Distributed sensor and actuator reconfiguration for fault-tolerant networked control systems

Herdeiro Teixeira, A.M.; Araujo, Jose; Sandberg, Henrik; Johansson, Karl H.

DOI
10.1109/TCNS.2017.2732158

Publication date
2017

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Control of Network Systems

Distributed sensor and actuator reconfiguration for fault-tolerant networked control systems

André M. H. Teixeira, José Araújo, Henrik Sandberg and Karl H. Johansson

***Abstract*—In this paper, we address the problem of distributed reconfiguration of networked control systems upon the removal of misbehaving sensors and actuators. In particular, we consider systems with redundant sensors and actuators cooperating to recover from faults. Reconfiguration is performed while minimizing a steady-state estimation error covariance and a quadratic control cost. A model-matching condition is imposed on the reconfiguration scheme. It is shown that the reconfiguration and its underlying computation can be distributed. Using an average dwell-time approach, the stability of the distributed reconfiguration scheme under finite-time termination is analyzed. The approach is illustrated in a numerical example.**

I. Introduction

Modern control systems are often operated over large-scale, complex networked infrastructures such as power networks, building automation systems, power plants and transportation systems. The proliferation of low-cost embedded systems with radio capabilities has enabled the deployment of systems with increased performance and flexibility. However, these systems become increasingly complex and must be efficiently designed and operated. Several steps have been taken in this direction, in the development of resilient and fault-tolerant architectures and technologies [1], [2] and plug-and-play control [3], [4], [5]. In this paper, we focus on distributed sensor and actuator reconfiguration in over-sensed and over-actuated networked control systems with a high degree of redundancy. In the event of malfunctioning actuators, sensors, or other system components, control systems may exhibit poor performances or even become unstable [2], [6]. Thus, the design of fault-tolerant control systems is of major importance. Examples of safety-critical systems that must be resilient to faults and cyberattacks include power networks, aircrafts, nuclear power plants and chemical plants.

Since the 1970s, much research has been conducted in fault-tolerant control systems, fault detection and diagnosis (FDD) and reconfigurable control [7], [2], [8], [9], [10], [1], [11]. FDD deals with the identification of faults [10], [1], [12], while reconfigurable control proposes methods to reconfigure a system after a faulty component has been detected and disabled. The objectives of reconfiguration are generally to recover stabilization of the system, maintaining the same state trajectory (also known as model-matching), achieving the same equilibrium point or minimizing the loss in performance inflicted by the fault. Model-matching reconfiguration, in particular, has been the focus of much of the research in this area [8]. Many types of faults in sensors, actuators and other system components have been considered in both linear and nonlinear systems. However, the vast majority of the solutions rely on a centralized approach [13], [14], [15], [16], [17]. Due to the increased complexity and size of current control systems, such techniques may be impractical [18], [6]. Through the increased computation and communication capabilities of embedded devices in these systems, FDD can technically move from a centralized implementation to a distributed one. However, distributed FDD and reconfiguration to enable distributed fault-tolerant systems has been much less explored. The architecture of such systems is discussed in [19], [20], [21], while in [22] a distributed FDD is employed to perform a centralized reconfiguration. To the best of our knowledge, distributed reconfiguration has not yet been addressed in the literature.

In this paper, we address the problem of distributed reconfiguration for networked control systems with misbehaving sensors and actuators by exploiting the existing redundancy. Assuming that the sensor and actuator redundancy is high enough to guarantee perfect model-matching of the nominal dynamics with only healthy sensors, we propose a distributed algorithm to perform the reconfiguration. The proposed distributed method guarantees closed-loop stability and minimizes the steady-state estimation error covariance and a linear-quadratic control cost under faults and cyberattacks while achieving model-matching: the desired closed-loop estimation error and dynamics remain the same before and after removing the misbehaving devices. The distributed algorithm is shown to converge to the optimal solution asymptotically. Additionally, the stability of the closed-loop system is analyzed when the distributed reconfiguration algorithms terminates in finite-time.

The rest of this paper is organized as follows. Section II presents the system architecture and formulates the problem. The centralized solution to the reconfiguration problem is presented in Section III. In Section IV it is shown that the reconfiguration can be distributed among the sensor or actuator nodes and an efficient algorithm is devised. For faults occurring sufficiently far apart in time, thus satisfying a given average dwell-time condition, stability properties of the system under the proposed distributed reconfiguration scheme are given in Section V. Finally, numerical examples illustrate

A. M. H. Teixeira is with the Department of Engineering Systems and Services, Delft University of Technology, Delft, the Netherlands. Email: andre.teixeira@tudelft.nl.

J. Araújo is with Ericsson Research, Stockholm, Sweden. Email: jose.araujo@ericsson.com.

H. Sandberg and K. H. Johansson are with the ACCESS Linnaeus Centre, KTH Royal Institute of Technology, Stockholm, Sweden. E-mail: {hsan, kallej}@kth.se.

The first and second authors contributed equally to this work. This work is supported by the Knut and Alice Wallenberg Foundation and the Swedish Research Council under Grants 2013-5523 and 2014-6282.

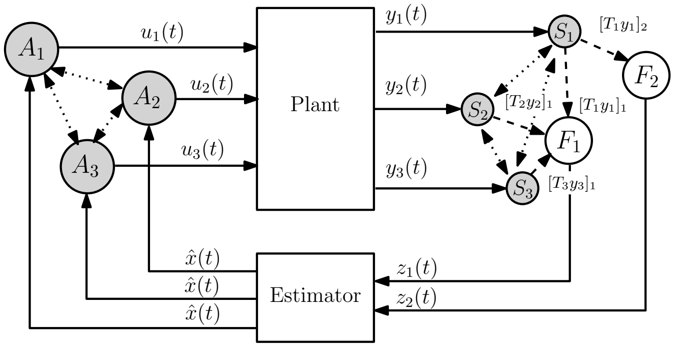


Fig. 1: Networked control system with a network of sensors S_1, S_2 and S_3, aggregator nodes F_1 and F_2 and actuators A_1, A_2 and A_3. Sensors and actuators are responsible for reconfiguring themselves when system failures occur.

the distributed reconfiguration methods in Section VI and Section VII concludes this paper.

A. Notation

The Kronecker product of matrix A and B is denoted as $A \otimes B$ and the vectorization operation as $\operatorname{vec}(A)$. A matrix A is denoted as positive definite if $A \succ 0$ and positive semi-definite when $A \succeq 0$. The trace of matrix A is denoted as $\operatorname{tr}(A)$. For a vector x, $\|x\| = \sqrt{x^\top x}$ denotes the Euclidean norm of x. Given a matrix A, $\|A\|_2 = \max_{u \neq 0} \frac{\|Au\|}{\|u\|}$ denotes the induced 2-norm of A, while $\|A\|_F = \left(\operatorname{tr}(A^\top A)\right)^{\frac{1}{2}}$ corresponds to its Frobenius norm. Let $\kappa(A) = \|A\|_2 \|A^\dagger\|_2$ denote the condition number of matrix A, and $A^\dagger$ the pseudoinverse of A. The notation $|\cdot|$ represents the cardinality of a set, and $\mathcal{A} \setminus \mathcal{B}$ denotes the set obtained by removing set $\mathcal{B} \subseteq \mathcal{A}$ from set $\mathcal{A}$.

A network is represented by an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The edge $e_k = (i, j) \in \mathcal{E}$ indicates that nodes i and j can exchange information. Denote $\mathcal{N}_i = \{j | j \neq i, (i, j) \in \mathcal{E}\}$ as the neighbor set of node i.

II. Problem formulation

The architecture of the networked control system considered in this work is depicted in Fig. 1. This architecture has two networks, one of sensors and one of actuators. Each network has a certain level of redundancy, which means that nominal operation can be maintained in spite of some components being removed. The precise meaning of redundancy in our setup will be given later in this section. Each sensor or actuator is able to exchange information with its neighbors within the network. In typical applications, such as building automation and industrial process control, a large number of sensors is expected to be deployed. To reduce the sensor-to-controller communication, the information from the sensor nodes is fused at aggregator nodes, which connect to the estimator. The estimator is responsible for computing the state-estimate to be broadcasted to the actuators in the network, which then compute the control input values. The individual components of the system are described below.

A. System model

Suppose the plant is modeled by a stochastic linear time-invariant differential equation,

$$\mathrm{d}x(t) = Ax(t)\,\mathrm{d}t + B\Gamma_u(t)u(t)\,\mathrm{d}t + \mathrm{d}w(t), \tag{1}$$

$$\mathrm{d}y(t) = \Gamma_y(t)\left(Cx(t)\,\mathrm{d}t + \mathrm{d}v(t)\right), \tag{2}$$

with a state $x(t) \in \mathbb{R}^n$, $y(t) \in \mathbb{R}^p$ and $u(t) \in \mathbb{R}^m$ are the measurement vector and input vector, respectively, with redundancy in their components, and $w(t) \in \mathbb{R}^n$ and $v(t) \in \mathbb{R}^p$ are independent Wiener processes with uncorrelated increments. The incremental covariances are $W\,\mathrm{d}t$ and $V\,\mathrm{d}t$, respectively. Moreover, processes $w(t)$ and $v(t)$ are assumed to be mutually uncorrelated [23].

The sensor nodes apply a local linear transformation to the measurements and transmit these values to aggregation nodes, which compute $z(t) \in \mathbb{R}^s$ as the fusion of the sensor data

$$\mathrm{d}z(t) = T\,\mathrm{d}y(t) = T\Gamma_y Cx(t)\,\mathrm{d}t + T\Gamma_y\,\mathrm{d}v(t), \tag{3}$$

where $T \in \mathbb{R}^{s \times p}$ is the aggregation matrix, with $s \leq p$, and $z(t)$ is transmitted to the estimator.

We consider the presence of misbehaving sensors and actuators, which could be acting according to different types of failures such as outages [16], partial degradation and loss of effectiveness [17], incipient faults [24], or even controlled by malicious cyber adversaries [25], [26]. Furthermore, we suppose that misbehaving devices are detected and isolated using suitable FDD schemes [1], [10], [12], after which they are removed from the system. Once the misbehaving devices have been removed, reconfiguration of the closed-loop system takes place, which is the focus of this work.

The removal of misbehaving sensors and actuators is modelled by the diagonal matrices $\Gamma_y(t) \in \mathbb{R}^{p \times p}$ and $\Gamma_u(t) \in \mathbb{R}^{m \times m}$, respectively, with $[\Gamma_y(t)]_{ii} = \gamma_{y_i}(t) \in \{0, 1\}$ and $[\Gamma_u(t)]_{ii} = \gamma_{u_i}(t) \in \{0, 1\}$. Here $\gamma_{y_i}(t)$ ($\gamma_{u_i}(t)$) represents the status of sensor (actuator) i at time t, where $\gamma_{y_i}(t) = 1$ ($\gamma_{u_i}(t) = 1$) means that the sensor (actuator) is healthy, while $\gamma_{y_i}(t) = 0$ ($\gamma_{u_i}(t) = 0$) indicates sensor (actuator) has been disabled. The system is initially under nominal conditions, hence $\Gamma_y(t) = I$ and $\Gamma_u(t) = I$ for $t < t_0$.

For the sake of clarity of the presentation, all misbehaving devices are assumed to be removed simultaneously at time $t = t_0$ and remain unchanged thereafter, which allows the time argument to be omitted. Note, however, that the methods devised in this paper directly apply to the non-simultaneous case, by running the proposed reconfiguration algorithm sequentially with the occurrence of each new fault, which is further investigated in Section V.

The sensor and actuator networks are represented by the connected and undirected graphs $\mathcal{G}_y(\mathcal{V}_y, \mathcal{E}_y)$ with $|\mathcal{V}_y| = p$ vertices and $\mathcal{G}_u(\mathcal{V}_u, \mathcal{E}_u)$ with $|\mathcal{V}_u| = m$ vertices, respectively. For simplicity of presentation, we assume that each aggregator node is connected to all sensor nodes. The set of sensor and actuator nodes is defined as $\mathcal{V} \triangleq \mathcal{V}_y \cup \mathcal{V}_u$, whereas we denote $\mathcal{V}^f \subseteq \mathcal{V}$ as the set of misbehaving nodes that have been removed and we let the set of healthy nodes be $\mathcal{V}^h \triangleq \mathcal{V} \setminus \mathcal{V}^f$.

We assume that the controller is given by the continuous-time linear-quadratic Gaussian (LQG) controller [23]. Let the pair (TC, A) be observable and (A, B) be controllable. Next, we describe the controller and estimator design under nominal conditions with $\Gamma_u = I$ and $\Gamma_y = I$. For LQG control, the feedback gain is obtained as the minimizer of the control cost $J_c \triangleq \lim\limits_{\tau\to\infty} J_c(\tau)$, where

$$J_c(\tau) \triangleq \frac{1}{\tau}\int_0^\tau \mathbf{E}\left\{x(t)^\top Q x(t) + u(t)^\top R u(t)\right\} \; \mathrm{d}t, \tag{4}$$

and $Q \succeq 0$ and $R \succ 0$ are weight matrices. We assume R is diagonal. The optimal LQ controller is given by

$$u(t) = -K\hat{x}(t) = -R^{-1}B^\top P\hat{x}(t), \tag{5}$$

where $\hat{x}(t)$ is the state estimate and P the solution to the Riccati equation $A^\top P + PA - PBR^{-1}B^\top P + Q = 0$. The estimate is computed by the Kalman-Bucy filter [23] as follows

$$\mathrm{d}\hat{x}(t) = (A - LTC)\hat{x}(t)\,\mathrm{d}t + Bu(t)\,\mathrm{d}t + L\,\mathrm{d}z(t), \tag{6}$$

with $L = \Sigma C^\top T^\top (TVT^\top)^{-1}$, where $\Sigma = \lim_{t\to\infty} \mathbf{E}\{e(t)e(t)^\top\}$ is the steady-state covariance matrix of the estimation error $e(t) = \hat{x}(t) - x(t)$ given by the Riccati equation $A\Sigma + \Sigma A^\top - \Sigma C^\top T^\top (TVT^\top)^{-1} TC\Sigma + W = 0$. The Kalman-Bucy filter minimizes the expected mean-squared error, which we denote as the estimation cost function:

$$J_e \triangleq \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau \mathbf{E}\left\{e(t)^\top e(t)\right\}\mathrm{d}t. \tag{7}$$

From now on we drop the time argument (t) when it is clear from the context.

B. Reconfiguration problem

Consider a scenario where several misbehaving sensor and actuator nodes have been disabled, yielding $\Gamma_u \neq I$ and $\Gamma_y \neq I$. A possible corrective action is to modify the aggregation matrix T and feedback matrix K so that only the remaining healthy sensors and actuators are used to guarantee a certain level of performance of the system. Let $\tilde{u} \in \mathbb{R}^m$ and $\tilde{z} \in \mathbb{R}^s$ denote the reconfigured control and sensor fusion signals:

$$\begin{aligned} \mathrm{d}\tilde{z} &= \tilde{T}\,\mathrm{d}y = \tilde{T}\Gamma_y Cx\,\mathrm{d}t + \tilde{T}\Gamma_y\,\mathrm{d}v, \\ \tilde{u} &= -\tilde{K}\hat{x}. \end{aligned} \tag{8}$$

Denote $\tilde{A}_c(\tilde{K}) = A - B\Gamma_u\tilde{K}$ and $\tilde{A}_e(\tilde{T}) = A - L\tilde{T}\Gamma_y C$ as the system matrices for the closed-loop dynamics of the system and estimator, respectively. The objective of the reconfiguration is to achieve model-matching [15], [8] for both the estimation dynamics and the closed-loop system dynamics by computing $\tilde{T}$ and $\tilde{K}$ after the removal of sensors and actuators, respectively. Model-matching is a common reconfiguration goal in fault-tolerant systems, as it guarantees that the original system dynamics are preserved even in the presence of faults. The definition of model-matching reconfiguration is as follows. Let us denote the closed-loop estimator dynamics before the fault as $A_e = A - LTC$ and the nominal closed-loop system matrix as $A_c = A - BK$. Then, *model-matching on the estimation error dynamics* is achieved if $\tilde{A}_e(\tilde{T}) = A_e$ for some new aggregation matrix $\tilde{T}$. *Model-matching on the closed-loop system dynamics* is achieved if $\tilde{A}_c(\tilde{K}) = A_c$ for some new feedback gain matrix $\tilde{K}$.

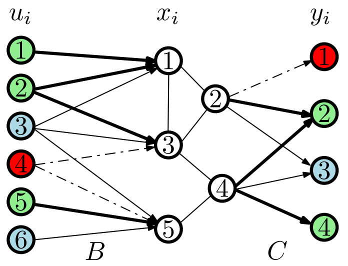


Fig. 2: Digraph representation of a system with high sensing and actuation redundancy. Faulty actuators and sensors (u_4 and y_1) are depicted in red and with dot-dashed edges. Nodes used to achieve perfect model-matching are represented in green.

A possible structure of a system with sufficiently high redundancy to allow perfect model-matching independently of K and T is illustrated in Figure 2. For instance, the input u_4 can be compensated by u_2 and u_5, as they affect x_3 and x_5, respectively. However, since u_2 also affects x_1, the use of u_2 must in turn be compensated by u_1. In short, denoting b_i as the i-th column of B, the structural system in Figure 2 has enough redundancy to ensure that there exist scalars α_1, α_2, and α_5 such that $\alpha_1 b_1 + \alpha_2 b_2 + \alpha_5 b_5 = b_4$. In other terms, the actuation redundancy ensures that $\mathrm{Im}(B) \equiv \mathrm{Im}(B\Gamma_u)$.

However, by taking the gain matrices into consideration, less redundancy can be considered, as in the following assumption.

Assumption 1: The sensor and actuator networks have sufficient redundancy such that model-matching is feasible when sensors and actuators are removed, i.e., $\mathrm{Im}(BK) \subseteq \mathrm{Im}(B\Gamma_u)$ and $\mathrm{Im}(C^\top T^\top) \subseteq \mathrm{Im}(C^\top \Gamma_y)$.

Although the perfect model-matching conditions may seem restrictive in classical control systems, large-scale networked control systems indeed have a large number of redundant components that may satisfy Assumption 1, as in the case of application examples such as distributed control of wind-farms [27], farming and livestock systems [3], smart grids with multiple distributed energy resources [26], and building management systems [25].

In case model-matching would not be feasible, i.e., A_e or A_c would no longer be achievable with the healthy nodes, different admissible closed-loop matrices must be considered. After new feasible matrices A_e and A_c have been computed, the methods proposed in this paper could be readily applied.

As the model-matching constraints are under-determined, i.e., they admit multiple solutions, we propose to find the model-matching solutions that minimize certain quadratic costs. In particular, the cost function for the sensor reconfiguration is the quadratic estimation cost (7)

$$J_e(\tilde{T}) = \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau \mathbf{E}\left\{\tilde{e}^\top \tilde{e}\right\} \; \mathrm{d}t, \tag{9}$$

where $\tilde{e}$ is the estimation error after the misbehaving sensors have been detected and removed. Furthermore, we define the objective function of the actuator reconfiguration as the quadratic control cost for the reconfigured control input

$$J_c(\tilde{K}) = \lim_{\tau\to\infty} \frac{1}{\tau} \int_0^\tau \mathbf{E}\left\{ x^\top \left(Q + \tilde{K}^\top \Gamma_u R \Gamma_u \tilde{K} \right) x \right\} \; \mathrm{d}t,$$
$$\text{s.t.} \quad \dot{x} = (A - B\Gamma_u \tilde{K})x, \tag{10}$$

where the expectation is taken with respect to the initial condition $x(0)$, which is a zero-mean Gaussian random variable with the positive definite covariance matrix $R_0 = \mathbf{E}\left\{x(0)x(0)^\top\right\}$.

The sensor and actuator networked reconfiguration problem is to find the reconfigured aggregation matrix $\tilde{T}$ and feedback gain matrix $\tilde{K}$ that minimize the estimation (9) and control cost (10), respectively, subject to the model-matching condition. The sensor reconfiguration can be re-formulated as

$$\begin{aligned} \min_{\tilde{T}} \quad & J_e(\tilde{T}) \\ \text{s.t.} \quad & A - L\tilde{T}\Gamma_y C = A - LTC, \end{aligned} \tag{11}$$

while the actuator reconfiguration problem is

$$\begin{aligned} \min_{\tilde{K}} \quad & J_c(\tilde{K}) \\ \text{s.t.} \quad & A - B\Gamma_u \tilde{K} = A - BK. \end{aligned} \tag{12}$$

The solution to these optimization problems may be achieved in a centralized or distributed manner. Next, we describe a centralized approach to solve them. Later, we propose an efficient distributed solution based solely on local information exchange among sensor nodes and actuators nodes. In both cases, we neglect the computation times and consider that the solutions are computed instantaneously with respect to the process dynamics. In Section V we analyze the stability properties of the proposed distributed algorithm when the reconfiguration is not instantaneous.

III. Centralized Sensor and Actuator Reconfiguration

We now tackle the centralized sensor and actuator reconfiguration problems. Their solutions are derived and the centralized reconfiguration mechanisms are illustrated.

A. *Centralized sensor reconfiguration*

The optimal solution to (11) can be characterized as follows.

Proposition 1: The solution to the sensor reconfiguration problem (11) is

$$\tilde{T}^\star = TC(C^\top V^{-1}\Gamma_y C)^\dagger C^\top \Gamma_y V^{-1}. \tag{13}$$

In order to prove Proposition 1 we use the following lemma.

Lemma 1: Optimization problem (11) is equivalent to

$$\begin{aligned} \min_{\tilde{T}} \quad & \mathrm{tr}\left((W + L\tilde{T}\Gamma_y V \Gamma_y \tilde{T}^\top L^\top) Z_e \right) \\ \text{s.t.} \quad & LTC = L\tilde{T}\Gamma_y C \\ & 0 = A_e^\top Z_e + Z_e A_e + I. \end{aligned} \tag{14}$$

Proof: The proof is given in the Appendix. ■

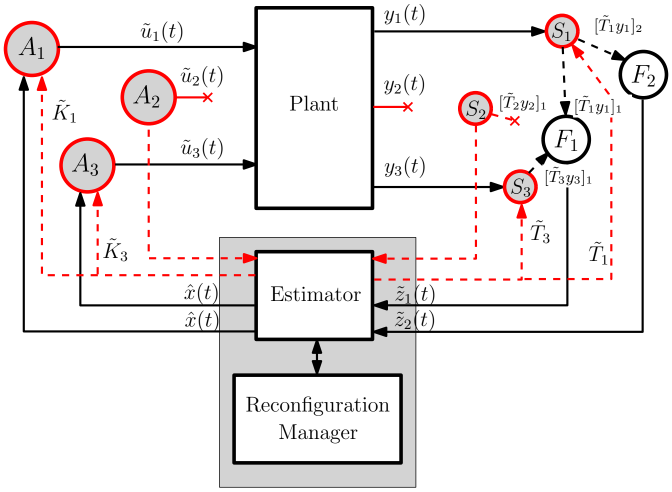


Fig. 3: Networked control system with centralized sensor and actuator reconfiguration. Faults are reported by the sensors and actuators to the centralized estimator. Red dashed arrows represent the transmission of information related to faults.

We now derive the optimal solution to (14), which is also the solution to the sensor reconfiguration problem (11).

Proof of Proposition 1: Consider the optimization problem (14), which is convex. Note that the second equality constraint is a Lyapunov equation with the Hurwitz system matrix A_e, determined by the model-matching condition. Hence, the variable Z_e is uniquely defined by the constraint and can be computed before hand. The Lagrangian function for (14) is $\mathcal{L}(\tilde{T}, \Lambda) = \mathrm{tr}\left(\left(W + L\tilde{T}\Gamma_y V \Gamma_y \tilde{T}^\top L^\top \right) Z_e \right) + \mathrm{tr}\left(\Lambda^\top \left(LTC - L\tilde{T}\Gamma_y C \right) \right)$, where $\Lambda \in \mathbb{R}^{n\times n}$ represents the Lagrange multipliers. Using the trace derivative expressions, the Karush-Kuhn-Tucker (KKT) optimality conditions can be written as

$$\begin{aligned} 0 &= \frac{\partial}{\partial \tilde{T}} \mathcal{L}(\tilde{T}, \Lambda) = 2L^\top Z_e L\tilde{T}\Gamma_y V \Gamma_y - L^\top \Lambda C^\top \Gamma_y \\ 0 &= LTC - L\tilde{T}\Gamma_y C \end{aligned}$$

and can be rewritten as

$$\begin{aligned} 0 &= \tilde{T}\Gamma_y - \frac{1}{2}(L^\top Z_e L)^\dagger L^\top \Lambda C^\top V^{-1} \Gamma_y \\ 0 &= LTC(C^\top V^{-1}\Gamma_y C)^\dagger - \frac{1}{2} L (L^\top Z_e L)^\dagger L^\top \Lambda. \end{aligned}$$

Solving the above equations yields the optimal solution (13). ■

Fig. 3 illustrates the centralized reconfiguration that is performed by a system component denoted as reconfiguration manager. A fault occurs at sensor S_2, which detects that it is faulty, reporting it to the reconfiguration manager which now knows Γ_y. The reconfiguration manager solves (13) to derive the new aggregation matrix $\tilde{T} = [\tilde{T}_1 \dots \tilde{T}_p]$, where $\tilde{T}_i$ is a column vector corresponding to the i-th column of $\tilde{T}$. Then, $\tilde{T}_1$ is sent to sensor S_1 and $\tilde{T}_3$ to sensor S_3, which compute $\tilde{T}_1 y_1$ and $\tilde{T}_3 y_3$, where $\tilde{T}_i y_i = [[\tilde{T}_i y_i]_1 \dots [\tilde{T}_i y_i]_s]^\top$. Each non-zero component $[\tilde{T}_i y_i]_j$ is sent to the j-th aggregator, allowing each aggregator node to compute z_j and transmit this value to the estimator.

B. Centralized actuator reconfiguration

The optimal centralized actuator reconfiguration is now presented, which uses the following lemma.

Lemma 2: The optimization problem (12) is equivalent to

$$\begin{aligned} \min_{\tilde{K}} \quad & \operatorname{tr}\left((Q + \tilde{K}^\top \Gamma_u R \Gamma_u \tilde{K}) Z_c\right) \\ \text{s.t.} \quad & BK = B\Gamma_u \tilde{K} \\ & 0 = A_c Z_c + Z_c A_c^\top + R_0. \end{aligned} \tag{15}$$

Following similar steps as in Proposition 1, the optimal centralized actuator reconfiguration is characterized as follows.

Proposition 2: The solution to the actuator reconfiguration problem (12) is

$$\tilde{K}^\star = \Gamma_u R^{-1} B^\top (B \Gamma_u R^{-1} B^\top)^\dagger BK. \tag{16}$$

Fig. 3 depicts also a fault in the actuator network. A fault occurs at actuator A_2, which reports to the reconfiguration manager. The reconfiguration manager then solves (15) to derive the new controller $\tilde{K} = [\tilde{K}_1^\top \ldots \tilde{K}_m^\top]^\top$, where $\tilde{K}_i$ is a row vector corresponding to the i-th row of $\tilde{K}$. Then, $\tilde{K}_1$ is transmitted to to actuator A_1 and $\tilde{K}_3$ to actuator A_3, which allows them to compute and apply $\tilde{u}_1$ and $\tilde{u}_3$, respectively.

We highlight that the centralized actuator reconfiguration solution may be also obtained through other problem formulations. In [11], the authors proposed to solve actuator redundancy through control allocation, which was formulated as an optimization problem using the concept of virtual actuators. By appropriately choosing the objective function, the solution (16) can be obtained. Moreover, the same result may be obtained using the pseudo-inverse method from [28], [29] when R has identical elements.

The centralized reconfiguration scheme requires a centralized entity to compute the optimal T and K matrices and then inform the corresponding sensors and actuators. However, since each sensor/actuator may have a unique encoding/control policy, the dissemination of the optimal matrices requires a point-to-point communication from the centralized entity to each node. This not only represents high computation and communication costs, but it also results in a single point of failure: the centralized entity. Therefore, this centralized approach does not enjoy the usual benefits of distributed solutions: increased scalability, modularity, and failure tolerance. In the next section we propose an optimal distributed solution to the reconfiguration problems (11) and (12).

IV. Distributed sensor and actuator reconfiguration

In this section, we propose a distributed algorithm to solve the reconfiguration problem. We begin by rewriting the equivalent centralized sensor and actuator reconfiguration problems (14) and (15), respectively, as quadratic optimization problems with a separable cost function and a global equality constraint. First, the following notation is introduced. Consider a set of l vectors $\eta_i \in \mathbb{R}^r$ and matrices $H_i \in \mathbb{R}^{n^2 \times r}$, for $i = 1, ..., l$, and define $H = \begin{bmatrix} H_1 & \ldots & H_l \end{bmatrix}$ and $\eta = \begin{bmatrix} \eta_1^\top & \ldots & \eta_l^\top \end{bmatrix}^\top$. Define $\omega \in \mathbb{R}^{n^2}$ and let $S \in \mathbb{R}^{l \times l}$ be a diagonal matrix with non-negative entries.

Lemma 3: The sensor and actuator reconfiguration problems (14) and (15) can be rewritten in the following form:

$$\begin{aligned} \min_{\eta_1, \ldots, \eta_l} \quad & \sum_{i=1}^{l} S_{ii} \|\eta_i\|^2 \\ \text{s.t.} \quad & \sum_{i=1}^{l} H_i \eta_i = \omega. \end{aligned} \tag{17}$$

For the sensor case, we have $l = p$, $\tilde{T} = \begin{bmatrix} \eta_1 & \ldots & \eta_p \end{bmatrix}$, $H = (C^\top \Gamma_y^\top) \otimes L$, $\omega = \operatorname{vec}(LTC)$ and $S_{ii} = [\Gamma_y]_{ii} V_{ii}$.

The actuator case is retrieved with $l = m$, $\tilde{K} = \begin{bmatrix} \eta_1 & \ldots & \eta_m \end{bmatrix}^\top$, $H = (I \otimes B\Gamma_u) P_r^{-1}$ with $P_r \in \mathbb{R}^{mn \times mn}$ being a permutation matrix such that $\operatorname{vec}\left(\tilde{K}\right) = P_r^{-1} \eta$, $\omega = \operatorname{vec}(BK)$ and $S_{ii} = [\Gamma_u]_{ii} R_{ii}$.

Proof: The proof is given in the Appendix. ■

Remark 1: The variables $\eta_i \in \mathbb{R}^r$ and $\omega_i \in \mathbb{R}^{n^2}$ have the following interpretation. For the case of sensor reconfiguration, each η_i represents the aggregation matrix $\tilde{T}$ components for the i-th sensor (i-th column of $\tilde{T}$), i.e., how sensor i transforms its information to be transmitted to each of the fusion nodes that it is connected to. In the same manner, each $\eta_i^\top$ corresponds to the i-th actuator state-feedback matrix $\tilde{K}$ components, i.e., the i-th row of $\tilde{K}$. The value of ω corresponds to the vectorization of the estimation error dynamics and closed-loop system dynamics before a fault occurs, for the case of sensor and actuator reconfiguration, respectively. This represents the quantity that ideally must be maintained by the combination of all sensor (actuator) nodes during the reconfiguration, which refers to the model-matching constraint.

The optimization problem (17) may be solved distributively using dual decomposition and iterative algorithms [30], [31]. A requirement is that the network remains connected when faults occur. Using dual decomposition methods, the optimal solution to problem (17) is guaranteed to be achieved asymptotically in the number of iterations [31]. The main drawback is that the global equality constraint of the problem is only ensured asymptotically. Therefore, model-matching is not guaranteed at every iteration. Due to this fact, we later analyse the stability of the system under the distributed reconfiguration in Section V.

To solve the dual optimization problem of (17) we resort to the distributed alternating direction method of multipliers (ADMM) algorithm [31]. In the following, the decision variable η at each iteration $k \geq 0$ is denoted as $\eta[k]$.

Theorem 1: Consider the equivalent form of the sensor and actuator reconfigurations problems (11) and (12), respectively, presented in Lemma 3. Define $q_1, \ldots, q_l \in \mathbb{R}^{n^2}$ such that $\sum_{i=1}^{l} q_i = \omega$ and the local variables $\zeta_1, \ldots, \zeta_l \in \mathbb{R}^{n^2}$. Let

$$\eta_i[k] = \frac{1}{2} S_{ii}^{-1} H_i^\top \zeta_i[k], \tag{18}$$

where $\zeta_i[k]$ is computed by the following algorithm:

$$\begin{aligned}
\zeta_i[k+1] &= \left(\frac{1}{2}H_i S_{ii}^{-1} H_i^\top + \rho|\mathcal{N}_i| I\right)^{-1} \\
&\quad \times \left(q_i - \rho \sum_{j\in\mathcal{N}_i} \mu_{i,(i,j)}[k] - \pi_{(i,j)}[k]\right) \\
\xi_{i,(i,j)}[k+1] &= \alpha\zeta_i[k+1] + (1-\alpha)\pi_{(i,j)}[k], \\
\pi_{(i,j)}[k+1] &= \frac{1}{2}\left(\xi_{i,(i,j)}[k+1] + \mu_{i,(i,j)}[k]\right. \\
&\quad \left. + \, \xi_{j,(i,j)}[k+1] + \mu_{j,(i,j)}[k]\right), \\
\mu_{i,(i,j)}[k+1] &= \mu_{i,(i,j)}[k] + \xi_{i,(i,j)}[k+1] - \pi_{(i,j)}[k+1],
\end{aligned} \tag{19}$$

where $\rho > 0$ is the step size, $\alpha \in (0,2)$ is a relaxation parameter, $\rho\mu_{i,(i,j)}$ is the Lagrange multiplier of node i associated with the constraint $\zeta_i = \pi_{(i,j)}$, and $\xi_{i,(i,j)}(k)$ is an auxiliary variable private to node i associated with the edge (i,j). Then, $\eta[k]$ converges to the solution of (17), $\eta^\star$, from which the solution to the sensor and actuator reconfigurations problems, (11) and (12), can be retrieved as $\tilde{T}^\star = \begin{bmatrix}\eta_1 & \dots & \eta_p\end{bmatrix}$ and $\tilde{K}^\star = \begin{bmatrix}\eta_1 & \dots & \eta_m\end{bmatrix}^\top$, respectively.

Note that the ADMM algorithm in Theorem 1 is distributed, since it only requires communication between neighbors to exchange local variables. Methods to choose the parameters ρ and α to increase the convergence speed are given in [32].

To prove Theorem 1, we first derive the dual form of (17).

Lemma 4: Let $f_i(\eta_i) = S_{ii}\eta_i^\top\eta_i$. The optimization problem (17) can be rewritten in the following dual form:

$$\begin{aligned}
\min_{\{\zeta_i\},\{\pi_{(i,j)}\}} \quad & \sum_{i=1}^{l}\left(\frac{1}{4}S_{ii}^{-1}\zeta_i^\top H_i H_i^\top \zeta_i - q_i^\top\zeta_i\right) \\
\text{s.t.} \quad & \zeta_i = \pi_{(i,j)}, \quad \forall i \in \mathcal{V},\ j \in \mathcal{N}_i.
\end{aligned} \tag{20}$$

Proof: [Proof of Theorem 1] The value of $\eta[k]$ is obtained as $\eta[k] = \text{argmin}_{\varphi_i} f_i(\varphi_i) - \zeta^T H_i \varphi_i = \frac{1}{2}S_{ii}^{-1}H_i^\top\zeta_i[k]$. The ADMM algorithm (19) follows from [31] and is thus omitted. ■

Remark 2: The variables $q_i \in \mathbb{R}^{n^2}$ and $\zeta_i \in \mathbb{R}^{n^2}$ have the following interpretation. The vector q_i describes how the vectorization of the closed-loop dynamics, i.e., ω, is assigned among all nodes in the network. Note that the assignment is only constrained by the condition $\sum_{i=1}^{l} q_i = \omega$, thus admitting several solutions. The variable ζ_i, only available at node i, is a local copy of the Lagrange multiplier associated with the model-matching constraint $H\eta = \omega$.

The following result indicates how the parameters q_i can be updated locally by the healthy nodes after a fault has occurred.

Lemma 5: Let $j \in \mathcal{V}^f$ be an arbitrary faulty node, denote $\mathcal{J} \subseteq \mathcal{N}_j \cap \mathcal{V}_h$ as a subset of its healthy neighbors and assume $\mathcal{J}$ is not empty. Given the set $\{\bar{q}_i\}_{i\in\mathcal{V}}$ such that $\sum_{i\in\mathcal{V}}\bar{q}_i = \omega$, the set $\{q_i\}_{i\in\mathcal{V}}$ satisfying $\sum_{i\in\mathcal{V}_h} q_i = \omega$ can be computed as

$$q_i = \begin{cases} \bar{q}_i, & i \notin \mathcal{J} \\ \bar{q}_i + \nu_i\bar{q}_j, & i \in \mathcal{J} \end{cases}, \tag{21}$$

where $\nu_i \geq 0$ for all $i \in \mathcal{J}$ and $\sum_{i\in\mathcal{J}}\nu_i = 1$.

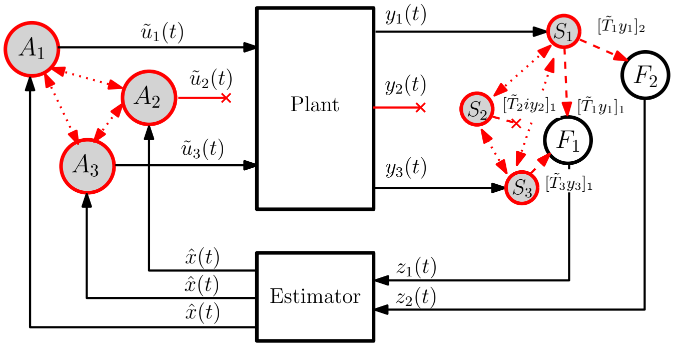


Fig. 4: Networked control system with distributed sensor and actuator reconfiguration. Faults are detected by the sensors and actuators which are responsible for the reconfiguration. Reconfiguration is achieved through the communication among sensors and among actuators in a distributed manner through the sensor and actuator network, respectively.

Algorithm 1 Distributed sensor/actuator reconfiguration

```
1) Detect and isolate sensor/actuator faults and disconnect
   the faulty nodes at $t = 0$;
2) Locally compute $q_i$ as per Lemma 5;
3) Compute the optimal solution $\zeta_i^\star$ to the dual problem (20) using the algorithm in Theorem 1;
4) Compute the primal optimal solution $\eta_i^\star = \frac{1}{2}S_{ii}^{-1}H_i^\top\zeta_i(k)$;
5) Each sensor/actuator node $i$ applies $\eta_i^\star$.
```

The distributed reconfiguration algorithm can be summarized in Algorithm 1. An illustration of the distributed sensor and actuator reconfiguration is shown in Fig. 4 where a fault occurs at sensor S_3 and actuator A_2 at $t = t_0$. The sensors locally infer that sensor S_2 is no longer functioning, so sensors S_1 and S_3 reconfigure themselves. This is performed locally by each sensor computing the value of $\tilde{T}_1$ and $\tilde{T}_3$, and calculating $\tilde{T}_1 y_1$ and $\tilde{T}_3 y_3$. Each component $[\tilde{T}_i y_i]_j$ is sent to the j-th aggregator, allowing each aggregator node to compute z_j and transmit this value to the controller node. Similarly, the actuators locally infer that actuator A_2 is faulty, so actuators A_1 and A_3 reconfigure themselves. This is a local operation where each actuator computes the value of $\tilde{K}_1$ and $\tilde{K}_3$.

V. Closed-loop stability under distributed reconfiguration

The proposed distributed algorithm converges to the optimum asymptotically as it solves the dual problem. Primal feasibility (model-matching), i.e., $H\eta[k] = \omega$, is only achieved in the limit as the number of iterations k grows to infinity. Therefore, one relevant concern is the system's stability when the dual algorithm is terminated in a finite number of iterations. The results of this section are two-fold. First, assuming that, on average, faults occur sufficiently far apart in time, we provide results that guarantee global exponential stability if the gain matrix produced by the reconfiguration algorithm in finite time yields a Hurwitz closed-loop system matrix with a known decay rate. Second, we derive an upper bound on the number of iterations which ensures that a Hurwitz closed-loop system matrix with a prescribed decay rate is obtained when the dual algorithm is terminated.

A. Stability analysis

Consider the system model (1) and (2) without noise, together with the control law (5) and the estimator (6), which may be affected by faults occurring at different times. Next we describe the behavior of the reconfiguration scheme, under only actuator faults, for simplicity, and analyze its stability. Similar results can be derived for the general case of both sensor and actuators faults.

Under the proposed reconfiguration scheme, after each fault i occurs, the distributed reconfiguration algorithm in Theorem 1 is run for a finite number of iterations $\bar{k}$ and a suboptimal gain matrix is applied, yielding the closed-loop system $A_c + \Delta^{(i)}[\bar{k}]$ with $\Delta^{(i)}[\bar{k}]$ defined by vec $\left(\Delta^{(i)}[\bar{k}]\right) = H\eta[\bar{k}] - \omega$. The algorithm may continue to run if no new fault occurs and, when the optimal gain matrix is obtained, it is applied to the system to recover the nominal dynamics, A_c.

Denoting $t_f^{(i)}$ as the time instant at which the i-th fault occurs, the system dynamics under multiple faults and the proposed reconfiguration scheme can be expressed by the switched system

$$\begin{aligned} \dot{x}(t) &= (A - B\Gamma_u^{(i)}K)x(t), && \text{for } t \in [t_f^{(i)},\, t_r^{(i)}) \\ \dot{x}(t) &= (A_c + \Delta^{(i)}[\bar{k}])x(t), && \text{for } t \in [t_r^{(i)},\, t_n^{(i)}) \\ \dot{x}(t) &= A_c x(t), && \text{for } t \in [t_n^{(i)},\, t_f^{(i+1)}), \end{aligned} \tag{22}$$

with initial condition $x(t_0) = x_0$, where $t_r^{(i)}$ and $t_n^{(i)}$ are the time instants at which the finite-time and optimal gain matrices are applied, with $t_0 \leq t_f^{(i)} \leq t_r^{(i)} \leq t_n^{(i)}$ and $t_n^{(i)} \leq t_f^{(i+1)}$ for all i. Note that the non-strict inequalities allow for new faults to occur at different stages of the reconfiguration.

Recall that (22) is globally exponentially stable if there exist positive scalar c and λ such that $\|x(t)\| \leq ce^{-\lambda(t-t_0)}\|x_0\|$. Next we provide sufficient conditions establishing the global stability of the switched system (22) when the faulty system is unstable and $A_c + \Delta^{(i)}[\bar{k}]$ is Hurwitz.

We make the following definitions and assumptions on the occurrence of faults. Let $N_f(t_0, t)$ be the number of faults occurring within $(t_0,\, t)$, τ_f the average dwell time between faults, and N_0 the chatter bound.

Assumption 2: The occurrence of faults is such that the following inequality holds: $N_f(t_0, t) \leq N_0 + \dfrac{t - t_0}{\tau_f}$.

Assumption 3: There exist $a \leq 0$ and positive scalars λ_f, λ_r, and λ_n such that

$$\begin{gathered} \|e^{(A-B\Gamma_u^{(i)}K)t}\| \leq e^{a+\lambda_f t}, \text{ for all } i \\ \|e^{(A_c+\Delta^{(i)}[\bar{k}])t}\| \leq e^{a-\lambda_r t}, \text{ for all } i \\ \|e^{A_c t}\| \leq e^{a-\lambda_n t}. \end{gathered}$$

Furthermore, we assume that $\lambda_n \geq \lambda_r$ holds, which captures the fact that the nominal system decays faster than the system reconfigured with a gain computed in finite time.

Remark 3: Given the system matrices of (22) for each i, [33] describes methods to determine the scalars a, λ_f, λ_r, and λ_n satisfying Assumption 3. While these methods can be executed to compute λ_f and λ_n, by enumerating all possible $\Gamma_u^{(i)}$, they cannot be used to determine λ_r since $\Delta^{(i)}[\bar{k}]$ is unknown. In Section V-B, we provide a way to determine a and λ_r satisfying the second inequality in Assumption 3 for any matrix $\Delta^{(i)}[\bar{k}]$ satisfying $\|\Delta^{(i)}[\bar{k}]\|_F \leq \delta$.

Let $\tau_c(\bar{k})$ be the time required for completing $\bar{k}$ iterations of the reconfiguration algorithm.

Assumption 4: There exists some $\lambda^\star \in (0, \lambda_r)$ such that the following inequality holds:

$$\tau_f > \max\left\{\frac{\lambda_f + \lambda_r}{\lambda_r - \lambda^\star}\tau_c(\bar{k}),\ \frac{a}{\lambda^\star}\right\}.$$

Theorem 2: Consider the system dynamics under multiple faults and the proposed reconfiguration scheme described in (22), with A_c and $A_c + \Delta^{(i)}[\bar{k}]$ being Hurwitz for all i. The switched system (22) is globally exponentially stable if the occurrence of faults is such that Assumption 2 holds with an arbitrary $N_0 > 0$ and with an average dwell time between faults $\tau_f > \tau_c(\bar{k}) + \dfrac{a + \lambda_f \tau_c(\bar{k})}{\lambda_r}$.

Proof: The proof may be found in the appendix. ■

Theorem 2 guarantees global exponential stability for a sufficiently large average dwell time between faults, even if the faulty systems have unstable dynamics. Apart from the requirements on the dwell time, the main required conditions are that $A_c + \Delta[\bar{k}]$ is Hurwitz and that one knows its decay rate λ_r. Next we tackle these aspects by providing criteria to terminate the reconfiguration algorithm in finite-time while ensuring that the computed gain matrix yields a Hurwitz closed-loop matrix with a prescribed decay rate.

B. Criteria for finite-time termination

Note that the closed-loop system dynamics and the estimation error dynamics may each be described by a generic system $\dot{v} = (D + \Delta)v$ with D stable and uncertainty Δ, where vec $(\Delta) = H\eta[k] - \omega$. For the sensor reconfiguration analysis, we have $v = \hat{x}$, $D = A_e$, $H = (C^\top\Gamma_y^\top) \otimes L$ and $\omega =$ vec (LTC). Similarly, in the actuator reconfiguration case $v = x$, $D = A_c$, $H = (I \otimes B\Gamma_u)\, P_r^{-1}$ and $\omega =$ vec (BK).

First, based on [34], we recall a sufficient condition for robust stability with bounded uncertainties that further ensures a given decay rate, thus complying with Assumption 3.

Lemma 6: Given a Hurwitz matrix D and $\lambda_r > 0$, if there exists a positive definite matrix X such that

$$D^\top X + XD + XX + \delta^2 I + 2\lambda_r X \prec 0,$$

then, for any norm-bounded uncertainty $\|\Delta\|_F \leq \delta$ with $\delta > 0$, the matrix $D + \Delta$ is Hurwitz and there exists a scalar $a > 0$ such that $\|e^{(D+\Delta)t}\| \leq e^{a-\lambda_r t}$.

Theorem 3: Consider the sequence of vectors $\{\eta[k]\}$ converging to $\eta^\star \in \mathcal{H} = \{\eta :\ H\eta = \omega\}$ and define $\Delta[k]$ such that vec $\left(\Delta[k]\right) = H\eta[k] - \omega$.

Suppose there exist matrices $X \succ 0$ and $M \succ 0$ satisfying the matrix equation $D^\top X + XD + X^2 + 2\lambda_r X + M = 0$

and a positive decreasing function of k, $\epsilon[k] > 0$, such that $\|\Delta[k]\|_F \le \epsilon[k]\|\Delta[0]\|_F$ holds for all k.

Define the integer $\bar{k}$ such that the following inequality holds:

$$\epsilon[\bar{k}] < \frac{\sqrt{\lambda_{\min}(M)}}{\|H\eta[0] - \omega\|}. \tag{23}$$

Then, the system matrix $D + \Delta[k]$ is Hurwitz with decay rate $\lambda_r \ge 0$ if the termination iteration k satisfies $k \ge \bar{k}$.

Proof: Suppose that $\|\Delta[k]\|_F \le \epsilon[k]\|\Delta[0]\|_F$ and let $\delta[k] = \|\Delta[k]\|_F$. From Lemma 6, the closed-loop system matrix at iteration k is guaranteed to be Hurwitz with decay rate $\lambda_r \ge 0$ if $D^\top X + XD + X^2 + 2\lambda_r X + \delta[k]^2 I = -M + \delta[k]^2 I \prec 0$, which is ensured for $\bar{k}$ when $\epsilon[\bar{k}]\delta[0] < \sqrt{\lambda_{\min}(M)}$. Recalling that $\epsilon[k]$ is decreasing concludes the proof. ■

The above result provides a method to terminate the dual algorithm while ensuring stability. It only requires knowledge of the convergence properties of the dual algorithm, namely the function $\psi[k]$, and the initial distance $\|\Delta[0]\|_F$. The latter can be computed when the reconfiguration algorithm is initialized, since it only depends on the nominal controller and the initial condition of the algorithm, $\eta[0]$, which is determined by the identification of the faulty nodes.

Convergence properties of distributed algorithms, and characterization of their respective functions $\psi[k]$, are readily available in the literature, see [31], [32], [35]. Next we combine the results of Theorem 3 with the distributed ADMM algorithm described in Theorem 1, and the respective convergence properties analyzed in [32], to derive an explicit lower bound on $\bar{k}$ that ensures robust stability with a given decay rate.

Lemma 7: Consider the optimization problem (17), its equivalent dual formulation (20), and the ADMM algorithm described in Theorem 1. Let $\zeta^\star = \lim_{k\to\infty} \zeta[k]$ be the optimal solution to (20). Then, we have $\|\zeta[k] - \zeta^\star\| \le \psi\|\zeta[k-1] - \zeta^\star\|$ for all k with $\psi \in [0\ 1)$.

Proof: The proof follows directly from [32, Theorem 1], where the decay rate ψ can be found. ■

Theorem 4: Consider the optimization problem (17), its equivalent dual formulation (20), and the ADMM algorithm described in Theorem 1. The closed-loop system matrix obtained at the iteration k from $\eta[k]$ is guaranteed to be Hurwitz with decay rate $\lambda_r \ge 0$ for all $k \ge \bar{k}$ with

$$\bar{k} = \left\lceil \frac{\log(\sqrt{\lambda_{\min}(M)}) - \log\left(\|H\eta[0] - \omega\|\kappa(HS^{-1}H^\top)\right)}{\log(\psi)} \right\rceil.$$

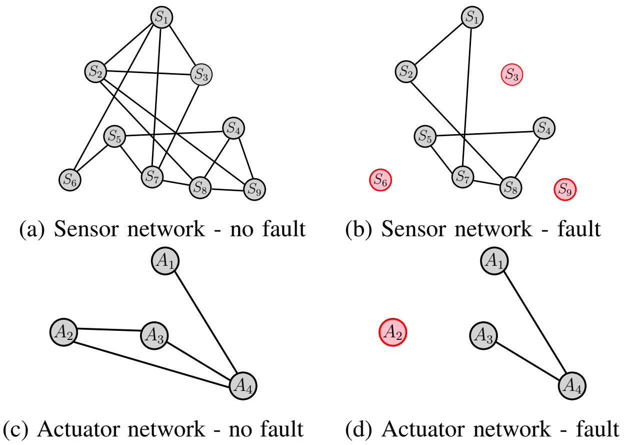
Proof: Since $H\eta[k] = -1/2HS^{-1}H^\top\zeta[k]$ for all k, we can derive the following bound $\|H\eta[k] - H\eta^\star\| \le \|1/2HS^{-1}H^\top\|_2\|(\zeta[k] - \zeta^\star)\|$. Using Lemma 7, we have

$$\|H\eta[k] - H\eta^\star\|_2 \le \kappa(HS^{-1}H^\top)\psi^k\|H\eta[0] - H\eta^\star\|.$$

Recalling that $\|\Delta[0]\|_F = \|H\eta[0] - \omega\| = \|H\eta[0] - H\eta^\star\|$ and applying Theorem 3, we observe that the closed-loop matrix satisfies the desired properties for all k such that

$$\psi^k < \frac{\sqrt{\lambda_{\min}(M)}}{\|H\eta[0] - H\eta^\star\|\kappa(HS^{-1}H^\top)}.$$

The proof concludes by taking the logarithm of both sides and rearranging the terms. ■



(a) Sensor network - no fault

(b) Sensor network - fault

(c) Actuator network - no fault

(d) Actuator network - fault

Fig. 5: Sensor and actuator network graph. The healthy nodes are colored black and the faulty nodes are colored red.

Next we compute the matrices X and M that maximize the magnitude of the uncertainty for which it is ensured that $D + \Delta$ is Hurwitz with decay rate $\lambda_r \ge 0$.

Proposition 3: Denote $X^\star$ and $\sigma^\star$ as the optimal solution to the convex optimization problem

$$\begin{aligned} \max_{X\succ 0,\, \sigma>0} \quad & \sigma \\ \text{s.t.} \quad & 0 \succ D^\top X + XD + \sigma I + 2\lambda_r X \\ & 0 \prec \begin{bmatrix} -D^\top X - XD - \sigma I - 2\lambda_r X & X \\ X & I \end{bmatrix}. \end{aligned} \tag{24}$$

Then, matrix $X^\star$ satisfies the robust stability constraint $D^\top X + XD + X^2 + \delta^2 I + 2\lambda_r X \prec 0$ with $\delta^2 = \sigma^\star$ being the largest disturbance magnitude for which it is ensured that $D + \Delta$ is Hurwitz with decay rate $\lambda_r \ge 0$. Additionally, we have that the optimal matrix M is given by $M^\star = -D^\top X^\star - X^\star D - 2\lambda_r X^\star - X^{\star 2} \succ 0$.

Proof: The proof follows from Lemma 6. ■

The value $\bar{k}$ assures that stability can be achieved in a finite iterations. We remark that the lower bound $\bar{k}$ obtained from Theorem 4 is expected to be conservative, which will be illustrated in the numerical example.

The calculation of $\bar{k}$ as per Theorem 4 can be efficiently performed in a centralized manner, by using the knowledge of which nodes are faulty to compute H and $\eta[0]$, which could then be broadcast to all nodes. A more conservative value of $\bar{k}$ can be obtained in a distributed manner, by setting $\eta[0] = 0$ and using an upper bound of $\kappa(HS^{-1}H^\top)$ and ψ.

VI. Numerical Example

This section provides a numerical example that illustrates the proposed distributed reconfiguration method. For an experimental evaluation of the proposed methods in a room heating scenario with a network of actuators, please see [36, Ch. 7].

A. Networked control system setup

In the following example, the aim is to control an unstable second-order system with 9 sensors and 4 actuators. The system dynamics, measured outputs and aggregated outputs are given by (1), (2) and (3), respectively, where

$$A = \begin{bmatrix} 9 & 2.5 \\ 4 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 2.83 & 4.01 & 0.21 & -0.58 \\ -0.16 & -0.64 & 2.86 & 4.73 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0.1 \\ -2 & -0.2 \\ 4 & 0.4 \\ 0.1 & 1 \\ -0.5 & -5 \\ 0.3 & 3 \\ 1 & 1 \\ 1 & 1 \\ 0.5 & 0.5 \end{bmatrix}, \quad T = \begin{bmatrix} 0.36 & 0.26 & 0 \\ 0.04 & 0.17 & 0 \\ 0.24 & 0 & 0.52 \\ 0 & 0.88 & 0.73 \\ 0.24 & 0 & 0.86 \\ 0 & 0.62 & 0.60 \\ 0 & 0.60 & 0.14 \\ 0 & 0.64 & 0.63 \\ 0.64 & 0 & 0.18 \end{bmatrix}^\top.$$

To enable reference tracking, the plant is augmented with two integral states, representing the integral error at each physical state. The control cost parameters are

$$R = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 150 & 0 \\ 0 & 0 & 0 & 200 \end{bmatrix}, \; Q = 100I,$$

while the noise covariances are $V = 0.4I$ and $W = I$. Moreover, the state estimate and control input are given by (6) and (5), respectively. The initial gains L and K are the solutions to the LQG controller design problem. The ADMM parameters in (19) are set to $\rho = 1$ and $\alpha = 1.5$.

The sensor network graph is given in Figs. 5a and 5b while the actuator network is depicted in Figs. 5c and 5d, for the nominal and faulty cases, respectively.

B. Convergence of the distributed reconfiguration algorithm

We start by analyzing the performance of the distributed reconfiguration scheme presented in Section IV for the sensor and actuator faults depicted in Fig. 5. As performance indicators, we consider the normalized objective function errors $|J_e[k] - J_e^\star|$ and $|J_c[k] - J_c^\star|$, the errors in the model-matching constraint $\|H^e\eta[k] - w^e\|$ and $\|H^c\eta[k] - w^c\|$ and the maximum real part of the eigenvalues of $A_e[k] = A - L\bar{T}[k]\Gamma_y C$ and $A_c[k] = A - B\Gamma_u\bar{K}[k]$ that relates to the stability of the intermediate reconfiguration solutions.

The results are depicted in Fig. 6. As it can be seen, the distributed method asymptotically achieves the optimal cost and guarantees the model-matching constraint. Moreover, the state estimation error dynamics is unstable for the first 2 steps, i.e., $\lambda_r[k] = \max_i \left\{\Re\{\lambda_i(A_e[k])\}\right\} > 0, \; k = 1, 2$, while the closed-loop dynamics are unstable for only the first step since $\lambda_r[k] = \max_i \left\{\Re\{\lambda_i(A_c[k])\}\right\} > 0, \; k = 1$. Applying Theorem 4 from Section V, with $\underline{\lambda_r} = 0$, we obtain the guarantee that $A_e[k]$ is stable for $k \geq \bar{k} = 53$ steps and $A_c[k]$ is stable for $k \geq \bar{k} = 8$ steps. Since Lemma 6 provides a conservative stability guarantee, the obtained $\bar{k}$ is expected to be conservative. The distributed sensor reconfiguration takes 15 steps to converge to $|J_e[k] - J_e^\star| < 10^{-3}$ and $\|H^e\eta[k] - w^e\| < 10^{-1}$. Similarly, the distributed actuator reconfiguration takes approximately 16 steps to converge.

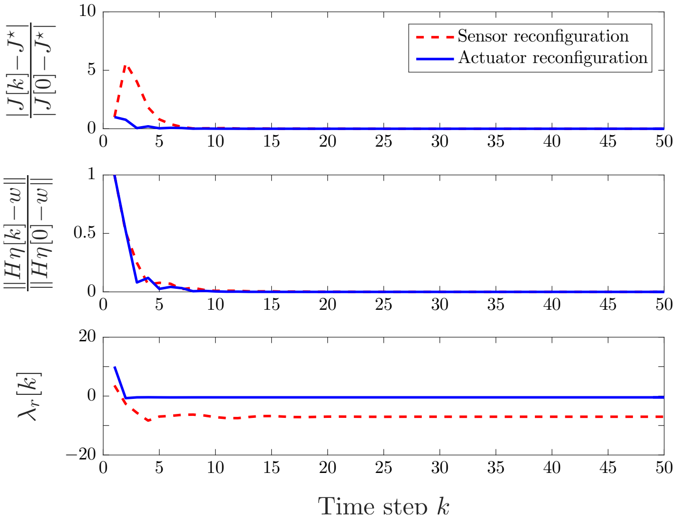


Fig. 6: Performance of the distributed sensor and actuator reconfiguration method for the networks depicted in Fig. 5, with $\lambda_r[k] \triangleq \max_i \left\{\Re\{\lambda_i(A_{(\cdot)}[k])\}\right\}$.

C. Simulation results

The time-responses of the closed-loop system under the faults in Fig. 5 are depicted in Fig. 7, which include the state trajectories $x(t)$, the control inputs $u(t)$, and the running control cost $J_c(t)$ defined in (4). In Fig. 7 we depict three cases: 1) no faults occur (solid line); 2) faults occur and detection and isolation are instantaneous, but reconfiguration is performed in real-time and intermediate reconfiguration solutions are utilized at each time-step (dash-dotted line); 3) faults occur, but no reconfiguration is performed (dashed line).

The second case aims at demonstrating the impact of applying the reconfigured output before the reconfiguration algorithm has converged to a stable closed-loop system. Therefore, to better observe the impact of a slow real-time reconfiguration in the system dynamics, the following two settings are considered. First, the control law under reconfiguration is set to zero immediately after the fault, which results in an unstable open-loop system. Second, each iteration of the reconfiguration is set to take 6 s to run, which includes both computation and communication time. However, in practice, much smaller computation and communication times can be obtained, while the control policy under reconfiguration may be, for instance, initialized at the pre-fault policy, thus improving the performance of a real-time reconfiguration.

The sensor faults occur at time $t = 20$ s and the actuator faults at $t = 100$ s. Observe that sensor faults have a small influence in all of the cases, as verified in the plots of $x(t)$ and $J_c(t)$. However, as it can be seen around $t = 40$ s, the state trajectory $x(t)$ when no reconfiguration is performed has a large deviation from the nominal trajectory, which does not occur when the proposed reconfiguration scheme is applied.

Fig.7 shows that the actuator fault has a more severe impact in the second and third cases. In the second case, when real-time reconfiguration is performed, we observe that the state trajectory $x(t)$ immediately deviates from the nominal trajectory. This deviation is mainly due to the initialization of

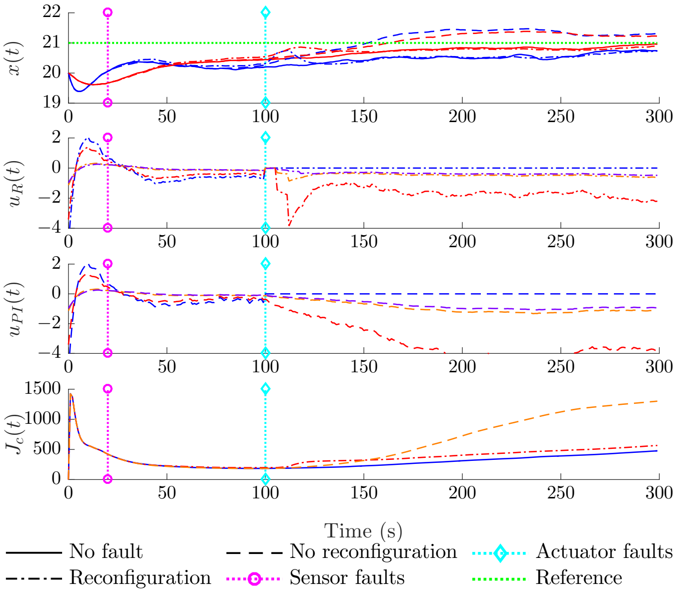


Fig. 7: Time-response of the state and estimation error trajectories and control input for the scenarios in Fig. 5. The reference value to be tracked is depicted by the dotted line. Sensor faults occur at time $t = 20\,\mathrm{s}$ and actuator faults at $t = 100\,\mathrm{s}$. Three cases are compared: no faults (solid), real-time reconfiguration (dash-dotted) and no reconfiguration (dashed). The control signals for the fault scenarios with and without reconfiguration are denoted as $u_R(t)$ and $u_{PI}(t)$, respectively.

the reconfiguration algorithm, where the control law of each actuator is initially set to zero, see the control signal plot $u_R(t)$ for $t \in [100, 106]$ s. However, as seen in the plot of $u_R(t)$, the reconfiguration scheme reaches a stabilizing control law after $\tau = 12\,\mathrm{s}$ (i.e., when two iterations are completed, c.f. Fig. 6) and $x(t)$ begins converging to the nominal trajectory.

On the other hand, the third case with no reconfiguration has a better transient behavior, but worse performance in the long-term. In fact, as seen in the plot of $J_c(t)$ for $t \in [110, 150]$ s, the system without reconfiguration has a lower running cost that the reconfigured system. However, as time runs on, the trajectories without reconfiguration substantially deviate from the nominal trajectories (i.e., trajectories of the system without faults), see the plots for $x(t)$ and $u_{PI}(t)$ from $t = 140\,\mathrm{s}$ onwards. This is further corroborated by the behavior of the cost $J_c(t)$ for $t \in [160, 300]$ s.

VII. Conclusions

In this work, we developed a distributed reconfiguration method for networked control systems under sensor and actuator faults. The proposed approach guarantees a model-matching reconfiguration while minimizing the steady-state estimation error covariance and a quadratic control cost. The distributed reconfiguration method is guaranteed to achieve the same solution as the centralized reconfiguration, while only requiring local cooperation among healthy sensors and actuators. A numerical example demonstrates the effectiveness of our approach.

References

[1] S. Ding, *Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Springer-Verlag Berlin Heidelberg, 2008.

[2] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, 2nd ed., ser. Engineering online library. Springer-Verlag, 2006.

[3] J. Bendtsen, K. Trangbaek, and J. Stoustrup, "Plug-and-play control - modifying control systems online," *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 1, pp. 79–93, 2013.

[4] S. Riverso, M. Farina, and G. Ferrari-Trecate, "Plug-and-play decentralized model predictive control for linear systems," *Automatic Control, IEEE Transactions on*, vol. 58, no. 10, pp. 2608–2614, Oct 2013.

[5] S. Bodenburg and J. Lunze, "Plug-and-play control - theory and implementation," in *Industrial Informatics (INDIN), 2013 11th IEEE International Conference on*, July 2013, pp. 165–170.

[6] R. Poovendran, K. Sampigethaya, S. K. S. Gupta, I. Lee, K. V. Prasad, D. Corman, and J. Paunicka, "Special issue on cyber - physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 6 –12, jan. 2012.

[7] J. Maciejowski, "Reconfigurable control using constrained optimization," in *Proceeding of European Control Conference, Brussels, Belgium*. Citeseer, 1997, pp. 107–130.

[8] J. Lunze and J. H. Richter, "Reconfigurable fault-tolerant control: a tutorial introduction," *European Journal of Control*, vol. 14, no. 5, pp. 359–386, 2008.

[9] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual Reviews in Control*, vol. 32, no. 2, pp. 229 – 252, 2008.

[10] A. Saberi, A. A. Stoorvogel, and P. Sannuti, *Filtering Theory - With Applications to Fault Detection, Isolation, and Estimation*, ser. Systems & Control: Foundations & Applications. Birkhäuser, 2007.

[11] O. Härkegård and S. T. Glad, "Resolving actuator redundancy: optimal control vs. control allocation," *Automatica*, vol. 41, no. 1, pp. 137 – 144, 2005.

[12] I. Hwang, S. Kim, Y. Kim, and C. E. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 636–653, May 2010.

[13] N. Wu, K. Zhou, and G. Salomon, "Control reconfigurability of linear time-invariant systems," *Automatica*, vol. 36, no. 11, pp. 1767 – 1771, 2000.

[14] M. Staroswiecki, H. Yang, and B. Jiang, "Progressive accommodation of parametric faults in linear quadratic control," *Automatica*, vol. 43, no. 12, pp. 2070 – 2076, 2007.

[15] M. Staroswiecki and F. Cazaurang, "Fault recovery by nominal trajectory tracking," in *American Control Conference, 2008*, 2008, pp. 1070–1075.

[16] M. Staroswiecki and D. Berdjag, "A general fault tolerant linear quadratic control strategy under actuator outages," *International Journal of Systems Science*, vol. 41, no. 8, pp. 971–985, 2010.

[17] J. H. Richter, W. P. M. H. Heemels, N. van de Wouw, and J. Lunze, "Reconfigurable control of piecewise affine systems with actuator and sensor faults: Stability and tracking," *Automatica*, vol. 47, no. 4, pp. 678–691, Apr. 2011.

[18] J. Åkerberg, M. Gidlund, and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *9th IEEE International Conference on Industrial Informatics (INDIN)*, July 2011.

[19] J. Campelo, F. Rodriguez, A. Rubio, R. Ors, P. Gil, L. Lemus, J. Busquets, J. Albaladejo, and J. Serrano, "Distributed industrial control systems: a fault-tolerant architecture," *Microprocessors and Microsystems*, vol. 23, no. 2, pp. 103 – 112, 1999.

[20] S. Jiang, P. Voulgaris, and N. Neogi, "Failure-robust distributed controller architectures," *International Journal of Control*, vol. 80, no. 9, pp. 1367–1378, 2007.

[21] X. Z. Jin and G. H. Yang, "Distributed fault-tolerant control systems design against actuator faults and faulty interconnection links: An adaptive method," in *American Control Conference, 2009. ACC '09.*, 2009, pp. 2910–2915.

[22] I. Yang, D. Kim, and D. Lee, "Fault-tolerant control strategy based on control allocation using smart actuators," in *Control and Fault-Tolerant Systems (SysTol), 2010 Conference on*, 2010, pp. 377–381.

[23] K. J. Åström, *Introduction to Stochastic Control Theory*. Academic Press, 1970, republished by Dover Publications, 2006.

[24] X. Zhang, M. Polycarpou, and T. Parisini, "A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 576–593, Apr. 2002.

[25] K. Paridari, A. E. D. Mady, S. L. Porta, R. Chabukswar, J. Blanco, A. Teixeira, H. Sandberg, and M. Boubekeur, "Cyber-physical-security framework for building energy management system," in *Proc. ACM/IEEE 7th Int. Conf. on Cyber-Physical Systems (ICCPS)*, Vienna, Austria, 2016, pp. 1–9.

[26] A. Teixeira, K. Paridari, H. Sandberg, and K. H. Johansson, "Voltage control for interconnected microgrids under adversarial actions," in *Proc. 20th IEEE International Conf. on Emerging Technologies and Factory Automation (ETFA)*, Luxembourg, 2015.

[27] K. Morrisse, G. Solimini, and U. Khan, "Distributed control schemes for wind-farm power regulation," in *North American Power Symposium (NAPS), 2012*, 2012, pp. 1–6.

[28] Z. Gao and P. J. Antsaklis, "Stability of the pseudo-inverse method for reconfigurable control systems," *International Journal of Control*, vol. 53, no. 3, pp. 717–729, 1991.

[29] M. Staroswiecki, "Fault tolerant control : The pseudo-inverse method revisited," in *16th Triennial World Congress*, 2005.

[30] H. Everett III, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations research*, vol. 11, no. 3, pp. 399–417, 1963.

[31] 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*, vol. 3, no. 1, pp. 1–122, 2011.

[32] E. Ghadimi, A. Teixeira, M. Rabbat, and M. Johansson, "The ADMM algorithm for distributed averaging: Convergence rates and optimal parameter selection," in *Proceedings of the 48th Asilomar Conference on Signals, Systems and Computers*, 2014.

[33] G. Zhai, B. Hu, K. Yasuda, and A. N. Michel, "Stability analysis of switched systems with stable and unstable subsystems: An average dwell time approach," *International Journal of Systems Science*, vol. 32, no. 8, pp. 1055–1061, 2001.

[34] J. H. Lee, W. H. Kwon, and J.-W. Lee, "Quadratic stability and stabilization of linear systems with Frobenius norm-bounded uncertainties," *Automatic Control, IEEE Transactions on*, vol. 41, no. 3, pp. 453–456, Mar. 1996.

[35] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *Journal of Optimization Theory and Applications*, vol. 129, pp. 469–488, 2006.

[36] J. Araújo, "Design, implementation and validation of resource-aware and resilient wireless networked control systems," Ph.D. dissertation, KTH Royal Institute of Technology, 2014.

Appendix

A. Proof of Lemma 1

The first constraint in (14) is the model-matching constraint which is derived as follows. Following Sec. II-B, model-matching is guaranteed if the closed-loop matrix before fault is the same as after the fault, i.e., $A - L\tilde{T}\Gamma_y C = A_e$. Moreover, the objective function and last constraint follow are given as follows. The objective function J_e in (9) is equivalent to $J_e = \mathrm{tr}(\tilde{\Sigma})$, where $\tilde{\Sigma}$ is steady-state covariance of the estimation error after a fault and defined as $\tilde{\Sigma} = \lim_{t\to\infty} \mathbf{E}\left\{\tilde{e}(t)\tilde{e}(t)^\top\right\}$. Additionally, under any given estimator gain L, $\tilde{\Sigma}$ is given by the following Lyapunov equation (see [23] for details),

$$A_e\tilde{\Sigma} + \tilde{\Sigma}A_e^\top + W + L\tilde{T}\Gamma_y V\Gamma_y{}^\top\tilde{T}^\top L^\top = 0.$$

The solution of the above Lyapunov equation, can also be expressed as $\tilde{\Sigma} = \int_0^\infty e^{A_e t}\left(W + L\tilde{T}\Gamma_y V\Gamma_y\tilde{T}^\top L^\top\right)e^{A_e^\top t}dt$. Noticing that the term $W + L\tilde{T}\Gamma_y V\Gamma_y\tilde{T}^\top L^\top$ is independent of time, one can arrive to the following equivalence of the cost $J_e = \mathrm{tr}(\tilde{\Sigma}) = \mathrm{tr}\left(\left(W + L\tilde{T}\Gamma_y V\Gamma_y\tilde{T}^\top L^\top\right)\int_0^\infty e^{A_e^\top t}e^{A_e t}dt\right)$. The proof concludes by noticing that $Z_e = \int_0^\infty e^{A_e^\top t}e^{A_e t}dt$ is the solution to the Lyapunov equation $A_e^\top Z_e + Z_e A_e + I = 0$.

B. Proof of Lemma 3

In order to prove Lemma 3, we rewrite the sensor and actuator reconfiguration problems (14) and (15) as quadratic optimization problems with equality constraints. Next we derive the proof for the sensor reconfiguration, while the actuator case is omitted for brevity.

Lemma 8: Define $\tilde{T} = \begin{bmatrix}\eta_1 \cdots \eta_p\end{bmatrix}$, $\eta_i \in \mathbb{R}^s$ and let $H_i \in \mathbb{R}^{n^2\times s}$ for $i = 1, \ldots, p$. Denoting $H = \begin{bmatrix}H_1 \ldots H_p\end{bmatrix} = \left(C^\top\Gamma_y^\top\right) \otimes L$ and $\omega = \mathrm{vec}\,(LTC)$, the optimization problem (14) can be rewritten as

$$\begin{aligned} \min_{\eta_1,\ldots,\eta_p} \quad & \sum_{i=1}^p [\Gamma_y]_{ii}V_{ii}\|\eta_i\|^2 \\ \text{s.t.} \quad & \sum_{i=1}^p H_i\eta_i = \omega. \end{aligned} \tag{25}$$

Proof: Recall that the cost J_e in (7) is given by $J_e = \mathrm{tr}(\tilde{\Sigma}) = \mathrm{tr}\left(\left(W + L\tilde{T}\Gamma_y V\Gamma_y T^\top L^\top\right)Z_e\right)$, as derived in (14). As shown in Proposition 1, the optimal solution is independent of the constant terms W and $L^\top Z_e L$, which can be replaced with 0 and I, respectively. Since V and Γ are diagonal, one can write the new objective function as $\mathrm{tr}\left(\tilde{T}\Gamma_y V\Gamma_y\tilde{T}^\top\right) = \mathrm{tr}\left(\sum_{i=1}^p[\Gamma_y]_{ii}V_{ii}\eta_i\eta_i^\top\right) = \sum_{i=1}^p[\Gamma_y]_{ii}V_{ii}\|\eta_i\|^2$. The model-matching constraint follows directly by applying the vectorization operation. ■

C. Proof of Theorem 2

The proof closely follows that of [33, Thm. 1]. For $t > t_0$ such that $t_n^{(q)} \le t < t_f^{(q+1)}$, we have $x(t) = \prod_{i=1}^{q+1} e^{A_c(t-t_n^{(i)})}e^{(A_c+\Delta^{(i)}[\bar{k}])(t_n^{(i)}-t_r^{(i)})}e^{(A-B\Gamma_u^{(i)}K)(t_r^{(i)}-t_0^{(i)})}x_0$. Using the Assumption 3, we derive the upper bound

$$\|x(t)\| \le \prod_{i=1}^{q+1}\left(e^a\right)e^{-\lambda_n T_n(t)}e^{-\lambda_r T_r(t)}e^{\lambda_f T_f(t)}\|x_0\|,$$

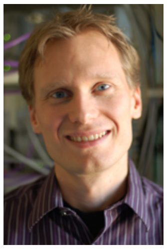
where $T_f(t)$, $T_r(t)$, and $T_n(t)$ are the total time for which the corresponding modes in (22) are active, with $t - t_0 = T_f(t) + T_r(t) + T_n(t)$. Note that, by design of the reconfiguration scheme, $t_r^{(i)} - t_0^{(i)} \le \tau_c(\bar{k})$ holds for all i, thus $T_f(t)$ is upperbounded by $T_f(t) \le N_f(t_0, t)\tau_c(\bar{k})$. From this bound, supposing $\lambda_r \le \lambda_n$, and Assumptions 2 and 4, we have

$$\begin{aligned} \frac{T_f(t)}{t-t_0} \le \frac{\tau_c(\bar{k})}{\tau_f} + \frac{N_0\tau_c(\bar{k})}{t-t_0} &\le \frac{N_0\tau_c(\bar{k})}{t-t_0} + \left(1 - \frac{\tau_c(\bar{k})}{\tau_f}\right)\frac{\lambda_r - \lambda^\star}{\lambda_f + \lambda^\star} \\ &\le \frac{N_0\tau_c(\bar{k})}{t-t_0} + \frac{\lambda_r - \lambda^\star}{\lambda_f + \lambda^\star}\frac{T_r(t)}{t-t_0} + \frac{\lambda_n - \lambda^\star}{\lambda_f + \lambda^\star}\frac{T_n(t)}{t-t_0}. \end{aligned}$$

Thus we reach the inequality $\lambda_f T_f(t) - \lambda_r T_r(t) - \lambda_n T_n(t) \le -\lambda^\star(t - t_0) + (\lambda_f + \lambda^\star)N_0\tau_c(\bar{k})$. Defining $c = a(N_0 + 1) + (\lambda_f + \lambda^\star)N_0\tau_c(\bar{k})$, the proof follows by using Assumption 2 to derive the inequality $\|x(t)\| \le ce^{-(\lambda^\star - \frac{a}{\tau_f})(t-t_0)}\|x_0\|$ and Assumption 4 to observe that $\lambda^\star - \frac{a}{\tau_f} > 0$. The proof concludes by observing that the lower bound on τ_f stated in the theorem satisfies Assumption 4 with $\lambda^\star = \arg\max_\lambda\left\{\frac{\mathrm{\lambda_f} + \mathrm{\lambda_r}}{\mathrm{\lambda_r} - \lambda}\mathrm{\tau_c}(\bar{\mathrm{k}}),\ \frac{\mathrm{a}}{\lambda}\right\} < \mathrm{\lambda_r}$.

André M.H. Teixeira is an Assistant Professor at the Faculty of Technology, Policy and Management, Delft University of Technology. He received the M.Sc. degree in electrical and computer engineering from the Faculdade de Engenharia da Universidade do Porto, Portugal, in 2009, and the Ph.D. degree in automatic control from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2014. From 2014 to 2015, he was a Postdoctoral Researcher at the Department of Automatic Control, KTH Royal Institute of Technology. His main research interests include cyber-secure and resilient control systems, distributed fault detection and isolation, distributed optimization, and power systems. He was the finalist for the Best Student-Paper Award from the IFAC NecSys in 2012, the recipient for the Best Student-Paper Award from the IEEE Multi-Conference on Systems and Control in 2014, and selected as the finalist for the European PhD Award in 2016 by the European Embedded Control Institute. One of his publications is listed in ACM Computing Review?s Notable Computing Books and Articles of 2012.

José Araújo is currently a Senior Researcher on Device Technologies at Ericsson Research in Stockholm, Sweden. He received the M.Sc. degree in electrical and computer engineering with a specialization in control and robotics in 2008 from the Faculty of Engineering, University of Porto (FEUP), Portugal and the Ph.D. degree in Automatic Control in 2014 from KTH Royal Institute of Technology, Sweden. He has held visiting researcher positions at the University of British Columbia (2008) and the University of California, Los Angeles (2012). His current research interests include the design and implementation of future device technologies and cyber-physical systems.



Henrik Sandberg is Professor at the Department of Automatic Control, KTH Royal Institute of Technology, Stockholm, Sweden. He received the M.Sc. degree in engineering physics and the Ph.D. degree in automatic control from Lund University, Lund, Sweden, in 1999 and 2004, respectively. From 2005 to 2007, he was a Post-Doctoral Scholar at the California Institute of Technology, Pasadena, USA. In 2013, he was a visiting scholar at the Laboratory for Information and Decision Systems (LIDS) at MIT, Cambridge, USA. He has also held visiting appointments at the Australian National University and the University of Melbourne, Australia. His current research interests include security of cyberphysical systems, power systems, model reduction, and fundamental limitations in control. Dr. Sandberg was a recipient of the Best Student Paper Award from the IEEE Conference on Decision and Control in 2004 and an Ingvar Carlsson Award from the Swedish Foundation for Strategic Research in 2007. He is Associate Editor of the IFAC Journal Automatica and the IEEE Transactions on Automatic Control.

Karl Henrik Johansson is Director of the Stockholm Strategic Research Area ICT The Next Generation and Professor at the School of Electrical Engineering, KTH Royal Institute of Technology. He received MSc and PhD degrees in Electrical Engineering from Lund University. He has held visiting positions at UC Berkeley, Caltech, NTU, HKUST Institute of Advanced Studies, and NTNU. His research interests are in networked control systems, cyber-physical systems, and applications in transportation, energy, and automation. He is a member of the IEEE Control Systems Society Board of Governors and the European Control Association Council. He has received several best paper awards and other distinctions, including a ten-year Wallenberg Scholar Grant, a Senior Researcher Position with the Swedish Research Council, the Future Research Leader Award from the Swedish Foundation for Strategic Research, and the triennial Young Author Prize from IFAC. He is member of the Royal Swedish Academy of Engineering Sciences, Fellow of the IEEE, and IEEE Distinguished Lecturer.