# Distributed Change Detection Based on a Consensus Algorithm

Srdjan S. Stanković[1], Nemanja Ilić[1], Miloš S. Stanković[2], Karl Henrik Johansson[2]

[1]*Faculty of Electrical Engineering, University of Belgrade, Belgrade, Serbia*

[2]*School of Electrical Engineering, Royal Institute of Technology, 100-44 Stockholm, Sweden*

**NecSys'10**

## Motivation

Change detection is one of typical tasks of sensor networks; possibility to test the decision variables **at any node in the network** and **in real time** is often desirable.

• In the classical multi-sensor detection schemes the local sensors send all their data to other sensors, and ultimately to a fusion center - the decision variables are tested only at predefined fusion nodes.

• Most of the recent attempts to apply consensus techniques to the distributed detection problem assume that the dynamic agreement process starts after all data have been collected - inapplicable to real time change detection problems.

## Contribution

A novel algorithm is proposed for *distributed change detection* while monitoring the environment through a wireless sensor network.

• All the nodes in the network generate local decision variables by recursive schemes belonging to the *geometric moving average control charts*, applicable in **real time**.

• A dynamic consensus scheme with preselected asymmetric communication gains is applied; an algorithm which asymptotically provides nearly equal behavior of all the nodes is obtained (*i.e.*, **any node** can be selected for testing the decision variable w.r.t. a pre-specified threshold).

## Algorithm

• Network with *n* nodes, each node collects measurements and generates at each discrete time instant *t* a scalar quantity $x_i(t)$, directly or as a result of local signal processing; $\{x_i(t)\}$ are considered as mutually independent stationary random sequences with means $m_i$ and covariances $r_i(\tau)$.

• Global decision function for the whole network

$$s_c(t+1) = \alpha\, s_c(t) + (1-\alpha)\sum_{i=1}^{n}\omega_i x_i(t+1), \quad s_c(0)=0, \quad 0<\alpha<1,$$

where $\omega_i = k\theta_i^1\sigma_i^{-2}$ are the components of the vector $\omega^T = k\theta^{1T}\Sigma^{-1}$ ($k = (\sum_{i=1}^n w_i)^{-1}$).

•The basic assumption: nodes in the network are connected in such a way that the $n\times n$ matrix *C* represents the weighted adjacency matrix for the underlying graph representing the network, and that *C* is *row stochastic.*

• The proposed algorithm generates the vector decision function $s(t) = [s_1(t)\cdots s_n(t)]^T$ of the network:

$$s(t+1) = \alpha\, Cs(t) + (1-\alpha)Cx(t+1), \quad s(0)=0, \qquad \text{where } x(t) = [x_1(t)\cdots x_n(t)]^T.$$

• Consensus matrix *C* performs for each node "convexification" of the neighboring states and enforces in such a way consensus between the nodes.

## Convergence analysis

• Assumptions:

A1) *C* has the eigenvalue 1 with multiplicity 1;

$\Rightarrow$ $C^i$ converges to a nonnegative row stochastic matrix with equal rows;

A2) $\lim_{i\to\infty} C^i = \mathbf{1}\omega^T$;

$\Rightarrow$ *C* can be constructed by solving the linear equation $\omega^T C = \omega^T$ under the constraints that some of the elements of *C* are equal to zero and that it is row stochastic.

• Error is defined as $e(t) = s(t) - \mathbf{1}s_c(t) = (I-\mathbf{1}\omega^T)s(t) = (1-\alpha)\sum_{i=0}^{t-1}\alpha^i\widetilde{C}^{i+1}\widetilde{x}(t-i)$,
where $\widetilde{C} = C - \mathbf{1}\omega^T$, $\widetilde{x}(t) = (I-\mathbf{1}\omega^T)x(t)$.

• $s(t)$ as the estimator of $s_c(t)$ is, in general, biased; $m_e = E\{e(t)\} = 0$ only when $m_i = m_j,\ i,j=1,\dots,n$; the bias is smaller when $\alpha$ is closer to 1 ($E\{e(t)\} \sim (1-\alpha)$).

• The focus of the analysis is placed on the mean-square error matrix,
defined as $Q(t) = E\{e(t)e(t)^T\} - m_e(t)m_e(t)^T = (1-\alpha)^2\Phi(t)^T\widetilde{R}(t)\Phi(t)$, where
$\Phi(t) = [\alpha^{t-1}\widetilde{C}^t \vdots \alpha^{t-2}\widetilde{C}^{t-1} \vdots\cdots\vdots \alpha^0\widetilde{C}]^T$, $\widetilde{R}(t) = R(t) - m_X m_X^T$, $R(t) = E\{X(t)X(t)^T\}$,
$X(t) = [x(1)^T\cdots x(t)^T]^T$, $m_X = E\{X(t)\}$; further, $\widetilde{R}(t) = [R_{ij}]$, where
$R_{ij} = \mathrm{diag}\{r_1(i-j),\dots,r_n(i-j)\}$.

*Theorem 1.*
Let assumptions A1) and A2) hold, together with A3) $\max_i \sum_{\tau=0}^{\infty}|r_i(\tau)| < K < \infty$.
Then,

$$\max_{i,j} Q_{ij}(t) = O((1-\alpha)^2).$$

*Proof:*

• The n-dimensional quadratic form $y^TQ(t)y = (1-\alpha)^2y^T\Phi(t)^T\widetilde{R}(t)\Phi(t)y$ is analyzed.

• The expression $y^T\Phi(t)^T\Phi(t)y$ is in the form of a sum of terms containing $y^T\widetilde{C}^i\widetilde{C}^{iT}y$;

• From assumptions A1) and A2) $\Rightarrow \widetilde{C}$ has the same eigenvalues as C, except for the eigenvalue 1 which is replaced by 0 $\Rightarrow$ modules of all of its eigenvalues are less than 1;

• The recursion $P(t+1) = \widetilde{C}P(t)\widetilde{C}^T$, $P(0) = I$ is considered ( $P(t) = \widetilde{C}^t\widetilde{C}^{tT}$);

• The column vectors of $P(t)$ are concatenated to obtain an $n^2$-vector $\mathrm{vec}\{P(t)\}$
$\Rightarrow \mathrm{vec}\{P(t+1)\} = (\widetilde{C}\otimes\widetilde{C})\mathrm{vec}\{P(t+1)\}$ ("$\otimes$" denotes the Kronecker's product);

• From A1) and A2) $\Rightarrow |\lambda(\widetilde{C}\otimes\widetilde{C})|_{\max} = \lambda_M < 1$ (eigenvalues of $\widetilde{C}\otimes\widetilde{C}$ take values from the cross products of the eigenvalues of $\widetilde{C}$ ) $\Rightarrow \|P(t)\| \le k_p\lambda_M^t$
$\Rightarrow y^T\widetilde{C}^i\widetilde{C}^{iT}y \le k_p\lambda_M^i\|y\|^2$ and $y^T\Phi(t)^T\widetilde{R}(t)\Phi(t)y \le \|y\|^2k'K\sum_{i=0}^{\infty}\alpha^i\lambda_M^{i+1} \le K_1 < \infty$

• By choosing $y = e_i$ ( $e_i$ denotes the *n*-vector of zeros with only i-th entry equal to one)
$\Rightarrow Q_{ii}(t) \le K_1(1-\alpha)^2$; furthermore, $|Q_{ij}(t)| \le \max_i Q_{ii}(t)$. Q.E.D.

**ACKNOWLEDGEMENTS**: This work was partially supported by the EU project PRODI, the Swedish Research Council, the Swedish Strategic Research Foundation, the Swedish Governmental Agency for Innovation Systems and the EU project FeedNetBack.

## Distributed detection based on time averaging

• The recursive algorithms with constant coefficient $\alpha$ are essentially tracking algorithms with exponential forgetting, able to cope with change detection phenomena.

• In the case when $\alpha$ is a function of time tending to 1 when $t$ tends to infinity, the algorithms are not directly suitable for change detection purposes.

*Theorem 2.* Let $\alpha$ be replaced by $\alpha(t+1) = 1-\gamma(t+1)$ , and let the assumptions A1), A2) and A3) be satisfied, together with:
A4) $\gamma(t)$ is a non-increasing sequence satisfying
$\gamma(t) > 0,\ \lim_{t\to\infty}\gamma(t) = 0,\ \sum_{t=1}^{\infty}\gamma(t) = \infty$.
Then,
$\|Q(t)\| = o(1)$.

*Theorem 3.* Under the assumptions of Theorem 2 and with $\gamma(t) = \frac{1}{t}$ we have
$\|Q(t)\| = o(t^{-2})$
while $\mathrm{var}\{s_c(t)\} = O(t^{-1})$.

## Simulation results

• Network with *n* = 10 nodes is considered, where the means are randomly taken from the interval [0, 1], and variances randomly taken from the interval [0.5, 1.5] (means are zero in the case of no change). The moment of change is chosen to be *t* = 200.

• Communication gains are obtained by solving the linear equation $\omega^T C = \omega^T$.

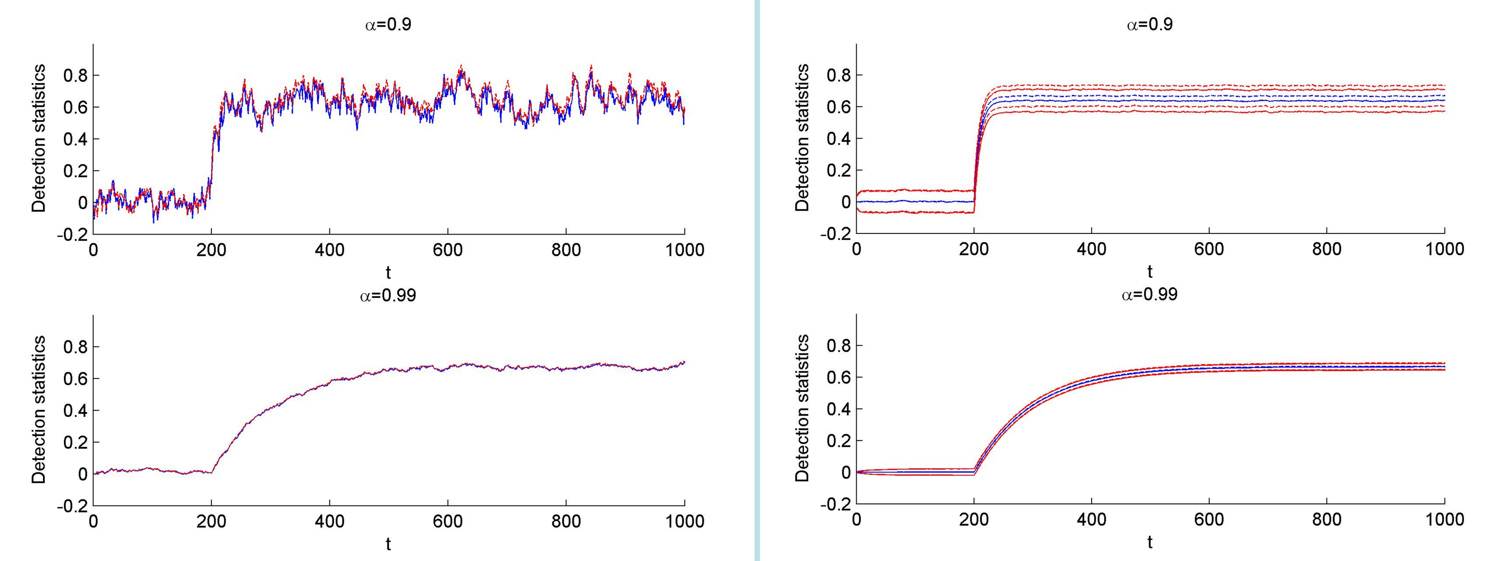
• *e.g.*

$$\omega_i = \frac{\theta_i^1\sigma_i^{-2}}{\sum_{i=1}^n\theta_i^1\sigma_i^{-2}} \Rightarrow C =$$

( $\lim_{i\to\infty} C^i = \mathbf{1}\omega^T$ )

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.4500 | 0 | 0 | 0.1291 | 0 | 0 | 0.1314 | 0.1683 | 0 | 0.1212 |
| 0 | 0.2761 | 0.2930 | 0 | 0 | 0 | 0.0942 | 0.2955 | 0 | 0.0411 |
| 0 | 0.2770 | 0.2944 | 0 | 0 | 0 | 0.0909 | 0.2997 | 0 | 0.0380 |
| 0.0176 | 0 | 0 | 0.1352 | 0 | 0.1609 | 0 | 0.2664 | 0.3233 | 0.0966 |
| 0 | 0 | 0 | 0 | 0.2226 | 0.2569 | 0 | 0 | 0.5205 | 0 |
| 0 | 0 | 0 | 0.1003 | 0.1076 | 0.1247 | 0.1129 | 0.2396 | 0.3149 | 0 |
| 0.0109 | 0.2191 | 0.2265 | 0 | 0 | 0.1213 | 0.1195 | 0.2287 | 0 | 0.0741 |
| 0.0013 | 0.2396 | 0.2570 | 0.0572 | 0 | 0.0869 | 0.0651 | 0.2689 | 0 | 0.0240 |
| 0 | 0 | 0 | 0.0903 | 0.1080 | 0.1371 | 0 | 0 | 0.6646 | 0 |
| 0.0253 | 0.1942 | 0.1976 | 0.1341 | 0 | 0 | 0.1427 | 0.2000 | 0 | 0.1062 |

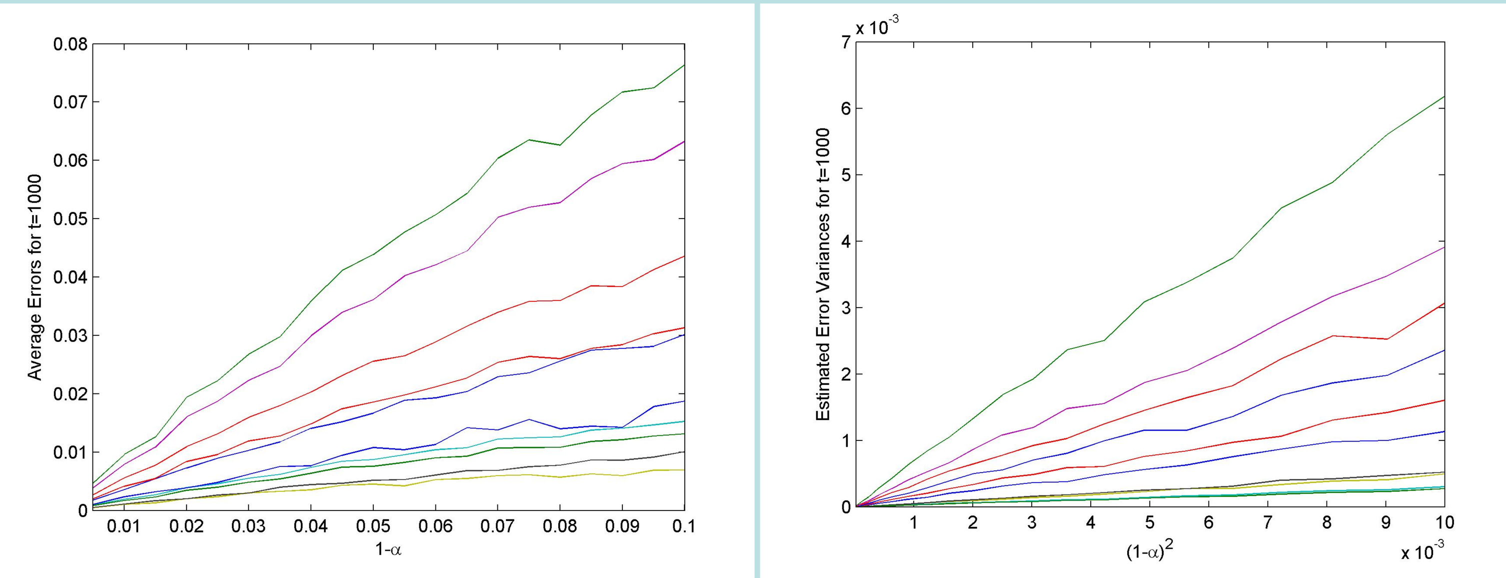
• The proposed algorithm effectively achieves very similar behavior of all of the nodes, with local decision functions getting closer to the global decision function as $\alpha \to 1$.




Decision functions – one realization: for one node (blue) and global (red)

Decision functions – mean +/- one standard deviation: for one node (solid lines) and global (dashed lines)

•The values of $|E\{e(t)\}|$ and $E\{e(t)e(t)^T\}_{ii}$ are estimated for different values of $\alpha$ for *t* = 1000 using 1000 Monte Carlo runs.

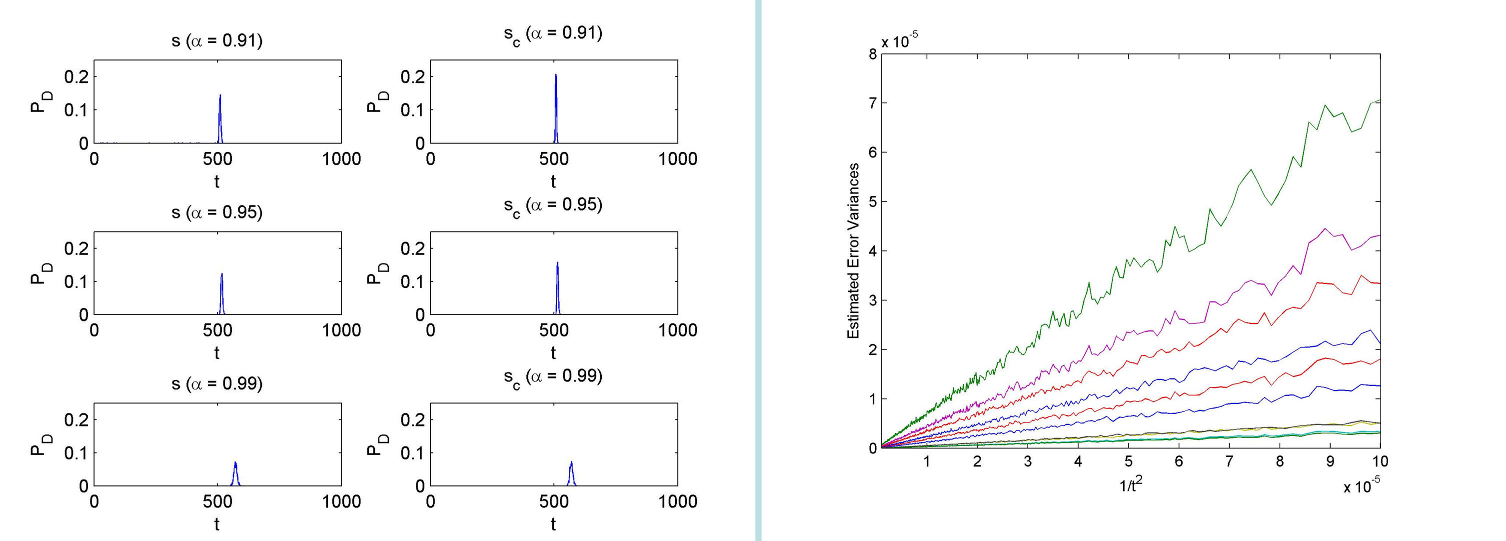



Average errors as functions of $\alpha$

Estimated error variances as functions of $\alpha$

• The distribution of the alarm times at which a detection occurs is estimated for different values of $\alpha$ (the moment of change is *t* = 500, the threshold is $\lambda_c = 0.5k\theta^{1T}\Sigma^{-1}\theta^1$).

• In the time averaging case the estimates of $E\{e(t)e(t)^T\}_{ii}$ are calculated using 1000 Monte Carlo runs.




The distribution of the alarm times for different values of $\alpha$: for one node (left) and global (right).

Estimated error variances as functions of time

**FURTHER WORK**: Generalization of the presented results to the case of stochastic time varying consensus matrices and application of the same methodology to the recursive Generalized Likelihood Ratio (GLR) algorithm.