Distributed Change Detection [⋆]

Henrik Ohlsson [*,**] **Tianshi Chen** [*]
Sina Khoshfetrat Pakazad [*] **Lennart Ljung** [*]
S. Shankar Sastry [**]

[*] *Division of Automatic Control, Department of Electrical Engineering, Linköping University, Sweden, e-mail: ohlsson@isy.liu.se.*
[**] *Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, USA.*

Abstract: Change detection has traditionally been seen as a centralized problem. Many change detection problems are however distributed in nature and the need for distributed change detection algorithms is therefore significant. In this paper a distributed change detection algorithm is proposed. The change detection problem is first formulated as a convex optimization problem and then solved distributively with the alternating direction method of multipliers (ADMM). To further reduce the computational burden on each sensor, a homotopy solution is also derived. The proposed method have interesting connections with Lasso and compressed sensing and the theory developed for these methods are therefore directly applicable.

1. INTRODUCTION

The change detection problem is often thought of as a centralized problem. Many scenarios are however distributed and lack a central node or require a distributed processing. A practical example is a sensor network. It may be vulnerable to select one of the sensors as a central node. Moreover, it may be preferable if the sensor failing can be detected in a distributed manner. Another practical example is the monitoring of a fleet of agents (airplanes/UAVs/robots) of the same type, see *e.g.,* Chu et al. [2011]. The problem is how to detect if one or more agents start deviating from the rest. Theoretically, this can be done straightforwardly in a centralized manner. The centralized solution, however, poses many difficulties in practice. For instance, the communication between the central monitor to the agents and the computation capacity and speed of the central monitor is highly demanding due to a large number of agents in the fleet and/or the extremely large data sets to be processed, Chu et al. [2011]. Therefore, it is desirable to deal with the change detection problem in a distributed way.

In a distributed setting, there will be no central node. Each sensor or agent makes use of measurements from itself and the other sensors or agents to detect if it has failed or not. To tackle the problem, we first formulate the change detection problem as a convex optimization problem. We then solve the problem in a distributed manner using the so-called alternating direction method of multipliers (ADMM, see for instance Boyd et al. [2011]). The optimization problem turns out to have connections with the Lasso [Tibsharani, 1996] and compressive sensing [Candès et al., 2006, Donoho, 2006] and the theory developed for these methods are therefore applicable. To further reduce the computational burden on each sensor, a homotopy solution (see *e.g.,* Garrigues and El Ghaoui [2008]) is also studied. Finally, we show the effectiveness of the proposed method by a numerical example.

2. PROBLEM FORMULATION

The basic idea of the proposed method is to use system identification, in a distributed manner, to obtain a nominal model for the sensors or agents and then detect whether one or more sensors or agents start deviating from this nominal model.

To set up the notation, assume that we are given a sensor network consisting of N sensors. Denote the measurement from sensor i at time t by $y_i(t)$ and assume that there is a linear relation of the form

$$y_i(t) = \varphi_i^{\mathsf{T}}(t)\theta + e_i(t), \tag{1}$$

describing the relation between the measurable quantity $y_i(t) \in \mathcal{R}^n$ and known quantity $\varphi_i^T(t) \in \mathcal{R}^{n\times m}$. We will call $\theta \in \mathcal{R}^m$ the state. The state is related to the sensor reading through $\varphi_i(t)$. $e_i(t) \in \mathcal{R}^n$ is the measurement noise and assumed to be white Gaussian distributed with zero mean and variance σ_i^2. Moreover, $e_i(t)$ is assumed independent of that of $e_j(t)$, for all $i = 1, \dots, N$ and $j = 1, \dots, i-1, i+1, \dots, N$. At time t it is assumed that sensor i obtains $y_i(t)$ and knows $\varphi_i(t)$.

The problem is now, in a distributed manner, to detect a failing sensor. That is, detect if the relation (1) is no longer valid.

Remark 2.1. (Time varying state). A dynamical equation or a random walk type of description for the state θ can be incorporated. This is straightforward but for the sake of clarity and due to page limitations, this is not shown here. Actually, the only restriction is that θ does not vary over sensors (that is, not dependent on i).

[⋆] Ohlsson, Chen and Ljung are partially supported by the Swedish Research Council in the Linnaeus center CADICS and by the European Research Council under the advanced grant LEARN, contract 267381. Ohlsson is also supported by a postdoctoral grant from the Sweden-America Foundation, donated by ASEA's Fellowship Fund, and by a postdoctoral grant from the Swedish Research Council.

Remark 2.2. (Partially observed state). Note that (1) does not imply that the sensors need to measure all elements of θ. Some sensors can observe some parts and other sensors other parts.

Remark 2.3. (Time-varying network topology). That sensors are added and taken away from the network is a very realistic scenario. We will assume that N is the maximum number of sensors in the network and set $\varphi_i(t) = 0$ if sensor i is not present at time t.

Remark 2.4. (Multidimensional $y_i(t)$). For notational simplicity, from now on, we have chosen to let $y_i(t) \in \mathcal{R}$. However, the extension to multidimensional $y_i(t)$ is straightforward.

Remark 2.5. (Distributed system identification). The proposed algorithm could also be seen as a robust distributed system identification scheme. The algorithm computes, in a distributed manner, a nominal model using observations from several systems and is robust to systems deviating from the majority.

A straightforward way to solve the distributed change detection problem as posted here would be to

(1) locally, at each sensor, estimate θ
(2) broadcast the estimates and the error covariances
(3) at each sensor, fuse the estimates
(4) at each sensor, use a likelihood ratio test to detect a failing sensor (see *e.g.,* Willsky and Jones [1976], Willsky [1976])

This method will work fine as long as the number of measurements available at each sensor well exceeds m. Let us say that $\{(y_i(t), \varphi_i(t))\}_{t=T-T_i+1}^{T}$ is available at sensor i, $i = 1, \dots, N$. It is hence required that $T_1, T_2, \dots, T_N \gg m$. If $m > T_i$ for some $i = 1, \dots, N$, the method will however fail. That $m > T_i$ for some $i = 1, \dots, N$, is a very realistic scenario. T_i may for example be very small if new sensors may be added to the network at any time. The case $T_1, T_2, \dots, T_N \gg m$ was previously discussed in Chu et al. [2011].

Remark 2.6. (Sending all data). One may also consider to broadcast data and solve the full problem on each sensor. Sending all data available at time T may however be too much. Sensor i would then have to broadcast $\{(y_i(t), \varphi_i(t))\}_{t=T-T_i+1}^{T}$.

3. BACKGROUND

Change detection has a long history (see *e.g.,* Gustafsson [2001], Patton et al. [1989], Basseville and Nikiforov [1993] and references therein) but has traditionally been seen as a centralized problem. The literature on distributed or decentralized change detection is therefore rather small and only standard methods such as CUSUM and generalized likelihood ration (GLR) test have been discussed and extended to distributive scenarios (see *e.g.,* Tartakovsky and Veeravalli [2002, 2003]). The method proposed here has certainly a strong connection to GLR (see for instance Ohlsson et al. [2012]) and an extensive comparison is seen as future work.

The change detection algorithm proposed has also connections to compressive sensing and ℓ_1-minimization. There are several comprehensive review papers that cover the literature of compressive sensing and related optimization techniques in linear programming. The reader is referred to the works of Candès and Wakin [2008], Bruckstein et al. [2009], Loris [2009], Yang et al. [2010].

4. PROPOSED METHOD – BATCH SOLUTION

Assume that the data set $\{(y_i(t), \varphi_i(t))\}_{t=T-T_i+1}^{T}$ is available at sensor i, $i = 1, \dots, N$. Since (1) is assumed to hold for a functioning sensors, we would like to detect a failing sensor by checking if its likelihood falls below some threshold. What complicates the problem is that:

- θ is unknown,
- $m > T_i$ for some $i = 1, \dots, N$, typically.

We first solve the problem in a centralized setting.

4.1 Centralized Solution

Introduce θ_i for the state of sensor i, $i = 1, \dots, N$. Assume that we know that k sensors have failed. The maximum likelihood (ML) solution for θ_i, $i = 1, \dots, N$ (taking into account that $N-k$ sensors have the same state) can then be computed by

$$\min_{\theta_1,\dots,\theta_N,\theta} \sum_{i=1}^{N} \sum_{t=T-T_i+1}^{T} \|y_i(t) - \varphi_i^{\mathsf{T}}(t)\theta_i\|_{\sigma_i^2}^2 \tag{2a}$$

$$\text{subj. to } \left\| [\|\theta_1 - \theta\|_p \; \|\theta_2 - \theta\|_p \; \cdots \; \|\theta_N - \theta\|_p] \right\|_0 = k, \tag{2b}$$

with $\|\cdot\|_p$ being the p-norm, $p \geq 1$, and $\|\cdot\|_{\sigma_i^2}$ defined as $\|\cdot/\sigma_i\|_2$. The k failing sensors could now be identified as the sensors for which $\|\theta_i - \theta\|_p \neq 0$. It follows from basic optimization theory (see for instance Boyd and Vandenberghe [2004]) that there exists a $\lambda > 0$ such that

$$\min_{\theta_1,\dots,\theta_N,\theta} \sum_{i=1}^{N} \sum_{t=T-T_i+1}^{T} \|y_i(t) - \varphi_i^{\mathsf{T}}(t)\theta_i\|_{\sigma_i^2}^2 + \lambda \left\| [\|\theta_1 - \theta\|_p \; \|\theta_2 - \theta\|_p \; \cdots \; \|\theta_N - \theta\|_p] \right\|_0, \tag{3}$$

gives exactly the same estimate for $\theta_1, \dots, \theta_N, \theta$, as (2). However, both (2) and (3) are non-convex and combinatorial, and in practice unsolvable.

What makes (3) non-convex is the second term. It has recently become popular to approximate the zero-norm by its convex envelope. That is, to replace the zero-norm by the one-norm. This is in line with the reasoning behind Lasso [Tibsharani, 1996] and compressed sensing [Candès et al., 2006, Donoho, 2006]. Relaxing the zero-norm by replacing it with the one-norm leads to the convex criteria

$$\min_{\theta,\theta_1,\dots,\theta_N} \sum_{i=1}^{N} \sum_{t=T-T_i+1}^{T} \|y_i(t) - \varphi_i^{\mathsf{T}}(t)\theta_i\|_{\sigma_i^2}^2 + \lambda \sum_{i=1}^{N} \|\theta - \theta_i\|_p. \tag{4}$$

θ should be interpreted as the nominal model. Most sensors will have data that can be explained by the nominal model θ and the criterion (4) will therefore give $\theta_i = \theta$ for most i's. However, failing sensors will generate a data sequence that could not have been generated by the nominal model represented by θ and for these sensors, (4) will give $\theta_i \neq \theta$.

In (4), λ regulates the trade off between miss fit to the observations and the deviation from the nominal model θ. In practice, a large λ will make us less sensitive to noise but may also imply that we miss to detect a deviating sensor. However, a too small λ may in a noisy environment generate false alarms. λ should be seen as an application dependent design parameter. The estimates of (2) and (3) are indifferent to the choice of p. The estimate of (4) is not, however. In general, $p = 1$ is a good choice if one is interested in detecting changes in individual elements of sensors' or agents' states. If one only cares about detecting if a sensor or agent is failing, $p > 1$ is a better choice.

What is remarkable is that under some conditions on $\varphi_i(t)$ and the number of failures, the criterion (4) will work exactly as good as (2). That is, (4) and (2) will pick out exactly the same sensors as failing sensors. To examine when this happens theory developed in compressive sensing can be used. This is not discussed here but is a good future research direction.

4.2 Distributed Solution

Let us now apply ADMM (see *e.g.,* Boyd et al. [2011], [Bertsekas and Tsitsiklis, 1997, Sec. 3.4]) to solve the identification problem in a distributed manner. First let

$$Y_i = \begin{bmatrix} y_i(T-T_i+1) \\ y_i(T-T_i+2) \\ \vdots \\ y_i(T) \end{bmatrix}, \ \Phi_i = \begin{bmatrix} \varphi_i^\mathsf{T}(T-T_i+1) \\ \varphi_i^\mathsf{T}(T-T_i+2) \\ \vdots \\ \varphi_i^\mathsf{T}(T) \end{bmatrix}. \quad (5)$$

The optimization problem (4) can then be written as

$$\min_{\theta,\theta_1,\vartheta_1,\dots,\theta_N,\vartheta_N} \sum_{i=1}^N \|Y_i - \Phi_i\theta_i\|^2_{\sigma_i^2} + \lambda\|\vartheta_i - \theta_i\|_p, \quad (6a)$$

$$\text{subj. to} \quad \vartheta_i - \theta = 0, \quad i = 1,\dots,N. \quad (6b)$$

Let $x^\mathsf{T} = \begin{bmatrix} \theta_1^\mathsf{T} \dots \theta_N^\mathsf{T} \ \vartheta_1^\mathsf{T} \dots \vartheta_N^\mathsf{T} \end{bmatrix}$, and let $\nu^\mathsf{T} = \begin{bmatrix} \nu_1^\mathsf{T} \ \nu_2^\mathsf{T} \dots \nu_N^\mathsf{T} \end{bmatrix}$, be the Lagrange multiplier vector and ν_i be the Lagrange multiplier associated with the ith constraint $\vartheta_i - \theta = 0$, $i = 1,\dots,N$. So the augmented Lagrangian takes the following form

$$L_\rho(\theta, x, \nu) = \sum_{i=1}^N \|Y_i - \Phi_i\theta_i\|^2_{\sigma_i^2} + \lambda\|\vartheta_i - \theta_i\|_p \quad (7a)$$

$$+\nu_i^\mathsf{T}(\vartheta_i - \theta) + (\rho/2)\|\vartheta_i - \theta\|^2. \quad (7b)$$

ADMM consists of the following update rules

$$x^{k+1} = \operatorname*{argmin}_x L_\rho(\theta^k, x, \nu^k) \quad (8a)$$

$$\theta^{k+1} = (1/N)\sum_{i=1}^N (\vartheta_i^{k+1} + (1/\rho)\nu_i^k) \quad (8b)$$

$$\nu_i^{k+1} = \nu_i^k + \rho(\vartheta_i^{k+1} - \theta^{k+1}), \text{ for } i = 1,\dots,N. \quad (8c)$$

Remark 4.1. It should be noted that given θ^k, ν^k, the criterion $L_\rho(\theta^k, x, \nu^k)$ in (8a) is separable in terms of the pairs ϑ_i, θ_i, $i = 1,\dots,N$. Therefore, the optimization can be done separately, for each i, as

$$\vartheta_i^{k+1}, \theta_i^{k+1} = \operatorname*{argmin}_{\vartheta_i,\theta_i} \|Y_i - \Phi_i\theta_i\|^2_{\sigma_i^2} + \lambda\|\vartheta_i - \theta_i\|_p$$
$$+ (\nu_i^k)^\mathsf{T}(\vartheta_i - \theta^k) + (\rho/2)\|\vartheta_i - \theta^k\|^2. \quad (9)$$

Remark 4.2. (Boyd et al. [2011]). It is interesting to note that no matter what ν^1 is

$$\sum_{i=1}^N \nu_i^k = 0, \qquad k \geq 2. \quad (10)$$

To show (10), first note that

$$\sum_{i=1}^N \nu_i^{k+1} = \sum_{i=1}^N \nu_i^k + \rho\sum_{i=1}^N \vartheta_i^{k+1} - N\rho\theta^{k+1}, \quad k \geq 1. \quad (11)$$

Inserting θ^{k+1} into the above equation yields (10). So without loss of generality, further assume

$$\sum_{i=1}^N \nu_i^1 = 0. \quad (12)$$

Then the update on θ reduces to

$$\theta^{k+1} = (1/N)\sum_{i=1}^N \vartheta_i^{k+1}, \qquad k \geq 1. \quad (13)$$

As a result, in order to implement the ADMM in a distributed manner each sensor or system i should follow the steps below.

(1) Initialization: set θ^1, ν_i^1 and ρ.
(2) $\vartheta_i^{k+1}, \theta_i^{k+1} = \operatorname{argmin}_{\theta_i,\vartheta_i} L_\rho(\theta^k, x, \nu^k)$.
(3) Broadcast ϑ_i^{k+1} to the other systems (sensors), $j = 1,\dots,i-1,i+1,\dots,N$.
(4) $\theta^{k+1} = (1/N)\sum_{i=1}^N \vartheta_i^{k+1}$.
(5) $\nu_i^{k+1} = \nu_i^k + \rho(\vartheta_i^{k+1} - \theta^{k+1})$.
(6) If not converged, set $k = k+1$ and return to step 2.

To show that ADMM gives:

- $\theta^k - \vartheta_i^k \to 0$ as $k \to \infty$, $i = 1,\dots,N$.
- $\sum_{i=1}^N \|Y_i - \Phi_i\theta_i^k\|^2_{\sigma_i^2} + \lambda\|\vartheta_i^k - \theta_i^k\|_p \to p^*$, where p^* is the optimal objective of (4).

it is sufficient to show that the Lagrangian ($L_0(\theta, x, \nu)$, the augmented Lagrangian evaluated at $\rho = 0$) has a saddle point according to [Boyd et al., 2011, Sect. 3.2.1] (since the objective consists of closed, proper and convex functions). Let θ^*, x^* denote the solution of (4). It is easy to show that θ^*, x^* and $\nu = 0$ is a saddle point. Since $L_0(\theta, x, 0)$ is convex,

$$L_0(\theta^*, x^*, 0) \leq L_0(\theta, x, 0) \quad \forall\theta, x \quad (14)$$

and since $L_0(\theta^*, x^*, 0) = L_0(\theta^*, x^*, \nu)$, $\forall\nu$, θ^*, x^* and $\nu = 0$ must be a saddle point. ADMM hence converges to the solution of (4) in the sense listed above.

5. PROPOSED METHOD – RECURSIVE SOLUTION

To apply the above batch method to a scenario where we continuously get new measurements, we propose to re-run the batch algorithm every Tth sample time:

(1) Initialize by running the batch algorithm proposed in the previous section on the data available.
(2) Every Tth time-step, re-run the batch algorithm using the sT, $s \in N$, last data. Initialize the ADMM iterations using the estimates of θ and ν from the previous run. Considering the fact that faults occur rarely over time, the optimal solution for different data batches are often similar. As a result, by using the estimates of θ and ν from the previous run for initializing the ADMM algorithm can speed up the convergence of the ADMM algorithm considerably.

To add T new observation pairs, one could possibly use an extended version of the homotopy algorithm proposed by Garrigues and El Ghaoui [2008]. The homotopy algorithm presented in Garrigues and El Ghaoui [2008] was developed for including new observations in Lasso.

The ADMM algorithm presented in the previous section could also be altered to use single data samples upon arrival. In such a setup, instead of waiting for a collection or batch of measurements, the algorithm is updated upon arrival of new measurements. This can be done by studying the augmented Lagrangian of the problem. The augmented Lagrangian in (7) can also be written in normalized form as

$$\bar{L}_\rho(\theta, x, \bar{\nu}) = \sum_{i=1}^{N} \|Y_i - \Phi_i\theta_i\|^2_{\sigma_i^2} + \lambda\|\vartheta_i - \theta_i\|_p \\ +(\rho/2)\|\vartheta_i - (\theta - \bar{\nu}_i)\|^2 - (\rho/2)\|\bar{\nu}_i\|^2, \quad (15)$$

where $\bar{\nu}_i = \nu/\rho$. Hence, for $p = 2$, the update in (9) can be achieved by solving the following convex optimization problem, which can be written as a Second Order Cone Programming (SOCP) problem, [Boyd and Vandenberghe, 2004],

$$\min_{\theta_i,\vartheta_i,t} \quad \theta_i^\mathsf{T} H_i\theta_i - 2\theta_i^\mathsf{T} h_i + (\rho/2)\vartheta_i^\mathsf{T}\vartheta_i - \rho\vartheta_i^\mathsf{T}\bar{h}_i^k + \lambda s \\ \text{subj. to} \quad \|\theta_i - \vartheta_i\| \le s \quad (16)$$

where the following data matrices describe this optimization problem

$$H_i = \Phi_i^\mathsf{T}\Phi_i/\sigma_i^2,\ h_i = \Phi_i^\mathsf{T} Y_i/\sigma_i^2,\ \bar{h}_i^k = \theta^k - \bar{\nu}_i^k. \quad (17)$$

As can be seen from (17), among these matrices only H_i and h_i are the ones that are affected by the new measurements. Let y_i^{new} and φ_i^{new} denote the new measurements. Then H_i and h_i can be updated as follows

$$H_i \leftarrow H_i + \varphi_i^{new}\varphi_i^{new\mathsf{T}}/\sigma_i^2,\ h_i \leftarrow h_i + \varphi_i^{new} y_i^{new}/\sigma_i^2. \quad (18)$$

To handle single data samples upon arrival, step 2 of the ADMM algorithm should therefore be altered to:

(2) If there exits any new measurements, update H_i and h_i according to (18). Find $\vartheta_i^{k+1}, \theta_i^{k+1}$ by solving (16).

Remark 5.1. In order for this algorithm to be responsive to the arrival of the new measurements, it is required to have network-wide persistent communication. As a result this approach demands much higher communication traffic than the batch solution.

6. IMPLEMENTATION ISSUES

Step 2 of ADMM requires solving the optimization problem

$$\min_x \ L_\rho(\theta^k, x, \nu^k). \quad (19)$$

This problem is solved locally on each sensor once every ADMM iteration. What varies from one iteration to the next are the values for the arguments θ^k and ν^k. However, it is unlikely that θ^k and ν^k differ significantly from θ^{k+1} and ν^{k+1}. To take use of this fact can considerably ease the computational load on each sensor. We present two methods for doing this, warm start and a homotopy method.

The following two subsections are rather technical and we refer the reader not interested in implementing the proposed algorithm to Section 7.

6.1 Warm Start for Step 2 of the ADMM Algorithm

For the case where $p = 2$, at each iteration of the ADMM, we have to solve an SOCP problem, which is described in (16) and (17). However, in the batch solution, among the matrices in (17), only $\bar{h}_i^k$ changes with the iteration number. Therefore, if we assume that the vectors θ^k and ν_i^k do not change drastically from iteration to iteration, it is possible to use the solution for the problem in (16) at the kth iteration to warm start the problem at the $(k+1)$th iteration. This is done as follows.

The Lagrangian for the problem in (16) can be written as

$$L(\theta_i, \vartheta_i, s, z_i) = \theta_i^\mathsf{T} H_i\theta_i - 2\theta_i^\mathsf{T} h_i + (\rho/2)\vartheta_i^\mathsf{T}\vartheta_i - \rho\vartheta_i^\mathsf{T}\bar{h}_i^k \\ + \lambda s - \left\langle \begin{bmatrix} z_{i1} \\ z_{i2} \end{bmatrix}, \begin{bmatrix} s \\ x_i - \vartheta_i \end{bmatrix} \right\rangle, \quad (20)$$

for all $\|z_{i2}\| \le z_{i1}$. By this, the optimality conditions for the problem in (16), can be written as

$$2H_i\theta_i - 2h_i - z_{i2} = 0 \quad (21a)$$
$$\rho\vartheta_i - \rho\bar{h}_i^k + z_{i2} = 0 \quad (21b)$$
$$\lambda - z_{i1} = 0 \quad (21c)$$
$$\|z_{i2}\| \le z_{i1} \quad (21d)$$
$$\|\theta_i - \vartheta_i\| \le s \quad (21e)$$
$$z_{i1}s + z_{i2}^\mathsf{T}(\theta_i - \vartheta_i) = 0. \quad (21f)$$

where $z_i = \begin{bmatrix} z_{i1} \\ z_{i2} \end{bmatrix}$ [Boyd and Vandenberghe, 2004]. Let θ_i^*, ϑ_i^*, t^* and z_i^* be the primal and dual optimums for the problem in (16) at the kth iteration and let $\bar{h}_i^{k+1} = \bar{h}_i^k + \Delta\bar{h}_i$. These can be used to generate a good warm start point for the solver that solves (16). As a result, by (21) the following vectors can be used to warm start the solver

$$\begin{aligned} \theta_i^w &= \theta_i^* \\ \vartheta_i^w &= \vartheta_i^* + \Delta\bar{h}_i \\ z_i^w &= z_i^* \\ s^w &= s^* + \Delta s \end{aligned} \quad (22)$$

where Δs should be chosen such that

$$\begin{aligned} &\|\theta_i^w - \vartheta_i^w\| \le s^* + \Delta s \\ &z_{i1}^w(s^* + \Delta s) + z_{i2}^{w\mathsf{T}}(\theta_i^w - \vartheta_i^w) = \mu, \end{aligned} \quad (23)$$

for some $\mu \ge 0$.

6.2 A Homotopy Algorithm

Since it is unlikely that θ^k and ν^k differ significantly from θ^{k+1} and ν^{k+1}, one can use the previously computed solution x^k and through a homotopy update the solution instead of resolving (19) from scratch every iteration. We will in the following assume that $p = 1$ and leave the details for $p > 1$.

First, define

$$\delta_i = \vartheta_i - \theta_i. \quad (24)$$

The optimization objective of (9) is then

$$g_i^k(\theta_i, \delta_i) \triangleq \|Y_i - \Phi_i\theta_i\|^2_{\sigma_i^2} + \lambda\|\delta_i\|_1 + (\nu_i^k)^\mathsf{T}(\delta_i + \theta_i - \theta^k) \\ +(\rho/2)\|\delta_i + \theta_i - \theta^k\|^2. \quad (25)$$

It is then straightforward to show that the optimization problem (9) is equivalent to

$$\theta_i^{k+1}, \delta_i^{k+1} = \operatorname*{argmin}_{\theta_i,\delta_i} g_i^k(\theta_i, \delta_i). \quad (26)$$

Moreover,

$$\vartheta_i^{k+1} = \delta_i^{k+1} + \theta_i^{k+1}. \quad (27)$$

Now, compute the subdifferential of $g_i(\theta_i,\delta_i)$ *w.r.t.* θ_i and δ_i. Simple calculations show that

$$\begin{aligned}\partial_{\theta_i} g_i^k(\theta_i,\delta_i) =&\nabla_{\theta_i} g_i^k(\theta_i,\delta_i) = -2/\sigma_i^2 Y_i^\top \Phi_i \\ &+2/\sigma_i^2 \theta_i^\top \Phi_i^\top \Phi_i + (\nu_i^k)^\top + \rho(\theta_i+\delta_i-\theta^k)^\top, \quad (28a)\\ \partial_{\delta_i} g_i^k(\theta_i,\delta_i) =&\lambda \partial \|\delta_i\|_1 + (\nu_i^k)^\top + \rho(\delta_i+\theta_i-\theta^k)^\top. \quad (28b)\end{aligned}$$

A necessary condition for the global optimal solution $\theta_i^{k+1}, \delta_i^{k+1}$ of the optimization problem (9) is

$$0 \in \partial_{\theta_i} g_i^k\left(\theta_i^{k+1},\delta_i^{k+1}\right), \quad (29a)$$
$$0 \in \partial_{\delta_i} g_i^k\left(\theta_i^{k+1},\delta_i^{k+1}\right). \quad (29b)$$

It follows from (28a) and (29a) that

$$\theta_i^{k+1} = R_i^{-1}\left(h_i - \nu_i^k/2 - (\rho/2)(\delta_i-\theta^k)\right) \quad (30)$$

where we have let

$$R_i = \Phi_i^\top \Phi_i/\sigma_i^2 + (\rho/2)I, \qquad h_i = \Phi_i^\top Y_i/\sigma_i^2. \quad (31)$$

With (30), the problem now reduces to how to solve (29b). Inserting (30) into (28b), and $Q_i \triangleq I - (\rho/2)R_i^{-1}$, yields

$$\partial_{\delta_i} g_i^k(\theta_i^{k+1},\delta_i) = \lambda\partial\|\delta_i\|_1 + \rho h_i^\top R_i^{-1} + (\nu_i^k - \rho\theta^k + \rho\delta_i)^\top Q_i$$

Now, replace θ^k with $\theta^k + t\Delta\theta^{k+1}$ and ν_i^k with $\nu^k + t\Delta\nu^{k+1}$. Let

$$\begin{aligned}G_i^k(t) = &\lambda\partial\|\delta_i\|_1 + \rho h_i^\top R_i^{-1} + (\nu_i^k - \rho\theta^k + \rho\delta_i)^\top Q_i\\ &+ t(\Delta\nu_i^{k+1} - \rho\Delta\theta^{k+1})^\top Q_i. \quad (32)\end{aligned}$$

$\partial_{\delta_i} g_i^k(\theta_i^{k+1},\delta_i)$ hence equals $G_i^k(0)$. Let $\delta_i^k(t) = \text{argmin}_{\delta_i} G_i^k(t)$. It follows that

$$\delta_i^{k+1} = \delta_i^k(0), \quad \delta_i^{k+2} = \delta_i^k(1) \quad (33)$$

Assume now that δ_i^{k+1} has been computed and that the elements have been arranged such that the q first elements are nonzero and the last $m-q$ zero. Let us write

$$\delta_i^k(0) = \begin{bmatrix}\bar{\delta}_i^k \\ 0\end{bmatrix}. \quad (34)$$

We then have that (both the sign and $|\cdot|$ taken element-wise)

$$\partial\|\delta_i^k(0)\|_1 = \left[\text{sign}(\bar{\delta}_i^k)^\top \ v^\top\right], \quad v \in \mathcal{R}^{m-q}, \ |v| \le 1. \quad (35)$$

Hence, that $0 \in G_i^k(0)$ is equivalent with

$$\lambda\text{sign}(\bar{\delta}_i^k)^\top + \rho h_i^\top \bar{R}_i^{-1} + (\nu_i^k - \rho\theta^k + \rho\delta_i^k)^\top \bar{Q}_i = 0 \quad (36a)$$
$$\lambda v^\top + \rho h_i^\top \tilde{R}_i^{-1} + (\nu_i^k - \rho\theta^k + \rho\delta_i^k)^\top \tilde{Q}_i = 0, \quad (36b)$$

with

$$R^{-1} = \left[\bar{R}^{-1} \ \tilde{R}^{-1}\right], \ \bar{R}^{-1} \in \mathcal{R}^{m\times q}, \tilde{R}^{-1} \in \mathcal{R}^{m\times m-q}, \quad (37)$$
$$Q = \left[\bar{Q} \ \tilde{Q}\right], \ \bar{Q} \in \mathcal{R}^{m\times q}, \tilde{Q} \in \mathcal{R}^{m\times m-q}. \quad (38)$$

It can be shown that the partition (34) or the support of $\delta_i^k(t)$ will stay unchanged for $t \in [0\, t^*)$, $t^* > 0$ (see Lemma 1 of Garrigues and El Ghaoui [2008]). It hence follows that for $t \in [0\, t^*]$

$$\begin{aligned}\bar{\delta}_i^k(t)^\top = &-(\lambda\text{sign}(\bar{\delta}_i^k)^\top/\rho + h_i^\top \bar{R}_i^{-1})\hat{Q}_i^{-1} + (\theta^k - \nu_i^k/\rho)^\top \bar{Q}\hat{Q}^{-1}\\ &+ t(\Delta\theta^{k+1} - \Delta\nu_i^{k+1}/\rho)^\top \bar{Q}\hat{Q}^{-1}, \quad (39)\end{aligned}$$

where we introduced $\hat{Q}$ for the $q \times q$ matrix made up of the top q rows of $\bar{Q}$. We can also compute

$$\begin{aligned}v^\top(t)/\lambda = &-\rho h_i^\top \tilde{R}_i^{-1} + (\rho\theta^k - \nu_i^k - \rho\delta_i^k(t))^\top \tilde{Q}_i\\ &+ t(\rho\Delta\theta^k - \Delta\nu_i^k)^\top \tilde{Q}_i\\ &= -\rho h_i^\top \tilde{R}_i^{-1} + (\rho\theta^k - \nu_i^k)^\top \tilde{Q}_i - \rho(\bar{\delta}_i^k(t))^\top \check{Q}_i\\ &+ t(\rho\Delta\theta^k - \Delta\nu_i^k)^\top \tilde{Q}_i\\ &= -\rho h_i^\top \tilde{R}_i^{-1} + (\rho\theta^k - \nu_i^k)^\top \tilde{Q}_i - \rho(\bar{\delta}_i^k(0))^\top \hat{Q}_i\\ &+ t(\rho\Delta\theta^k - \Delta\nu_i^k)^\top(\tilde{Q}_i - \bar{Q}\hat{Q}^{-1}\check{Q}_i)\end{aligned}$$

where $\check{Q}_i$ was used to denote the top q rows of $\tilde{Q}_i$. Now to find t^*, we notice that both $\bar{\delta}_i^k(t)$ and v are linear functions of t. We can hence compute the minimal t that:

- Make one or more elements of v equal to -1 or 1 or/and
- make one or more elements of $\bar{\delta}_i^k(t)$ equal to 0.

This minimal t will be t^*. At t^* the partition (34) changes:

- Elements corresponding to v-elements equal to -1 or 1 should be included in $\bar{\delta}_i^k$.
- Elements of $\bar{\delta}_i^k(t)$ equal to 0 should be fixed to zero.

Given the solution $\delta_i^k(t^*)$, we can now continue in a similar way as above to compute $\delta_i^k(t)$, $t \in [t^*, t^{**}]$. The procedure continues until $\delta_i^k(1)$ has been computed. Due to space limitations we have chosen to not give a summary of the algorithm. We instead refer the interested reader to download the implementation available from `http://www.rt.isy.liu.se/~ohlsson/code.html`.

7. NUMERICAL ILLUSTRATION

For the numerical illustration, we consider a network of $N = 10$ sensors with $\theta_1 = \theta_2 = \cdots = \theta_{10}$ being random samples from $N(0, I) \in \mathcal{R}^{20}$. We assume that there exist 10 batches of measurements, each consisting of 15 samples. The regressors were unit Gaussian noise and the measurement noise variance was set to one. In the scenario considered in this section, we simulate failures in sensors 2 and 5, upon the arrival of the 4th and 6th measurement batches, respectively. This is done by changing the 5th component of θ_2 by multiplying it by 5 and changing the 8th component of θ_5 by shifting (adding) it by 5. It is assumed that the faults are persistent.

With $\lambda = \rho = 20$ both the centralized, the ADMM and the ADMM with homotopy give identical results (up to the 2nd digit, 10 iterations were used in ADMM). As can be seen from Figures 1-3 the result correctly detected that the 2nd and 5th sensors are faulty. In addition as can be seen from Figures 2 and 3, ADMM and ADMM with homotopy show that for how many data batches the sensors remained faulty. Also the results detect which elements from θ_2 and θ_5 deviated from the nominal value. In this example (using the ADMM algorithm), each sensor had to broadcast $m \times$ ADMM iterations $\times$ number of batches $= 20 \times 10 \times 10 = 2000$ scalar values. If instead all data would have been shared, each sensor would have to broadcast $(m+1) \times T \times$ number of batches $= (20+1) \times 15 \times 10 = 3150$ scalar values. Using proposed algorithm, the traffic over the network can hence be made considerably lower while keeping the performance of a centralized change detection algorithm. Using the Homotopy algorithm (or warm start) to solve step 2 of the ADMM algorithm will not affect the traffic over the network, but could lower the computational burden on the sensors. It is also worth noting that the

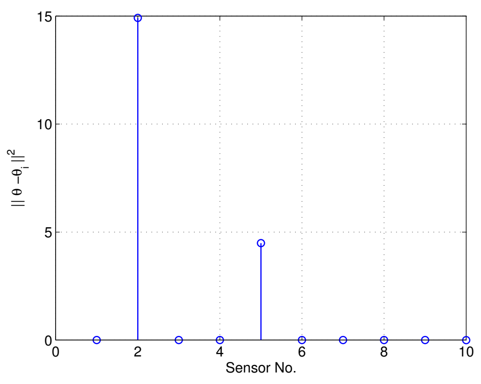

Fig. 1. Results from the centralized change detection. As can be seen sensors 2 and 5 are detected to be faulty.

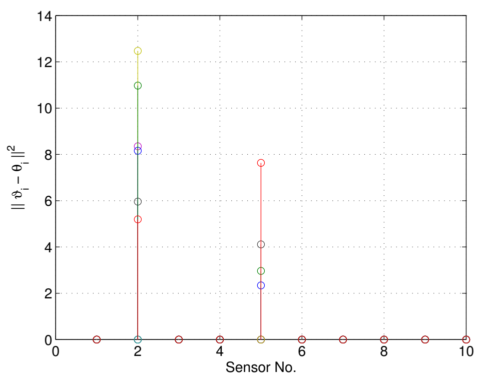

Fig. 2. Results from the ADMM batch solution. Sensors 2 and 5 have been detected faulty for 6 and 4 batches.

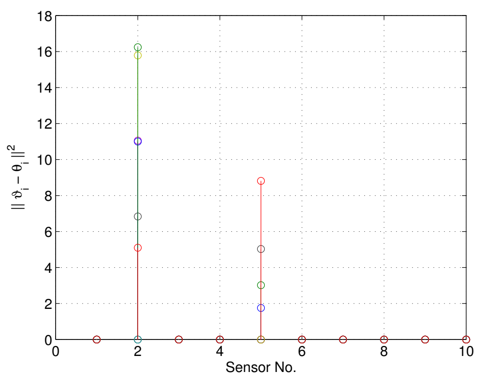

Fig. 3. Results from the Homotopy solution. Sensors 2 and 5 have been detected faulty for 6 and 4 batches.

classical approach using likelihood ration test, as described in Section 3, would not work on this example since $20 = m > T = 15$.

8. CONCLUSION

This paper has presented a distributed change detection algorithm. Change detection is most often seen as a centralized problem. As many scenarios are naturally distributed, there is a need for distributed change detection algorithms. The basic idea of the proposed distributed change detection algorithm is to use system identification, in a distributed manner, to obtain a nominal model for the sensors or agents and then detect whether one or more sensors or agents start deviating from this nominal model. The proposed formulation takes the form of a convex optimization problem. We show how this can be solved distributively and present a homotopy algorithm to easy the computational load. The proposed formulation has connections with Lasso and compressed sensing and theory developed for these methods are therefore directly applicable.

REFERENCES

M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes – Theory and Application*. Prentice-Hall, Englewood Cliffs, NJ, 1993.

D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

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

A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.

E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, March 2008.

E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52:489–509, February 2006.

E. Chu, D. Gorinevsky, and S. Boyd. Scalable statistical monitoring of fleet data. In *Proceedings of the 18th IFAC World Congress*, pages 13227–13232, Milan, Italy, August 2011.

D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.

P. Garrigues and L. El Ghaoui. An homotopy algorithm for the lasso with online observations. In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS)*, 2008.

F. Gustafsson. *Adaptive Filtering and Change Detection*. Wiley, New York, 2001.

I. Loris. On the performance of algorithms for the minimization of ℓ_1-penalized functionals. *Inverse Problems*, 25:1–16, 2009.

H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. Smoothed state estimates under abrupt changes using sum-of-norms regularization. *Automatica*, 48(4):595–605, 2012.

R. Patton, P. Frank, and R. Clark. *Fault Diagnosis in Dynamic Systems – Theory and Application*. Prentice Hall, 1989.

A. Tartakovsky and V. Veeravalli. Quickest change detection in distributed sensor systems. In *Proceedings of the 6th International Conference on Information Fusion*, pages 756–763, Cairns, Australia, July 2003.

A.G. Tartakovsky and V.V. Veeravalli. An efficient sequential procedure for detecting changes in multichannel and distributed systems. In *Proceedings of the Fifth International Conference on Information Fusion*, pages 41–48, 2002.

R. Tibsharani. Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society B (Methodological)*, 58(1): 267–288, 1996.

A. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12:601–611, 1976.

A. Willsky and H. Jones. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic Control*, 21(1):108–112, February 1976.

A. Yang, A. Ganesh, Y. Ma, and S. Sastry. Fast ℓ_1-minimization algorithms and an application in robust face recognition: A review. In *ICIP*, 2010.