1}) + \frac{\rho}{2}\|x_{k+1} - \nu_{k+1}\|^2 = \frac{1}{2\rho}(\|w_k\|^2 - \|w_{k+1}\|^2)$, we obtain:

$$\begin{aligned} &f_k(x_{k+1}) + g(\nu_{k+1}) - (f_k(x^\star) + g(\nu^\star)) \leq \\ &\leq \frac{1}{2\rho}(\|w_k\|^2 - \|w_{k+1}\|^2) - \frac{\rho}{2}\|x_{k+1} - \nu_k\|^2 + \\ &\frac{\rho}{2}(\|\nu^\star - \nu_k\|^2 - \|\nu^\star - \nu_{k+1}\|^2) + \frac{\eta}{2}(\|x^\star - x_k\|^2_{P_k^{-1}} \\ &- \|x^\star - x_{k+1}\|^2_{P_k^{-1}} - \|x_{k+1} - x_k\|^2_{P_k^{-1}}) \end{aligned} \tag{A.1}$$

Considering that $f_k(x_k) - f_k(x_{k+1}) \leq \nabla f_k(x_k)^T(x_k - x_{k+1}) \leq \frac{1}{2\alpha\eta}\|\nabla f_k(x_k)\|^2 + \frac{\alpha\eta}{2}\|x_k - x_{k+1}\|^2$, where the second inequality is due to Fenchel–Young's inequality, and considering Assumption A1.a of the theorem, we have that $f_k(x_k) + g(\nu_{k+1}) - (f_k(x^\star) + g(\nu^\star)) \leq \frac{1}{2\rho}(\|w_k\|^2 - \|w_{k+1}\|^2) - \frac{\rho}{2}\|x_{k+1} - \nu_k\|^2 + \frac{\rho}{2}(\|\nu^\star - \nu_k\|^2 - \|\nu^\star - \nu_{k+1}\|^2) + \frac{1}{2\alpha\eta}\|\nabla f_k(x_k)\|^2 + \frac{\eta}{2}(\|x^\star - x_k\|^2_{P_k^{-1}} - \|x^\star - x_{k+1}\|^2_{P_k^{-1}})$.

Summing from 0 to $N-1$ and considering Assumption A3 we get

$$R_f(N) = \sum_{k=0}^{N-1}(f_k(x_k) + g(v_{k+1}) - (f_k(x^\star) + g(v^\star)))$$
$$+ g(v_0) - g(v_N) \leq \frac{1}{2\rho}(\|w_0\|^2 - \|w_N\|^2) + \frac{\rho}{2}(\left\|v^\star - v_0\right\|^2$$
$$- \left\|v^\star - v_N\right\|^2) + \frac{1}{2\alpha\eta}\sum_{k=0}^{N-1}\|\nabla f_k(x_k)\|^2 + \frac{\eta}{2}\left\|x^\star - x_0\right\|^2_{P_0^{-1}}$$
$$+ \frac{\eta}{2}\sum_{k=1}^{N-1}\left\|x^\star - x_k\right\|^2_{(P_k^{-1}-P_{k-1}^{-1})}$$

and, therefore, $R_f(N) \leq \frac{\rho}{2}\|v^\star\|^2 + \frac{1}{2\alpha\eta}\sum_{k=0}^{N-1}\|\nabla f_k(x_k)\|^2 + \frac{\eta}{2}\|x^\star\|^2_{P_0^{-1}} + \frac{\eta}{2}\sum_{k=1}^{N-1}\|x^\star - x_k\|^2_{(P_k^{-1}-P_{k-1}^{-1})}$. Because of Assumption A2, $\frac{1}{2}\sum_{k=1}^{N-1}\|x^\star - x_k\|^2_{(P_k^{-1}-P_{k-1}^{-1})} = \frac{1}{2}\sum_{k=1}^{k_n}\|x^\star - x_k\|^2_{(P_k^{-1}-P_{k-1}^{-1})} \leq M_{k_n}$, and taking into account Assumptions A1.b and A1.c we get $R_f(N) \leq \frac{\rho D_v}{2} + \frac{NG_f^2}{2\alpha\eta} + \eta(D_x^2 + M_{k_n})$. Setting $\eta \overset{\text{def}}{=} \frac{G_f\sqrt{N}}{D_x\sqrt{2\alpha}}$ and $\rho \overset{\text{def}}{=} \sqrt{N}$, we get the sublinear regret bound $R_f(N) \leq \frac{\sqrt{N}D_v}{2} + \frac{G_f D_x\sqrt{N}}{\sqrt{2\alpha}} + \frac{G_f\sqrt{N}(D_x^2+M_{k_n})}{D_x\sqrt{2\alpha}}$. Considering now $R_c(N)$, we can rearrange (A.1) and consider Assumption A1.d, $\|x_{k+1} - v_k\|^2 \leq \frac{2F}{\rho} + \frac{1}{\rho^2}(\|w_k\|^2 - \|w_{k+1}\|^2) + (\|v^\star - v_k\|^2 - \|v^\star - v_{k+1}\|^2) + + \frac{\eta}{\rho}\left(\|x^\star - x_k\|^2_{P_k^{-1}} - \|x^\star - x_{k+1}\|^2_{P_k^{-1}} - \|x_{k+1} - x_k\|^2_{P_k^{-1}}\right)$. Summing from 0 to $N-1$, we get:

$$R_c(N) = \sum_{k=0}^{N-1}\|x_{k+1} - v_k\|^2 \leq \frac{2FN}{\rho} + \left\|v^\star\right\|^2$$
$$+ \frac{\eta}{\rho}\left(\left\|x^\star\right\|^2_{P_0^{-1}} + \sum_{k=1}^{N-1}\left\|x^\star - x_k\right\|^2_{(P_k^{-1}-P_{k-1}^{-1})}\right).$$

Considering Assumptions A1.c and A2, we have $R_c(N) \leq \frac{2FN}{\rho} + D_v + \frac{2\eta}{\rho}(D_x^2 + M_{k_n})$ and setting $\eta \overset{\text{def}}{=} \frac{G_f\sqrt{N}}{D_x\sqrt{2\alpha}}$ and $\rho \overset{\text{def}}{=} \sqrt{N}$ we finally get $R_c(N) \leq 2F\sqrt{N} + D_v + \frac{2G_f}{D_x\sqrt{2\alpha}}(D_x^2 + M_{k_n})$. □

References

Abulikemu, A., & Changliu, L. (2020). Robust online model adaptation by extended Kalman filter with exponential moving average and dynamic multi-epoch strategy. *vol. 120*, In *Proc. 2nd conference on learning for dynamics and control* (pp. 65–74).

Akpan, Vincent A., & Hassapis, George D. (2011). Nonlinear model identification and adaptive model predictive control using neural networks. *ISA Transactions*, *50*(2), 177–194.

Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., & Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, *11*(1), 1–36.

Bemporad, A. (2021). Recurrent neural network training with convex loss and regularization functions by extended Kalman filtering. *IEEE Transactions on Automatic Control*, *68*(1), 5661–5668.

Bemporad, A. (2023). Training recurrent neural networks by sequential least squares and the alternating direction method of multiplier. *Automatica*, *156*(1), Article 111183.

Bemporad, A. (2025). An L-BFGS-B approach for linear and nonlinear system identification under ℓ_1 and group-Lasso regularization. *IEEE Transactions on Automatic Control*, *70*(7), 4857–4864.

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, *3*(1), 1–122.

Hoi, S. C. H., Sahoo, D., Lu, J., & Zhao, P. (2021). Online learning: A comprehensive survey. *Neurocomputing*, *459*, 249–289.

Jeffrey, H., Preston, R., & Jeremy, W. (2012). A fresh look at the Kalman filter. *SIAM Review*, *54*(4), 801–823.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, *82*(1), 35–45.

McMahan, H. B. (2017). A survey of algorithms and analysis for adaptive online learning. *Journal of Machine Learning Research*, *18*, 1–50.

Parikh, N., & Boyd, S. (2013). Proximal algorithms. *Foundations and Trends in Optimization*, *1*(3), 123–231.

Pillonetto, G., Aravkin, A., Gedon, D., Ljung, L., Ribeiro, A. H., & Schön, T. B. (2025). Deep networks for system identification: A survey. *Automatica*, *171*, Article 111907.

Ranstam, J., & Cook, J. A. (2018). LASSO regression. *British Journal of Surgery*, *105*(10), 1348.

Scardapane, Simone, Comminiello, Danilo, Hussain, Amir, & Uncini, Aurelio (2017). Group sparse regularization for deep neural networks. *Neurocomputing*, *241*, 81–89.

Schoukens, J., & Ljung, L. (2019). Nonlinear system identification: A user-oriented road map. *IEEE Control Systems*, *39*, 28–99.

Shalev-Shwartz, Shai, & Tewari, Ambuj (2009). Stochastic methods for ℓ_1 regularized loss minimization. In *Proceedings of the 26th annual international conference on machine learning* (pp. 929–936).

Thornton, C. L., & Bierman, G. J. (1975). Gram-Schmidt algorithms for covariance propagation. In *IEEE conference on decision and control* (pp. 489–498).

Van de Geer, Sara (2010). ℓ_1-Regularization in high-dimensional statistical models. In *Proceedings of the international congress of mathematicians* (pp. 2351–2369).

Wang, H., & Banjaree, A. (2012). Online alternating direction method. In *Proc. 29th international conference on machine learning* (pp. 1699–1706). Edinburgh, Scotland, UK.

Zhang, Y., Xiao, Z., Wu, J., & Zhang, L. (2024). Online alternating direction method of multipliers for online composite optimization. arXiv preprint arXiv:1904.02862.

Lapo Frascati received his Master's degree in Control Engineering from the University of Florence in 2017. He worked as control engineer for General Motors, Turin (Italy) and Detroit (US). Currently, he is control engineer at ODYS S.r.l., Milan (Italy), and industrial Ph.D. student in Information Technology at the Polytechnic University of Milan. He is co-inventor of 3 international patents and author of commercial software for embedded model predictive control and machine learning. His research interests include model predictive control, machine learning and their applications in the automotive domain.

Alberto Bemporad received his Master's degree cum laude in Electrical Engineering in 1993 and his Ph.D. in Control Engineering in 1997 from the University of Florence, Italy. In 1996/97 he was with the Department of Systems Science & Mathematics, Washington University, St. Louis. In 1997–1999 he held a post-doctoral position at the Automatic Control Laboratory, ETH Zurich, Switzerland, where he collaborated as a Senior Researcher until 2002. In 1999–2009 he was with the Department of Information Engineering of the University of Siena, Italy, becoming an Associate Professor in 2005. In 2010–2011 he was with the Department of Mechanical and Structural Engineering of the University of Trento, Italy. Since 2011 he has been a Full Professor at the IMT School for Advanced Studies Lucca, Italy, where he served as the Director of the institute from 2012 to 2015. In 2011 he co-founded ODYS S.r.l. He has published more than 400 papers in the areas of model predictive control, hybrid systems, optimization, and automotive control, is the co-inventor of 21 patents, and author or coauthor of various software packages for model predictive control design and implementation, including the Model Predictive Control Toolbox (The Mathworks, Inc.) and the Hybrid Toolbox for MATLAB. He was an Associate Editor of the IEEE Transactions on Automatic Control during 2001–2004 and Chair of the Technical Committee on Hybrid Systems of the IEEE Control Systems Society from 2002 to 2010. He received the IFAC High-Impact Paper Award for the 2011–14 triennial, the IEEE CSS Transition to Practice Award in 2019, the 2021 SAE Environmental Excellence in Transportation Award, the 2024 Beale–Orchard-Hays Prize for Excellence in Computational Mathematical Programming, and an ERC Advanced Research Grant in 2024. He is an IEEE Fellow since 2010 and an IFAC Fellow since 2025.