

A causal filter of gradient information for enhanced robustness and resilience in distributed convex optimization

David Angeli¹, Sabato Manfredi², Tianyi Zhong¹

Abstract

Exchange or communication of gradient samples is crucial in many distributed convex optimization set-ups, featured in modern Cyber-Physical Systems applications (i.e. smart grid, coordination of mobile robots/vehicles, predictive maintenance, just to cite a few). It is, however, prone to noise injection of various nature, including malicious attacks. In this paper we propose a causal filter to process a (possibly corrupted) sequence of sub-gradient samples of an unknown convex function, so as to restore, by minimally perturbing the sequence, compatibility with the underlying convexity prior of the original function.

The algorithm is recursive in nature (to reduce its computational complexity) and computes an optimal filtered gradient sequence that, in real-time, minimizes the square of the perturbation applied to its latest sample under the assumption that prior (filtered) samples are accurate. The filter is initially tested on simple convex functions to illustrate its performance. Then, a consensus-based distributed optimization scheme is considered to emphasize the robustness benefits to convergence of the protocol achieved through the filter, in the presence of corrupt data.

¹David Angeli and Tianyi Zhong are with the Department of Electrical and Electronic Engineering, Imperial College London, email: dangeli@imperial.ac.uk, tianyi.zhong20@imperial.ac.uk. Prof D. Angeli is also with the Dept of Information Engineering of University of Florence.

²Prof S. Manfredi is with the Department of Electrical Engineering and Information Technology of University of Naples Federico II, Italy email: sabato.manfredi@unina.it

1. Introduction

The pervasive computation and communication capabilities of Cyber-Physical Systems are enabling new challenges in the design of distributed algorithms among cooperative/competitive smart devices/systems which aim to collectively
5 implement specific tasks. Several challenges arising in this context such as estimation, decision, learning and control applications ([3]-[6]), can be stated as convex optimization problems and often entail exchange or communication of gradient samples of otherwise private cost functions. As a consequence the resilience and reliability of gradient information is becoming ever more crucial.

10 Indeed, in distributed optimization, sub-gradient descent and consensus-based algorithms are exploited. The common assumption is that all agents cooperate to seek the minimum of a convex global function, through merely local communication. Algorithms exchange local estimates of the global solution (i.e. [7]) and/or local gradients (i.e. [8]), or a local estimation of the gradient of
15 the global cost (i.e. [9]) and locally update their estimation of the global minimum through a blending of sub-gradient descent methods and consensus-based algorithms.

In the control framework, the recent development of reinforcement learning, gradient-based policy and data-driven techniques [10] is increasing the interest
20 in model-free optimal control approaches: they mainly deal with a cost function and its gradient estimation evaluated by experimental data on the physical system.

In distributed and/or collaborative machine learning systems [11, 13, 14, 15], a group of workers (or clients) jointly train a global model coordinated by a
25 server. At each iteration the server broadcasts the parameters of the global learning model to the clients, that use them and private data to locally train the gradient of a loss function. Then clients send the trained gradients to the server that applies a Gradient Aggregation Rule (GAR), typically computing their average [16, 17, 18]. The result is used by the server as a single gradient
30 direction to get the next update of global model parameters and initiates next

learning round.

However, in practical scenarios, the measurements and communication network's surrounding the above optimization and learning frameworks may be non-benign or unreliable so that the gradient exchanged information can be
35 affected by noise injection of various nature, including poisoning attacks from adversarial agents intentionally injecting false data into the network.

Recently, several approaches are proposed to analyze or make consensus and distributed optimization resilient to attacks and network imperfections [19]-[22]. In [23] the effect of errors in the gradient estimation is analyzed. The notion of
40 input to state stability (ISS) is devised by considering as input a perturbation from the true gradient.

Other works consider the scenario of gradient information affected by attacks or noise within gradient-like distributed learning and stochastic optimization frameworks [24]-[31]. Usually the noise affecting the gradient information is
45 assumed unbiased with bounded variance. For instance in [24] a distributed algorithm (called S-AB) is proposed that minimizes a sum of smooth and strongly-convex local cost functions. By choosing a sufficiently small constant step size, the algorithm converges to a neighborhood of the global minimum under the assumption that each agent has access to a stochastic first-order oracle (SFO)
50 with gradient noise having zero-mean and bounded variance.

Most of the approaches to detecting malicious agents within distributed optimization and learning frameworks evaluate a measure of distance among gradients, assuming that gradient of attackers tends to be large. For instance in distributed optimization application, the trends of averaged gradient variations
55 is used to detect the neighbors that are likely to be malicious [29]. The assumption is that malicious nodes tend to give themselves away with the intention to drive the solution value away from the optimal point for the non-malicious nodes in the network. Similar gradient distance-based strategies are considered in distributed learning set-ups, where GARs apply a brute-force search for an optimal
60 clients' subset with the minimal maximum gradient pairwise distance [31, 32], or minimal geometric median [33]. Such gradient-distance based approaches

could not effectively deal with stealthy attacks [34], that operate quietly, hiding evidence of an attacker’s actions.

65 As a potential remedy for detection of anomalous data samples, we point out that one common a priori assumption in the context of distributed optimization is *convexity* of individual objective functions. While this assumption is always carefully exploited in convergence analysis, not much is normally done to guarantee that it is indeed fulfilled by the sampled data provided. Our goal
70 is to explicitly take advantage of this assumption in the exchange of gradient information phase. In a nutshell and to fix ideas, let f be a convex function of scalar argument, and consider that a sequence of 3 gradient values be sampled in the points $x_1 = -1$, $x_2 = 1$ and $x_3 = 0$ respectively. Say that the following values are communicated, $g_1 = -1$, $g_2 = 1$ (as it would happen for instance
75 in the case of clean data corresponding to the function $f(x) = x^2/2$). Then, convexity already constrains our next sample g_3 to be in the interval $[-1, 1]$. If this is the case the value is simply recorded and used for processing of future data; if not i.e. $g_3 = 2$, the filter will activate and replace it with $\tilde{g}_3 = 1$, which is the smallest perturbation of g_3 compatible with convexity.

80 In general, we aim to compute a sequence of gradient samples which is consistent with that of a convex function. In particular, we allow data compatible with convexity to be unaffected by our proposed filter, thus preserving the convergence and optimality properties of the underlying distributed optimization protocols, while, on the other hand, we adjust gradient samples inconsistent
85 with the prior by minimally perturbing them with respect to the value received.

An important contribution of our paper is to conceive of robustness with respect to gradient noise, as the ability of convexity-based optimization algorithms to retain convergence regardless of noise injections.

90 In this respect, we present a gradient filtering approach with the following features:

1. the proposed filters only validate information compatible with an underlying and unknown convex function, and minimally perturb the data to accommodate for one in order to enhance robustness of the optimization algorithm with respect to exogenous attacks and unreliable communications;
95
2. the filters offer improved protection for disturbances or attacks acting after a window of gradient data unaffected by noise. Specifically, as the filter processes a growing number of noise-free data samples, the worst-case impact of arbitrary future disturbances is gradually reduced;
100
3. the filtering action proposed in (7), differently than most approaches proposed in the literature to deal with noisy gradients (see for instance [24, 25]), does not require any assumption about the nature of the noise (e.g. bounded, i.i.d. with zero mean and bounded variance) or of the available data. This should not be seen, however, as a remark on performance of the filter, since different types of noise might have different impact on the quality of the final estimated convex function;
105
4. different from resilient gradient-distance based approaches (see for instance [29, 31, 32]), the filter is able to bound the effects of the most difficult attack to be detected, viz. stealthy attacks [34]. Indeed the latter are typical of adversaries able to finely tune their actions to avoid detection while working toward their objective: to cause the maximum damage while keeping the magnitude of the variation of the attacked signal to a minimum. Therefore such attacks cannot be easily rejected by gradient-distance based approaches;
110
5. persistent noise affecting the gradient information communication channels cannot disrupt the convergence properties of distributed optimization algorithms which embed the filters, as they implicitly reconstruct a time-invariant convex function on the basis of subsequent cutting planes derived from the provided gradient samples. Moreover, while the filter could be used to detect poisoning attacks (and potentially disconnect them), it also allows to discourage self-interested agents which may provide corrupted
115
120

gradient information in response to the trajectory followed by the social optimization problem.

125 The rest of paper is organized as follows. In Section 2 the problem framework and main criteria ensuring the compatibility of preassigned subgradient with at least one convex function are presented, while in Section 3 related Examples and counterexamples are given. In Section 4 the design of a causal filter based on criteria in Section 2 is presented. In Section 5 the filter is firstly validated
 130 in the presence of noisy gradient data for simple convex functions on $\mathbb{R}^2$. Then in Section 6 an example of application scenario within optimization framework is outlined, while a related simulation validation is shown in Section 7. Finally conclusions are remarked in Section 8.

135 **Notation:** in the following, for a convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $\nabla f(x)$ denotes its Clarke's subgradient at x , which is a set of n -dimensional column vectors.

2. Problem framework and preliminary results

Let us consider a finite set of points $\mathcal{X} := \{x_1, x_2, \dots, x_N\} \subset \mathbb{R}^n$. To each
 140 $x_k \in \mathcal{X}$ we associate a corresponding $g_k \in \mathbb{R}^n$ meant to represent sub-gradient samples of some unknown convex function $f(x)$ evaluated at x_k , i.e. we expect existence of a convex $f(x)$ such that $g_k \in \nabla f(x_k)$ for all $k = 1, \dots, N$.

Notice that no information is available to the algorithm about the value of $f(x_k)$, as we envision application scenarios where the function f is private and only
 145 queried with respect to its gradient value.

The following notion is central in the subsequent discussion:

Definition 1. *We say that our data sequence $x_1, \dots, x_N$, and associated gradient samples $g_1, \dots, g_N$ are **compatible with convexity**, if there exists a continuous convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with bounded subgradient, such that*

$$g_k \in \nabla f(x_k) \quad \forall k \in \{1, 2, \dots, N\}.$$

Our end goal is to detect on the fly whether the data available is compatible with existence of such convex function and, if not, take appropriate actions to
 150 minimally adjust the data so as to recover compatibility with the convexity assumption.

The problem of deciding “compatibility with convexity” of our data, can be equivalently reformulated as the problem of finding N scalar values $f_k \in \mathbb{R}$, for $k = 1 \dots N$, such that the function:

$$\hat{f}(x) = \max_{k \in \{1, \dots, N\}} f_k + g_k^T(x - x_k)$$

fulfills $\hat{f}(x_k) = f_k$. In fact, $\hat{f}(x)$ is by construction a convex function, and whenever $\hat{f}(x_k) = f_k$, it holds:

$$\hat{f}(x) \geq f_k + g_k^T(x - x_k) = \hat{f}(x_k) + g_k^T(x - x_k),$$

viz. $g_k \in \nabla \hat{f}(x_k)$. Hence, existence of such values f_k implies compatibility with convexity of the available sub-gradient data samples.

155

Conversely, when gradient samples g_k correspond to some convex function $f(x)$, viz. $g_k \in \nabla f(x_k)$, one might define $f_k := f(x_k)$ and, recalling that:

$$f(x_k) \geq f(x_m) + g_m^T(x_k - x_m)$$

the corresponding function $\hat{f}$ fulfills:

$$f(x_k) \leq \hat{f}(x_k) = \max_{m \in \{1, \dots, N\}} [f(x_m) + g_m^T(x_k - x_m)] \leq f(x_k) = f_k.$$

It is useful to recall that, for a convex function $f(x)$, the following holds:

$$f(x) \geq f(x_0) + \nabla f(x_0)^T(x - x_0) \tag{1}$$

for any x and x_0 on its convex domain of definition and provided $\nabla f(x_0)$ is a singleton. In addition, the same inequality holds for any of its elements,

160 whenever $\nabla f(x_0)$ is a set of cardinality bigger than 1.

More closely related to the approach taken in this note, conditions of *cyclical monotonicity* were proven by Rockafellar in [2, 1] which characterize the subgradient mapping of any lower semicontinuous convex function.

These amount to the following inequalities:

$$\sum_{k=1}^{L-1} g_k^T(x_{k+1} - x_k) \leq 0$$

being fulfilled for any sequence $x_k \in \mathbb{R}^n$, $k = 1, \dots, L$, fulfilling $x_1 = x_L$, and any $g_k \in \nabla f(x_k)$. It turns out that the same conditions not only characterize convexity when they are fulfilled for arbitrary cyclical sequences of points in the convex domain of definition of a convex function, but also characterize existence of a convex function when fulfilled on arbitrary sequences belonging to a finite (non-convex) set $\mathcal{X}$, provided we seek existence of a convex function defined on $\text{co}(\mathcal{X})$. In fact, the values f_k which we are seeking should fulfill the following inequalities:

$$f_k \geq f_m + g_m^T(x_k - x_m), \quad \forall k, m \in \{1, \dots, N\}.$$

For any sequence $x_{i_1}, x_{i_2}, \dots, x_{i_L}$ of length L , of elements of $\mathcal{X}$, we construct the following sum:

$$\sum_{k=1}^{L-1} g_{i_k}^T(x_{i_{k+1}} - x_{i_k}) \tag{2}$$

which, for a convex function f of gradient g_k at x_k amounts to a lower-bound to its total variation- between x_{i_1} and x_{i_L} based on its first-order approximations along the sequence x_{i_k} . Adding up for $k = 1$ to $L - 1$ the following inequalities

$$f_{i_{k+1}} \geq f_{i_k} + g_{i_k}^T(x_{i_{k+1}} - x_{i_k})$$

yields:

$$\sum_{k=1}^{L-1} f_{i_{k+1}} \geq \sum_{k=1}^{L-1} f_{i_k} + g_{i_k}^T(x_{i_{k+1}} - x_{i_k}).$$

In particular, when $i_1 = i_L$, the following holds: $\sum_{k=1}^{L-1} f_{i_{k+1}} = \sum_{k=1}^{L-1} f_{i_k}$, so that the following inequality should hold:

$$0 \geq \sum_{k=1}^{L-1} g_{i_k}^T(x_{i_{k+1}} - x_{i_k}). \tag{3}$$

Indeed the above condition fully characterize existence of f with the required
 165 properties. In particular we are ready to state the following Proposition.

Proposition 1. *Let x_k , $k = 1, \dots, N$ be a sequence of points in $\mathbb{R}^n$, and g_k , $k = 1, \dots, N$ a corresponding sequence of associated sub-gradient vectors (also in $\mathbb{R}^n$). The following statements are equivalent:*

1. *There exist values $f_k \in \mathbb{R}$, for $k = 1 \dots N$, such that the function:*

$$\hat{f}(x) = \max_{k \in \{1, \dots, N\}} [f_k + g_k^T(x - x_k)]$$

fulfills $\hat{f}(x_k) = f_k$.

2. *For any finite sequence $i_1, \dots, i_L \in \{1, 2, \dots, N\}$ with $i_1 = i_L$ it holds:*

$$\sum_{k=1}^{L-1} g_{i_k}^T(x_{i_{k+1}} - x_{i_k}) \leq 0.$$

3. *For any finite sequence $i_1, \dots, i_L \in \{1, 2, \dots, N\}$ (without repetitions of intermediate elements, viz. $i_k \neq i_m$ for all $k \neq m < L$) with $i_1 = i_L$ and $L \leq N$ it holds*

$$\sum_{k=1}^{L-1} g_{i_k}^T(x_{i_{k+1}} - x_{i_k}) \leq 0.$$

Proof. 1. $\Rightarrow$ 2.: Assume that 1. holds. Then, the equality $\hat{f}(x_k) = f_k$ implies

$$f_k \geq f_m + g_m^T(x_k - x_m) \quad \forall k, m$$

170 In turn, this implies condition (3) for all finite sequences $i_1, \dots, i_L$ with $i_L = i_1$, as shown above. Hence 2. holds.

2. $\Rightarrow$ 3. : This implication is trivial as 3. requires the same inequality of 2. on a smaller set of sequences.

175

3. $\Rightarrow$ 2. : To see this, consider any sequence $i_1, i_2, \dots, i_L \in \{1, \dots, N\}$ with $i_1 = i_L$. This induces a weighted digraph on the set of nodes $\{1, \dots, N\}$, where each edge (i, j) has multiplicity equal to the cardinality of $\{q \in \{1, \dots, L-1\} : (i_q, i_{q+1}) = (i, j)\}$. Moreover, since $i_1 = i_L$, the in-degree and out-degree of each

node are equal to each other. By Veblen's Theorem for directed graphs, (see Th. 4.1.1 in [35]), the set of (multi)-edges can be decomposed into the disjoint union of simple cycles (without node repetitions). Let $C_1, C_2, \dots, C_q$ denote such cycles. We have:

$$\sum_{k=1}^{L-1} g_{i_k}^T(x_{i_{k+1}} - x_{i_k}) = \sum_{h=1}^q \sum_{(i,j) \in C_h} g_i^T(x_i - x_j) \leq 0,$$

where the last inequality holds because of the inequality in item 3. applied to each of the simple cycles C_h .

2. $\Rightarrow$ 1. To prove this implication it is useful to introduce the following iterative procedure:

$$f_k^{(t+1)} := \max_{m \in \{1, \dots, N\}} [f_m^{(t)} + g_m^T(x_k - x_m)]. \quad (4)$$

For any initialization $f^{(0)} \in \mathbb{R}^N$, the update Equation (4) generates a sequence $f^{(t)} \in \mathbb{R}^N$ which may (or may not) converge to some equilibrium point $\bar{f}$ fulfilling:

$$\bar{f}_k \geq \bar{f}_m + g_m^T(x_k - x_m), \quad \forall k, m \in \{1, 2, \dots, N\}.$$

It turns out that convergence of $f^{(t)}$ is an alternative way to characterize existence of the convex function $\hat{f}$. The following Lemma holds:

Lemma 1. *The following statements are equivalent:*

1. For any finite sequence $i_1, \dots, i_L \in \{1, 2, \dots, N\}$ with $i_1 = i_L$ it holds:

$$\sum_{k=1}^{L-1} g_{i_k}^T(x_{i_{k+1}} - x_{i_k}) \leq 0.$$

2. There exists an equilibrium $\bar{f} \in \mathbb{R}^N$ for the iteration (4)

3. For all initial conditions $f^{(0)} \in \mathbb{R}^N$, the solution $f^{(t)}$ of (4) is bounded

Proof of Lemma 1. 3. $\Rightarrow$ 2. : By construction, $f_k^{(t+1)} = \max_{m \in \{1, 2, \dots, N\}} f_m^{(t)} + g_m^T(x_k - x_m) \geq f_k^{(t)} + g_k^T(x_k - x_k) = f_k^{(t)}$. Hence $f_k^{(t)}$ is monotonically non-decreasing for all $k \in \{1, 2, \dots, N\}$. Hence $\lim_{t \rightarrow +\infty} f_k^{(t)}$ always exists, possibly infinite. Therefore, the implication 3. $\Rightarrow$ 2. is now obvious as boundedness of solutions implies, by their monotonicity, convergence to some equilibrium $\bar{f}$.

2. $\Rightarrow$ 1. : Notice that any equilibrium $\bar{f}$ corresponds to a function $\hat{f}(x)$ (convex) fulfilling $\hat{f}(x_k) = \bar{f}_k$. Hence, condition 1. follows (as already proved in the implication 1. $\Rightarrow$ 2. of Proposition 1).

1 $\Rightarrow$ 3. : The following formula can be shown by induction:

$$f_k^{(T)} = \max_{i_1, i_2, \dots, i_T: i_{T+1}=k} \left[f_{i_1}^{(0)} + \sum_{k=1}^T g_{i_k}^T(x_{i_{k+1}} - x_{i_k}) \right] \quad (5)$$

Notice, that every sequence of arbitrary length $T \geq N$, can be seen as a cyclic sequence concatenated with a non-cyclic sequence of length strictly less than N . In particular, as there exist at most a finite number of such sequences (of length $< N$), there exists a sufficiently large M , such that:

$$\sum_{k=1}^L g_{i_k}^T(x_{i_{k+1}} - x_{i_k}) \leq M$$

for all sequences $i_1, i_2, \dots, i_L$ with $L < N$. In particular, then, by the induction formula (and condition 1) we see that:

$$\begin{aligned} f_k^{(T)} &= \max_{i_1, i_2, \dots, i_T: i_{T+1}=k} \left[f_{i_1}^{(0)} + \sum_{k=1}^T g_{i_k}^T(x_{i_{k+1}} - x_{i_k}) \right] \leq \\ &\max_{i_1, i_2, \dots, i_T: i_{T+1}=k} f_{i_1}^{(0)} + M = \max_m f_m^{(0)} + M < +\infty \end{aligned}$$

thus showing boundedness of solutions and proving 3. $\square$

We complete the proof of implication 2. $\Rightarrow$ 1. of Proposition 1 as follows. Assume 2. holds. By Lemma 1, there exists an equilibrium $\bar{f}$ of iteration 4. Hence we define:

$$\hat{f}(x) := \max_m [\bar{f}_m + g_m^T(x - x_m)].$$

By definition of equilibrium and construction of $\hat{f}$ it holds $\hat{f}(x_k) = \bar{f}_k$ which shows existence of the required values f_k and of the corresponding convex function $\hat{f}(x)$. This completes the proof of Proposition 1.

Remark 1. *It is worth pointing out that iteration (4) is a special case of so called max-plus discrete linear systems. Their dynamics have been studied for the quantitative analysis of timed Discrete Event Systems.*

3. Examples and counterexamples

Condition (3) for cyclic sequences of length 3 as $k m k$ with $k \neq m$ reads as:

$$(g_k - g_m)^T(x_k - x_m) \geq 0 \quad \forall k, m \in \{1, 2, \dots, N\} \quad (6)$$

corresponding to the well known inequality: $(\nabla f(x_1) - \nabla f(x_2))^T(x_1 - x_2) \geq 0$ holding for convex functions. One might wonder if condition (6), which is a special case of (3) would still be sufficient in order to characterize existence of convex functions matching the prescribed subgradient information. It turns out that this is the case only for scalar valued functions, namely when $x_k \in \mathbb{R}$ for all k . To see this, consider the following counter-examples. Let $\mathcal{X} = \{x_1, x_2, x_3\}$ with:

$$x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Assume the following gradient samples are provided:

$$g_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad g_2 = \begin{bmatrix} -1 \\ +1 \end{bmatrix}, \quad g_3 = \begin{bmatrix} +1 \\ +1 \end{bmatrix}.$$

Then, it is easy to verify that $(g_i - g_k)^T(x_i - x_k) = 0$ for all $i \neq k$, so that condition (6) is fulfilled. On the other hand, condition (3) is not fulfilled on arbitrary sequences of length 3. In particular,

$$g_1^T(x_2 - x_1) + g_2^T(x_3 - x_2) + g_3^T(x_1 - x_3) = 1 + 1 + 0 = 2 > 0.$$

This shows that there is no convex function compatible with the assigned values of subgradient. Notice that, provided gradient samples are compatible with convexity in sets of cardinality 2, there exist functions

$$\hat{f}_{12}(x) = \max\{-1 + g_1^T(x - x_1), g_2^T(x - x_2)\}$$

$$\hat{f}_{23}(x) = \max\{g_2^T(x - x_2), 1 + g_3^T(x - x_3)\}$$

$$\hat{f}_{13}(x) = \max\{g_1^T(x - x_1), g_3^T(x - x_3)\}$$

which admit as subgradient vectors, g_1 and g_2 (at x_1 and x_2), g_2 and g_3 (at x_2 and x_3) and g_1 and g_3 (at x_1 and x_3) respectively. However, no convex function exists that admits all 3 subgradient vectors simultaneously.

4. Optimal filtering of subgradient samples

The results of Section 2 highlight criteria that ensure compatibility of preassigned subgradient samples at specific query points x_k with existence of at least one convex function exhibiting such subgradient vectors. In this section we propose the design of a causal filter that accepts as input a sequence of subgradient samples g_k (evaluated at known points x_k) and seeks to project them onto the space of subgradient vectors compatible with existence of a convex function. In particular, we seek to apply a filtering action that minimizes the norm of the difference between the latest subgradient sample g_k and the corresponding output $\tilde{g}_k$ compatible with existence of a convex function of gradient $\tilde{g}_1, \dots, \tilde{g}_{k-1}$, at the points $x_1, \dots, x_{k-1}$, respectively.

Notice that the filter will act as an identity whenever the g_k s are a faithful “closed-loop” evaluation of the subgradient of a given convex function $f(x)$, viz. when $g_k \in \nabla f(x_k)$ for all k .

The most direct way to formulate such a filtering action is as follows:

$$\begin{aligned} \tilde{g}_{k+1} &= \arg \min_{\tilde{g}_{k+1} \in \mathbb{R}^n} \|\tilde{g}_{k+1} - g_{k+1}\|^2 \\ &\text{subject to} \\ &\sum_{k=1}^{L-1} \tilde{g}_{i_k}^T (x_{i_{k+1}} - x_{i_k}) \leq 0 \\ &\forall L \in \{2, \dots, k\}, \\ &\forall i_1, i_2, \dots, i_L \in \{1, \dots, k+1\} : i_1 = i_L \end{aligned} \tag{7}$$

with initialization $\tilde{g}_1 := g_1$.

Remark 2. The formulation directly implements the criteria of Proposition 1, by ensuring that the latest filtered gradient sample g_{k+1} is minimally perturbed so as to fulfill the constraints in Proposition 1. This criterion, which is causal and meant to operate in real-time, is particularly suitable when it is assumed that initial data are more reliable than samples at later times, as it happens when an attack or a disturbance comes in after some finite time interval. If,

instead, one expects a stationary amount of noise on gradient samples, it would probably make more sense to optimize the integral of the error $\sum_{l=1}^{k+1} \|\tilde{g}_l - g_l\|^2$ with respect to all $\tilde{g}_l, l = 1 \dots k+1$. However, to the best of our knowledge, no recursive formulation of this problem is known, the number of constraints
 215 involved grows exponentially in time, and updating past gradient filtered samples does not guarantee asymptotic convergence of the filtered data to the gradient samples of some time-invariant convex function. In this respect, it would be recommended to run the optimization problem with the integrated cost for a finite time window prior to switching to the recursive algorithm.

Notice that the optimization in (7) entails an exponentially growing number of constraints. To avoid this drawback we propose the following modified iteration which extracts the same filtered gradient subject to a number of constraints that grow quadratically in k :

$$\begin{aligned} \tilde{g}_{k+1} &= \arg \min_{\tilde{g}_{k+1} \in \mathbb{R}^n} \|\tilde{g}_{k+1} - g_{k+1}\|^2 \\ \text{subject to} & \quad (8) \\ \tilde{g}_{k+1}^T (x_j - x_{k+1}) + F_{j,m}(k) + \tilde{g}_m^T (x_{k+1} - x_m) &\leq 0 \\ \forall j, m \in \{1, 2, \dots, k\}, \end{aligned}$$

where the values of $F_{j,m}(k)$ are updated according to:

$$\begin{aligned} F_{j,k+1}(k+1) &= \\ \max_{m \in \{1, \dots, k\}} [F_{j,m}(k) + \tilde{g}_m^T (x_{k+1} - x_m)] \\ F_{k+1,m}(k+1) &= \\ \max_{j \in \{1, \dots, k\}} [\tilde{g}_{k+1}^T (x_j - x_{k+1}) + F_{j,m}(k)] \\ F_{j,m}(k+1) &= \\ \max\{F_{j,m}(k), F_{j,k+1}(k+1) + F_{k+1,m}(k+1)\} \end{aligned} \quad (9)$$

220 with initialization, $F_{1,1}(1) := 0$.

Remark 3. The numerical implementation of problem (8) can further be improved by reducing the number of constraints from k^2 to k by precomputing the

quantities:

$$\tilde{F}_j(k) = \max_{m \in \{1, \dots, k\}} [F_{j,m}(k) + \tilde{g}_m^T(x_{k+1} - x_m)]$$

and modifying the minimization (8) as follows:

$$\begin{aligned} \tilde{g}_{k+1} &= \arg \min_{\tilde{g}_{k+1} \in \mathbb{R}^n} \|\tilde{g}_{k+1} - g_{k+1}\|^2 \\ &\text{subject to} \\ \tilde{g}_{k+1}^T(x_j - x_{k+1}) &\leq -\tilde{F}_j(k) \quad \forall j \in \{1, 2, \dots, k\}, \end{aligned} \tag{10}$$

Our first result, concerning the filter (8) is as follows:

Lemma 2. *For any positive integer t and any $j, m \in \{1, 2, \dots, t\}$ the values of $F_{j,m}(t)$, defined according to (9) fulfill:*

$$\begin{aligned} F_{j,m}(t) &= \max_{L, i_1, i_2, \dots, i_L} \sum_{k=1}^{L-1} \tilde{g}_{i_k}^T(x_{i_{k+1}} - x_{i_k}) \\ &\text{subject to} \\ L &\geq 2, \quad i_1 = j, \quad i_L = m. \end{aligned} \tag{11}$$

Moreover, the sequence $\tilde{g}_k$ is such that:

$$\sum_{k=1}^{L-1} \tilde{g}_{i_k}^T(x_{i_{k+1}} - x_{i_k}) \leq 0 \tag{12}$$

for all $L \geq 2$ and for all $i_1, i_2, \dots, i_L$ such that $i_1 = i_L$.

Proof. The proof of the two statements will be carried out simultaneously and by induction with respect to t . In particular, for (12), we will assume that the claim is true for $i_1, \dots, i_L \in \{1, \dots, t\}$ (with t an arbitrary positive integer) and prove that it then holds for $t+1$. First of all, the claim is trivial for $t=1$. Indeed, for any word $1 = i_1, \dots, i_L = 1$, we see that $F_{1,1}(1) = 0$ consistently with the initialization of (9). Similarly, (12) is fulfilled when $1 = i_1 = i_2 = \dots = i_L$. Assume next that the conditions are fulfilled for a given t . We wish to show that they hold for $t+1$. To this end consider, without loss of generality, an arbitrary cyclic sequence such that $t+1 = i_1, \dots, i_L = t+1$ (viz. such that

$t + 1$ is the first and last element). Clearly:

$$\begin{aligned}
& \sum_{k=1}^{L-1} \tilde{g}_{i_k}(x_{i_{k+1}} - x_{i_k}) = \tilde{g}_{t+1}^T(x_{i_2} - x_{t+1}) + \\
& \sum_{k=2}^{L-2} \tilde{g}_{i_k}^T(x_{i_{k+1}} - x_{i_k}) + \tilde{g}_{i_{L-1}}^T(x_{t+1} - x_{i_{L-1}}) \\
& \leq \tilde{g}_{t+1}^T(x_{i_2} - x_{t+1}) + F_{i_2, i_{L-1}}(t) + \tilde{g}_{i_{L-1}}^T(x_{t+1} - x_{i_{L-1}}) \leq 0
\end{aligned}$$

where the last inequality follows by definition of $\tilde{g}_{t+1}$ and the first one from the induction hypothesis on Equation (11). The previous inequality shows that (12) holds for arbitrary sequences in $\{1, 2, \dots, t+1\}$. We proceed next to show that (11) holds for $t+1$.

Because of the latter inequality,

$$\max_{L, i_1, i_2, \dots, i_L} \sum_{k=1}^{L-1} \tilde{g}_{i_k}^T(x_{i_{k+1}} - x_{i_k})$$

subject to $L \geq 2$, $i_1 = j < t+1$, $i_L = t+1$ and $i_k \in \{1, 2, \dots, t+1\}$ is achieved on a sequence without any repetitions (any loop gives a negative contribution), and equals $\max_{m \in \{1, \dots, t\}} F_{j,m}(t) + \tilde{g}_m^T(x_{t+1} - x_m) := F_{j,t+1}(t+1)$. Similarly,

$$\max_{L, i_1, i_2, \dots, i_L} \sum_{k=1}^{L-1} \tilde{g}_{i_k}^T(x_{i_{k+1}} - x_{i_k})$$

subject to $L \geq 2$, $i_1 = t+1$, $i_L = m < t+1$, $i_k \in \{1, 2, \dots, t+1\}$ is achieved on a sequence without repetitions and equals $\max_{j \in \{1, \dots, t+1\}} \tilde{g}_{t+1}^T(x_j - x_{t+1}) + F_{t+1,m}(t) := F_{t+1,m}(t+1)$. Moreover,

$$\max_{L, i_1, i_2, \dots, i_L} \sum_{k=1}^{L-1} \tilde{g}_{i_k}^T(x_{i_{k+1}} - x_{i_k})$$

subject to $L \geq 2$, $i_1 = j < t+1$, $i_L = m < t+1$, $i_k \in \{1, 2, \dots, t+1\}$ is achieved on a sequence without repetitions and, depending on whether $t+1$ is included or not into the optimal sequence, equals

$$\max\{F_{j,m}(t), F_{j,t+1}(t+1) + F_{t+1,m}(t+1)\} := F_{jm}(t+1).$$

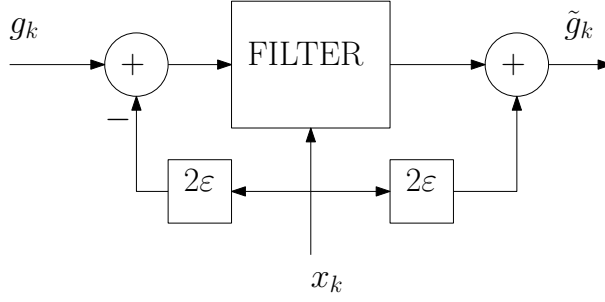


Figure 1: Modified filter for strongly convex functions.

This concludes the proof of (11).

Remark 4. *In some optimization scenarios a priori assumptions on the considered convex functions are stronger; for instance, strong convexity of f is needed. This amounts to convexity of function h , defined as $h(x) := f(x) - \varepsilon x^T x$ for some $\varepsilon > 0$. If the specific value of ε is known one might modify the filter by applying it to the gradient of h . This can provide improved performance and better noise reduction. See Fig. 1 as an illustration of the modification needed to cope with strongly convex function.*

5. Filter performance on noisy gradients

In this section we test the performance of the filter on a simple quadratic function $f(x) : \mathbb{R}^2 \rightarrow \mathbb{R}$,

$$f(x) = 0.6 \|Ax - b\|^2 \quad (13)$$

where $A = \begin{bmatrix} 1.2 & 0.1 \\ 0.1 & 1.2 \end{bmatrix}$ and $b = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$. We select a sequence of 60 points x_k along a spiral trajectory:

$$x_k = 0.8[\cos(0.015k), \sin(0.015k)]^T \log(1799 - 30k).$$

Figure 2 shows the function f and the considered trajectory of points. We then sample the gradient of the function along the considered sequence, and add a

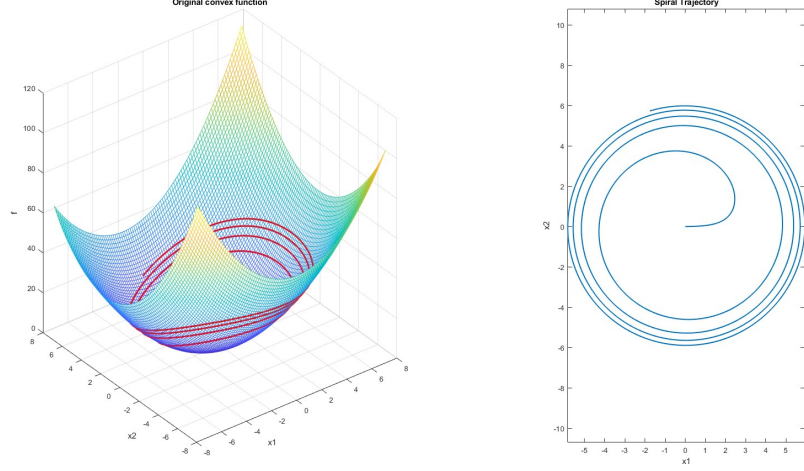


Figure 2: Convex function f and spiral trajectory.

240 white noise of variance 20 to both components of the gradient $\nabla f(x_k)$ for $k > 25$. In particular, the first 25 samples are clean gradient data. The resulting true, noisy and filtered gradient directions are shown in Fig. 3, where blue arrows describe true gradients, red arrows show noisy gradients and yellow arrows show filtered ones. An alternative representation of the filter action can be shown
 245 by plotting component-wise gradient values as a function of k : see Fig. 4, where we interpolated the noisy and filtered data using cubic splines to facilitate interpretation. For $k < 25$ the true, “noisy” and filtered gradients happen to coincide. Notice, in particular, that the filter behaves as an identity for the first 25 samples. Finally, the piecewise linear convex function reconstructed from
 250 filtered data is shown in Fig. 5.

Remark 5. *The optimality criterion proposed for the filter in (7) and its connection with robustness and resilience properties it provides is non-trivial. For instance, different types of weighted norms could be used to assess the distance of $\tilde{g}_{k+1}$ to g_{k+1} . In principle all of them may perform equally well, but it
 255 is clear that specific problems might benefit from specific selection of weights and/or norms. One strong qualitative resilience feature which is always pro-*

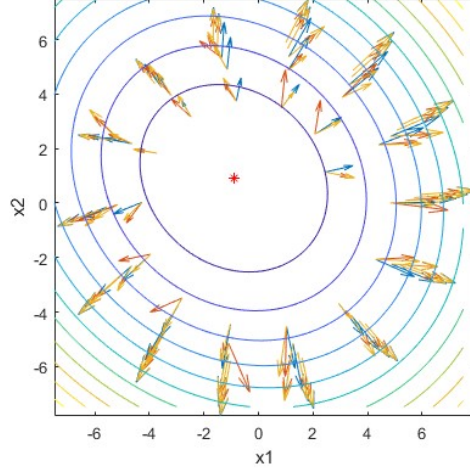


Figure 3: Gradient directions: blue (true), red (noisy) and yellow (filtered).

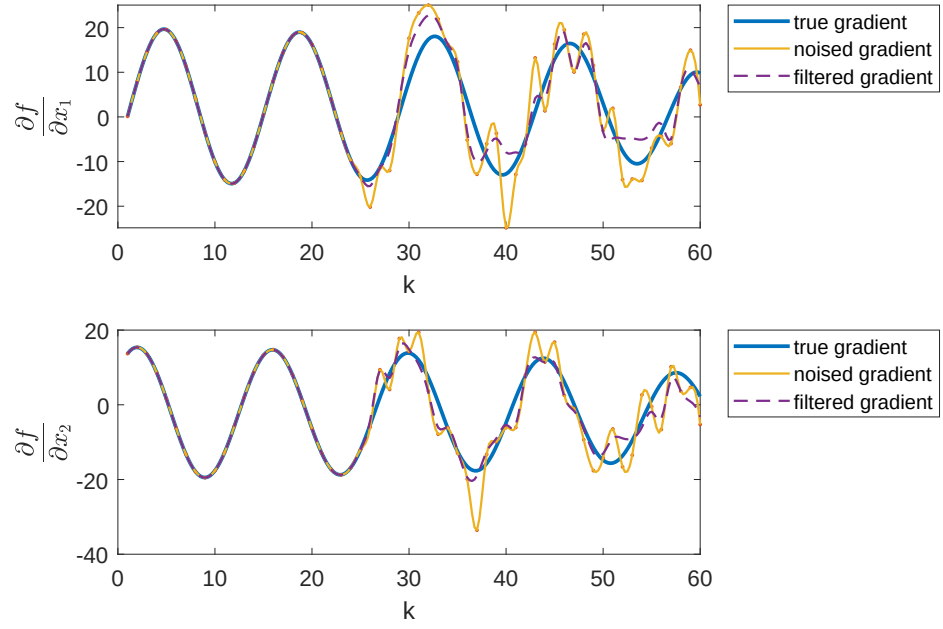


Figure 4: Component-wise gradient information along considered sequence of points x_k

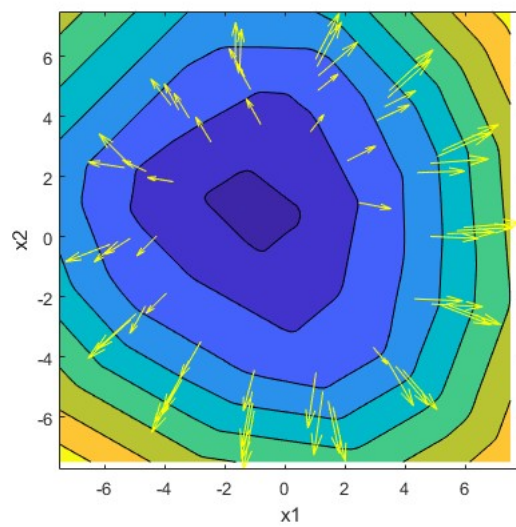
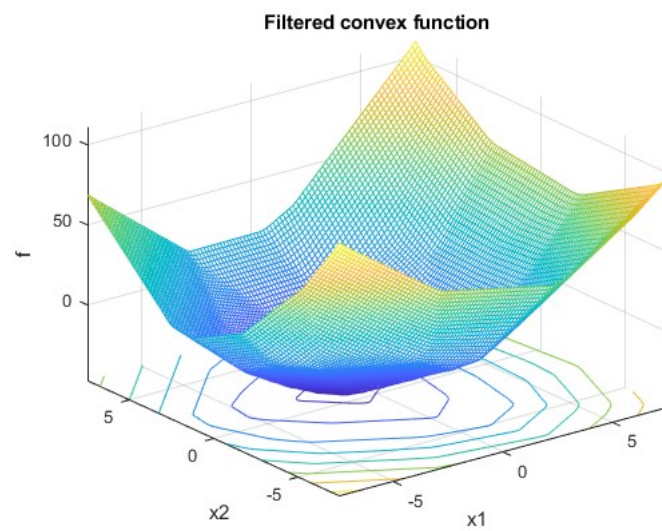


Figure 5: Filtered convex function, graph and sublevel sets.

vided, regardless of such fine tuning, is that it allows distributed (or centralized) gradient-based convex optimization algorithms to converge in the presence of arbitrary noise injections. This follows as the filter provides the optimization algorithm with data compatible with the convexity assumption it is designed for. Of course the quality of the minimum computed is a complicated function of the noise intensity as well as of the time the disturbance injection begins. Bounding such function in relation to the specific feature of the true convex cost and the filter parameters is an interesting direction for future research.

6. A distributed optimization set-up

Consider a finite set of agents, $\mathcal{N} = \{1, 2, \dots, N_a\}$, identified with a slight abuse of notation with the first N_a positive integers. We assume that a finite family of non-empty subsets of $\mathcal{N}$ is given, and denoted by:

$$\mathcal{F} = \{S_1, S_2, \dots, S_Q\}$$

where Q is a positive integer and $S_k \subset \mathcal{N}$ for all $k \in \{1, 2, \dots, Q\}$. For each agent, we define a decision variable x_n , with $n \in \mathcal{N}$ which belongs to a convex set $\mathbb{X}_n$ embedded in Euclidean space. Accordingly, for each $k \in \{1, 2, \dots, Q\}$ we define x_{S_k} the collection of all x_n , variables, for $n \in S_k$ arranged in a column vector. Each S_k represents a *cost center* of our distributed optimization set-up, in the sense that we assume existence of a convex function:

$$f_k : \prod_{n \in S_k} \mathbb{X}_n \rightarrow \mathbb{R}.$$

The value $f_k(x_{S_k})$ quantifies the costs incurred collectively by agents in S_k , with respect of the k -th cost center, when choosing $x_{S_k} \in \prod_{n \in S_k} \mathbb{X}_n$. The function f_k , in turn, is separable (with respect to individual agents):

$$f_k(x_{S_k}) = \sum_{n \in S_k} f_{k,n}(x_n),$$

for some suitable convex functions $f_{k,n} : \prod_{n \in S_k} \mathbb{X}_n \rightarrow \mathbb{R}$. Our aim is to design a distributed algorithm to solve the following optimization problem:

$$\min_{x \in \prod_{n \in \mathcal{N}} \mathbb{X}_n} f(x) := \sum_{k=1}^Q f_k(x_{S_k}). \quad (14)$$

Notice that this amounts to a social optimization problem, as all cost-centers are considered and equally weighted. With respect to standard separable objective functions, of the type:

$$\tilde{f}(x) := \sum_{n \in \mathcal{N}} \tilde{f}_n(x), \quad (15)$$

which emphasize the costs incurred by individual agents, the separation property of $f(x) := \sum_{k=1}^Q f_k(x_{S_k})$ is based on the notion of coupling through cost, in the sense that each cost-center corresponds to a coupled cost function involving agents in S_k . One might recast the considered cost separation according to (15), by remarking:

$$\sum_{k=1}^Q \sum_{n \in S_k} f_{k,n}(x_{S_k}) = \sum_{n \in \mathcal{N}} \underbrace{\sum_{k: n \in S_k} f_{k,n}(x_{S_k})}_{\tilde{f}_n(x)}. \quad (16)$$

Below we distribute the optimization of $f(x)$ according to the cost centers, S_k , $k = 1, \dots, Q$. Each cost-center updates an estimate of the optimal value for x_{S_k} and on the basis of the gradient information provided by agents in S_k , as well as the values shared by neighboring S_k . We say that two cost centers are neighbors if they share at least one agent. Asymptotically we expect cost centers to reach consensus on the shared variables, so as to achieve a global social optimum. In particular, the following distributed gradient method is implemented for all S_j and all $i \in S_j$:

$$x_{S_j}^i(k+1) = (1 - \beta)x_{S_j}^i(k) + \beta \frac{\sum_{q: i \in S_q} x_{S_q}^i(k)}{|\{q: i \in S_q\}|} - \alpha_k \frac{\partial f_j}{\partial x_i}(x_{S_j}(k)), \quad (17)$$

for some $\beta \in (0, 1)$ and for a positive decreasing sequence α_k such that $\sum_k \alpha_k = +\infty$ while $\sum_k \alpha_k^2 < +\infty$.

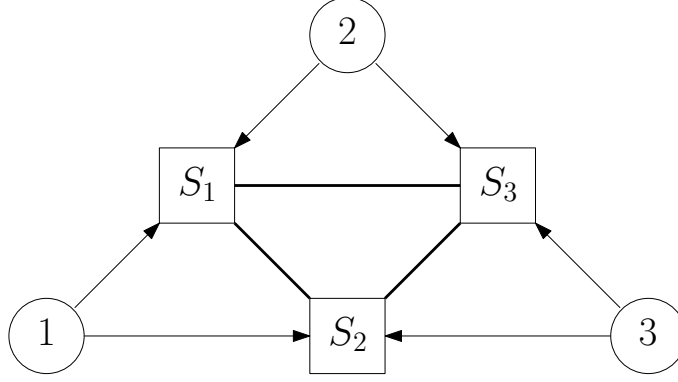


Figure 6: Distributed architecture for social optimization.

Remark 6. *The proposed filter can be also exploited to enhance robustness and resilience of GAR in distributed and collaborative learning architectures (i.e. [11, 12, 15]). Indeed, in distributed and/or collaborative machine learning applications, model training is accomplished through a collaborative procedure in which the participating clients (or learners) receive the current model gradient or parameters from the server, and then separately perform updates via stochastic gradient descent (SGD) using the own locally collected datasets. Finally the clients send their updated gradients (in GAR-based architectures) or model weights to the aggregator server which updates the global model, typically through simple averaging (i.e. as in federated learning). In such scenarios with convex objective functions, the filter could be used by either the clients or the server to enhance the robustness of the convergence and optimality properties with respect to perturbations due to communication noise and/or exogenous poisoning attacks. In both cases, an extensive analysis of advantages and limitations of the proposed filter when applied to a learning scenario is a wider and interesting research aim that will be subject of future work.*

7. Simulation results

As an example we consider the distributed architecture shown in Fig. 6, with agents $\mathcal{N} = \{1, 2, 3\}$ and cost-centers $S_1 = \{1, 2\}$, $S_2 = \{1, 3\}$ and $S_3 = \{2, 3\}$.

The variables x_1, x_2, x_3 associated to individual agents take value in $\mathbb{X}_1, \mathbb{X}_2$ and $\mathbb{X}_3$ (respectively), with $\mathbb{X}_i \subset \mathbb{R}^2$ for all $i \in \mathbb{N}$. We denote by $x = [x'_1, x'_2, x'_3]'$. The cost functions associated to individual agents are of the form:

$$\begin{aligned}\tilde{f}_1(x) &= k_{12}\|Ax_1 - x_2 - b_{12}\|_2^2 + k_{13}\|x_1 - x_3 - b_{13}\|_2^2 \\ \tilde{f}_2(x) &= k_{21}\|x_1 - x_2 - b_{21}\|_2^2 + k_{23}\|x_2 - x_3 - b_{23}\|_2^2 \\ \tilde{f}_3(x) &= k_{31}\|x_1 - x_3 - b_{31}\|_2^2 + k_{32}\|x_2 - x_3 - b_{32}\|_2^2\end{aligned}$$

for some positive coefficients k_{ij} , $i \neq j \in \mathcal{N}$, some vectors $b_{ij} \in \mathbb{R}^2$, $i \neq j \in \mathcal{N}$ and matrix $A \in \mathbb{R}^{2 \times 2}$. Accordingly, the following functions are associated to each cost center:

$$\begin{aligned}f_1(x_{S_1}) &= k_{12}\|Ax_1 - x_2 - b_{12}\|_2^2 + k_{21}\|x_1 - x_2 - b_{21}\|_2^2 \\ f_2(x_{S_2}) &= k_{13}\|x_1 - x_3 - b_{13}\|_2^2 + k_{31}\|x_1 - x_3 - b_{31}\|_2^2 \\ f_3(x_{S_3}) &= k_{23}\|x_2 - x_3 - b_{23}\|_2^2 + k_{32}\|x_2 - x_3 - b_{32}\|_2^2\end{aligned}$$

Table 1 shows the parameter values adopted for the simulation results, while $A = \mathbf{1}\mathbf{1}'$, with $\mathbf{1} := [1, 1]'$, $\beta = 0.5$, $\alpha_k = 0.4$ for $k \leq 50$ and $0.4(k-50)^{-1/2}$ for $k > 50$.

k_{12}	k_{13}	k_{21}	k_{23}	k_{31}	k_{32}
0.3	0.3	0.3	0.3	0.3	0.3
b_{12}	b_{13}	b_{21}	b_{23}	b_{31}	b_{32}
$[1, 2]'$	$[3, 4]'$	$[1, 1]'$	$[2, 1]'$	$[5, 2]'$	$[2, 2]'$

Table 1: Parameters in numerical simulation

We compare the outcome of the distributed optimization protocol in (17) with the one shown in Fig. 7, where the same update equation is implemented by the cost centers, while gradient information communicated by the agents is filtered according to (8). Finally we compare our filter with the detection strategy provided in [29] and apply this strategy to the push-pull algorithm ([36]). In particular we consider the case of agent 3 being under attack/malicious after some steps, and communicating gradient information relative to a time-varying cost function. Two attack scenarios are considered, corresponding to an injection of strongly corrupt data (easier to detect), and one where the attack is stealthier and harder to detect.

7.1. Comparison for strong noise injection

In this attack scenario agent 3 is under attack after 50 time steps and provides gradient information relative to the function:

$$\begin{aligned} f_3(x) = & k_{31} \|x_1 - x_3 - b_{31} + [15 \cos(0.3k - 1), 0]'\|_2^2 \\ & + k_{32} \|x_2 - x_3 - b_{32} + [15 \sin(0.3k - 1), 0]'\|_2^2 \end{aligned} \quad (18)$$

Notice, in Fig. 8, the evolution of $|x_1|$, $|x_2|$ and $|x_3|$ and norm of related errors with respect to the true optimum based on the variables computed at the nodes S_1 , S_2 and S_3 , subject to the same noisy gradient data, both with and without the filters action. Notice how the filters significantly reduce oscillations in the consensus and distributed optimization protocol. Generally speaking, guaranteeing that the filtered data is consistent with some underlying time-invariant convex function does not necessarily reduce the optimization error; nevertheless, it is beneficial to ensure asymptotic convergence of the consensus and optimization protocols. The same attack handled by the detection scheme in [29] is shown in Fig. 9. A metric $S_{ij}(k)$ is used by agent i to detect the malicious neighbor. This metric compares each neighbor j 's gradient estimate to the average gradient in the rest of the neighborhood. Each agent keeps a track of these metrics for $k_T = 80$ steps and if the difference between any pair of neighbor agents' metrics surpass a threshold γ , $\|S_{ij}(k_T) - S_{im}(k_T)\|_2 \geq \gamma$, agent i severs tie with the neighbor j who possesses the highest magnitude of $S_{ij}(k_T)$. While the malicious agent is easily detected after time-step 80, the optimization still will discard the malicious agent (and implicitly all the clean data it provided) and converge to the minimum of the remaining two cost functions. Finally, to facilitate the comparison, we plot in Fig. 10 the averaged optimization error of the three methods as a function of the iteration k . We point out that, from the strictly computational point of view, the algorithm has a quadratically growing number of operations as a function of k . This is coherent with the computational time observed in numerical trials, as shown in

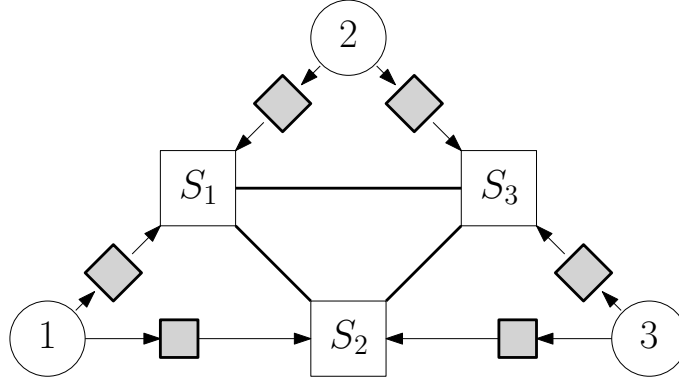


Figure 7: Distributed architecture with gradient filters (gray squares).

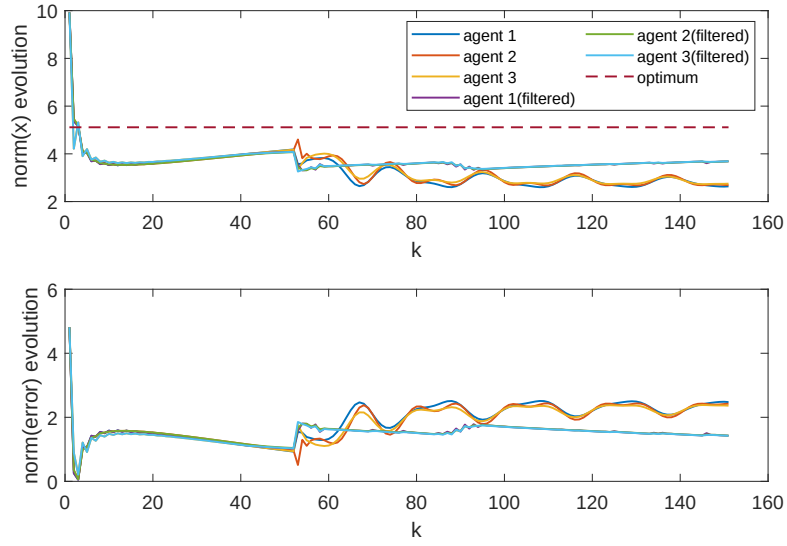


Figure 8: Evolution of $|x_1|, |x_2|, |x_3|$ and error norms at S_1, S_2 and S_3 .

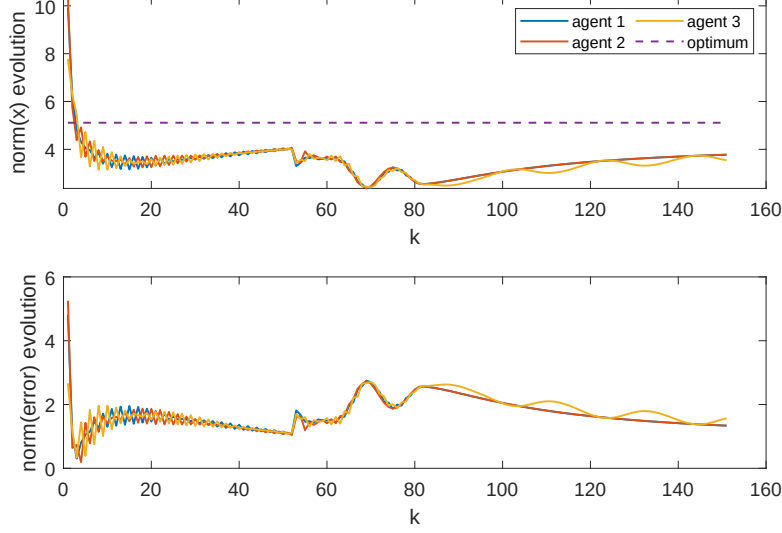


Figure 9: Evolution of $|x|$ and error norms at individual nodes for the detection scheme.

Fig. 11. If the computational burden grows too much within the considered horizon one might consider resetting the algorithm after a finite time interval.

7.2. Comparison for stealth noise injection

We assume agent 3 is acting maliciously (or is under attack) and starts changing its cost function after $k_m = 45$ steps according to:

$$f_3(x) = k_{31} \left\| x_1 - x_3 - [-5 - \log(5 + 5 \sin(0.1k)), 2]' \right\|_2^2 + k_{32} \|x_2 - x_3 - b_{32}\|_2^2 \quad (19)$$

where $k \geq k_m$ denotes the iteration step and e_1 is the first vector of the canonical basis in $\mathbb{R}^2$. By using the time-varying cost function, the result will be influenced and will start oscillating and hence preventing the algorithm from convergence. Simulation results (highlighting the norm of the variables and of the optimization errors) are shown in Fig.12 both for the nominal (unfiltered) and filtered algorithm. Moreover, in Fig.13 are shown the results provided by the detection scheme [29]. While the method provides some benefits the value of

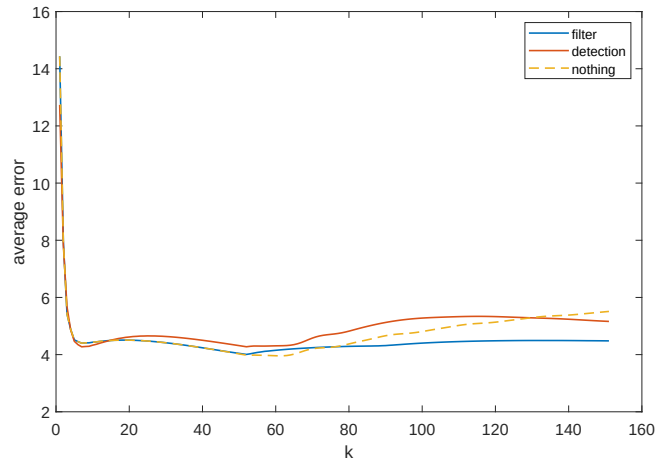


Figure 10: Average optimization errors.

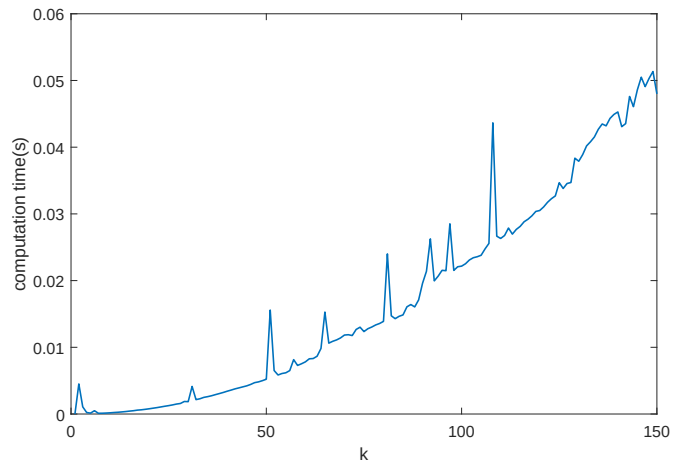


Figure 11: Total computational time, over all agents, as a function of k .

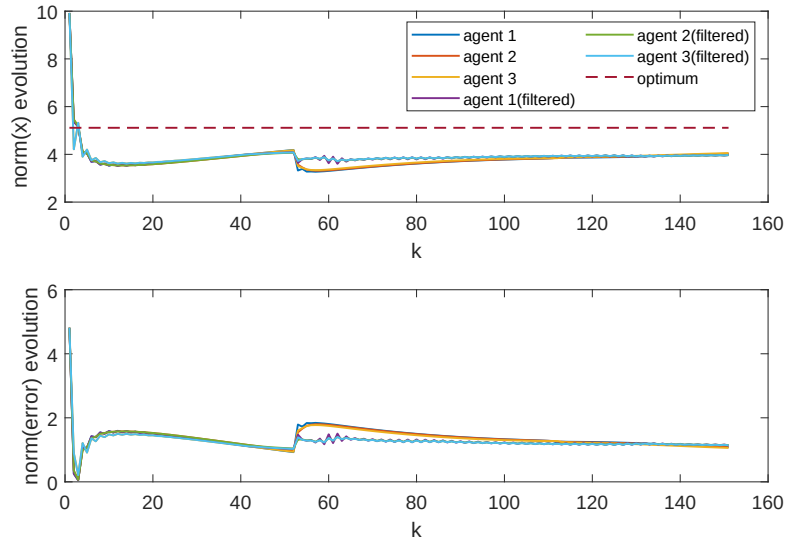


Figure 12: Evolution of $|x_i|$ variables at S_i and error norms.

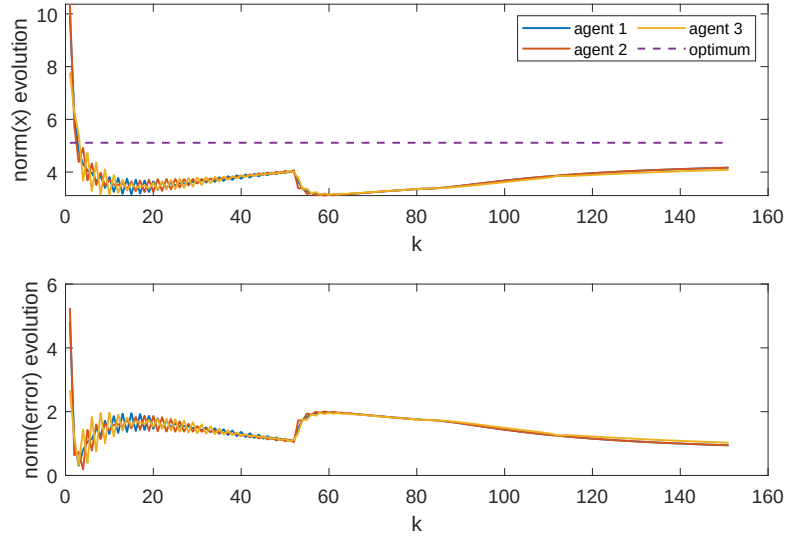


Figure 13: Evolution of $|x_i|$ and error norms with detection scheme.

the threshold needs to be fine tuned on the basis of the corresponding attack in-
 335 tensity. Moreover, the value of γ cannot be arbitrarily lowered as this inevitably
 yields false detections of malicious agents. This highlights the advantages of a
 method which does not rely on a priori assumptions on the intensity and nature
 of disturbances.

8. Conclusions

340 A novel filter was introduced to ensure consistency of gradient samples of
 a convex function communicated through a channel which is unreliable and/or
 under attack by a malicious agent. The filter ensures an output sequence of
 samples which is compatible with the a priori *convexity* assumption. More
 specifically, it minimally modifies the current gradient sample so as to ensure
 345 existence of at least one convex function compatible with the previous sequence
 of filtered gradients and the one currently considered. The proposed version
 of the algorithm updates a square matrix of dimension $t \times t$, where t is the
 index of the current gradient sample, and on this basis constructs the space of
 admissible gradient samples at the current time. An application of the filter to
 350 the context of distributed convex optimization is considered and simulated. One
 major benefit of this approach is that it retains the convergence properties of
 convexity-based optimization algorithms which operate on the basis of gradient
 information, even in the presence of arbitrary disturbances.

References

- 355 [1] R.T. Rockafellar, On the maximal monotonicity of subdifferential map-
 pings, Pacific Journal of Mathematics, 33, (1), (1970)
- [2] R.T. Rockafellar, Characterization of the subdifferentials of convex map-
 pings, Pacific Journal of Mathematics, 17, (3), (1966).
- [3] G. Notarstefano, I. Notarnicola, A. Camisa, Distributed Optimization for
 360 Smart Cyber-Physical Networks, Foundations and Trends in Systems and
 Control Series, (2019).

- [4] S. Manfredi, Multilayer Control of Networked Cyber-Physical Systems. Application to Monitoring, Autonomous and Robot Systems, Advances in Industrial Control, Springer International Publishing, (2017).
- 365 [5] J. Qin, S. Wang, Y. Kang, Q. Liu Circular formation algorithms for multiple nonholonomic mobile robots: an optimization-based approach, IEEE Trans. on Industrial Electronics; 66, (5), (2019).
- [6] J. Qin J, Y. Wan, X. Yu, F. Li, C. Li, Consensus-based distributed coordination between economic dispatch and demand response, IEEE Transactions on Smart Grid, 10, (4), (2019).
- 370 [7] Nedić, A. , & Ozdaglar, A. Distributed subgradient methods for multi-agent optimization. IEEE Transactions on Automatic Control, 54 (1), (2009).
- [8] Van Sy Mai, Eyad H. Abed, Distributed optimization over directed graphs with row stochasticity and constraint regularity. Automatica, 102, (2019).
- 375 [9] Z. Qiu, S. Liu, L. Xie. Necessary and sufficient conditions for distributed constrained optimal consensus under bounded input. International Journal of Robust and Nonlinear Control, 28 (6), (2018).
- [10] Buşoniu, L., De Bruin, T., Tolić, D., Kober, J. & Palunko, I. Reinforcement learning for control: Performance, stability, and deep approximators. Annual Reviews In Control. 46, (2018).
- 380 [11] A. Nedic, Distributed Gradient Methods for Convex Machine Learning Problems in Networks: Distributed Optimization, in IEEE Signal Processing Magazine, 37, (3), (2020).
- 385 [12] Lowy, A. & Razaviyayn, M. Private Federated Learning Without a Trusted Server: Optimal Algorithms for Convex Losses. arXiv, (2021).
- [13] Guo, S., Zhang, X., Yang, F., Zhang, T., Gan, Y., Xiang, T. & Liu, Y. Robust and Privacy-Preserving Collaborative Learning: A Comprehensive Survey, arXiv, (2021).

- 390 [14] Konečný, J., McMahan, H., Yu, F., Richtarik, P., Suresh, A. & Bacon, D. Federated Learning: Strategies for Improving Communication Efficiency. NIPS Workshop On Private Multi-Party Machine Learning. (2016).
- [15] N. Wu, F. Farokhi, D. Smith and M. A. Kaafar, The Value of Collaboration in Convex Machine Learning with Differential Privacy, 2020 IEEE Symposium on Security and Privacy (SP), (2020).
395
- [16] Zhang, S., Choromanska, A. & LeCun, Y. Deep learning with Elastic Averaging SGD. Advances In Neural Information Processing Systems. 28, (2015).
- [17] Tsitsiklis, J., Bertsekas, D. & Athans, M. Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms. 1984 American Control Conference. (1984).
400
- [18] Bottou, L. Online Algorithms and Stochastic Approximations. Online Learning And Neural Networks, (2012).
- [19] L. Su and N. H. Vaidya, Byzantine-Resilient Multiagent Optimization, IEEE Trans. on Automatic Control, 66, (5), (2021).
405
- [20] Angeli D., Manfredi S, A Resilient Consensus Protocol for Networks with Heterogeneous Confidence and Byzantine Adversaries, IEEE Control Systems Letters, 6, (2022).
- [21] C. Zhao, J. He and Q. -G. Wang, Resilient Distributed Optimization Algorithm Against Adversarial Attacks, IEEE Transactions on Automatic Control, 65, (10), (2020).
410
- [22] Iyanuoluwa Emiola, Laurent Njilla, Chinwendu Enyioha ,On Distributed Optimization in the Presence of Malicious Agents, arXiv, (2022).
- [23] E.D. Sontag, Remarks on input to state stability of perturbed gradient flows, motivated by model-free feedback control learning, Systems & Control Letters, 105138, (161), (2022).
415

- [24] R. Xin, A. K. Sahu, U. A. Khan and S. Kar, Distributed stochastic optimization with gradient tracking over strongly-connected networks, IEEE Conf. on Decision and Control (CDC), (2019).
- 420 [25] Peng Yi, Jinlong Lei, Yiguang Hong, Distributed resource allocation over random networks based on stochastic approximation, Systems & Control Letters, 114, (2018).
- [26] K. Srivastava and A. Nedić. Distributed asynchronous constrained stochastic optimization. IEEE Journal of Selected Topics in Signal Processing, 5, 425 (4), (2011).
- [27] P. Bianchi and J. Jakubowicz. Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization. IEEE Transactions on Automatic Control, 58, (2), (2013).
- [28] N. Chatzipanagiotis and M. M. Zavlanos. A distributed algorithm for convex constrained optimization under noise. IEEE Transactions on Automatic Control, 430 61, (9), (2016).
- [29] N. Ravi, A. Scaglione and A. Nedić, A Case of Distributed Optimization in Adversarial Environment, ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), (2019).
- 435 [30] D. Alistarh, Z. Allen-Zhu, and J. Li. Byzantine stochastic gradient descent. In NeurIPS, (2018).
- [31] P. Blanchard, R. Guerraoui, J. Stainer, et al. Machine learning with adversaries: Byzantine tolerant gradient descent. In NeurIPS, (2017).
- [32] R. Guerraoui, S. Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In ICML, (2018). 440
- [33] M. B Cohen, Y. Tat Lee, G. Miller, J. Pachocki, and A. Sidford. Geometric median in nearly linear time. In STOC, (2016).

- [34] E.M. Rudd, A. Rozsa, M. Gunther, T. E. Boulton, A Survey of Stealth
Malware Attacks, Mitigation Measures, and Steps Toward Autonomous
Open World Solutions. IEEE Communications Surveys & Tutorials, 19,
445 (2), (2017).
- [35] J.C. Bermond and C. Thomassen, Cycles in digraphs-a Survey, Journal of
Graph Theory, 5, (1), (1981).
- [36] S. Pu, W. Shi, J. Xu and A. Nedić, Push-Pull Gradient Methods for Dis-
tributed Optimization in Networks, in IEEE Transactions on Automatic
450 Control, 66, (1) (2021).