Distributed actuator reconfiguration in networked control systems [★]

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

ACCESS Linnaeus Centre, KTH Royal Institute of Technology,
Stockholm, Sweden.
(e-mail: {andretei, araujo, hsan, kallej}@kth.se)

Abstract: In this paper, we address the problem of distributed reconfiguration of first-order networked control systems under actuator faults. In particular, we consider the scenario where a network of actuators cooperates in order to recover from actuator faults. Such recovery is performed through a reconfiguration which minimizes the performance loss due to actuator faults, while guaranteeing that the same state trajectory is obtained. The design of the distributed reconfiguration scheme is proposed and evaluated in numerical examples.

Keywords: Distributed reconfiguration, reconfigurable control, networked control systems, actuator networks, distributed optimization, large-scale systems.

1. INTRODUCTION

Increasingly, control systems are operated over large-scale, complex networked infrastructures as in power networks, building automation, power plants and factories. The proliferation of low cost embedded systems with radio capabilities has enabled the deployment of systems which allow for increased performance and flexibility. However, these systems become increasingly complex but must be efficiently designed and operated. Several steps have been taken in this direction, as the development of resilient fault tolerant architectures and technologies [Ding, 2008, Blanke et al., 2006], and the introduction of plug-and-play control [Bendtsen et al., 2013] which reduces installation costs and increases flexibility. In this paper we focus on distributed actuator reconfiguration in networked systems. In the event of malfunction in actuators, sensors or other system components, control systems may exhibit poor performances or even become unstable if not properly designed [Blanke et al., 2006, Poovendran et al., 2012]. Thus, the design of fault-tolerant control systems is of major importance. A few examples of safety-critical systems that must be resilient to faults are power networks, aircrafts, nuclear power plants and chemical plants.

Since the 1970s, much research has been conducted in the field of Fault-Tolerant Control Systems, Fault detection and diagnosis (FDD) and Reconfigurable Control [Blanke et al., 2006, Zhang and Jiang, 2008, Ding, 2008]. The field of FDD deals with the identification that a fault exists and determines where it is located, while reconfigurable control proposes methods that reconfigure/recover a system after a fault has been detected and isolated. The objectives of reconfiguration are generally to obtain the stabilization of the system, maintaining the same state trajectory (also known as model-matching), achieving the same equilibrium point and/or minimizing the loss in performance inflicted by the fault. Many different types of faults in actuators, sensors 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 as in [Wu et al., 2000, Staroswiecki et al., 2007, Staroswiecki and Cazaurang, 2008, Staroswiecki and Berdjag, 2010, Richter et al., 2011]. Due to the increased complexity and size of current control systems, such techniques may be impractical due to technical and economical constraints [Åkerberg et al., 2011]. Through the increased computation and communication capabilities of devices in these systems, FDD has moved from a centralized task to a more distributed one. However, distributed FDD and reconfiguration to enable distributed fault tolerant systems has been much less explored. The architecture of such systems is explored in [Campelo et al., 1999, Voulgaris and Jiang, 2004, Jin and Yang, 2009]. In [Yang et al., 2010] 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 actuator reconfiguration for networked control systems with actuator faults. Note that this problem is similar to a distributed controller design problem, which is inherently complex in general due to the couplings between actuators arising from both the system dynamics and the input-to-state mapping. The paper focuses on distributed actuator reconfiguration scheme to handle actuator faults for first-order systems. Using the proposed scheme, healthy actuators are able to locally compensate for faults disabling a given set of actuators in the network. Application examples where distributed actuator reconfiguration is beneficial are, distributed control of wind-farms [Morrisse et al., 2012], farming and livestock systems [Bendtsen et al.,

[★] The first and second authors contributed equally to this work. This work is supported by the Knut and Alice Wallenberg Foundation, the Swedish Research Council (VR 2009-4565) and the HYCON2 EU project.

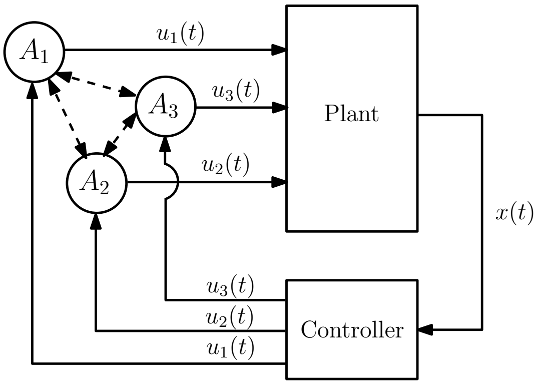


Fig. 1. Networked control system with a network of actuators A_1, A_2 and A_3.

2013] and data-server cooling systems [Ellsworth et al., 2008].

The proposed distributed algorithm is able to minimize the loss in performance under faults while achieving a desired closed-loop trajectory, where the trajectory of the system with and without the fault is the same. Numerical examples illustrate the performance of the proposed technique.

The rest of this paper is organized as follows. Section 2 presents the system components and architecture considered in this paper and formulates the problem we aim to solve. The centralized solution to the reconfiguration problem is presented in Section 3. In Section 4 the distributed solution is devised. Finally, numerical examples illustrate the distributed reconfiguration solution and Section 4 concludes this paper.

2. PROBLEM FORMULATION

The architecture of the networked control system considered in this paper is depicted in Fig. 1. We consider that there exists a network of redundant actuators with each device applying an individual control input to the plant. Each node is able to exchange information with its neighbors in the actuator network. The controller is responsible for computing control inputs for each individual actuator based on the current state of the system and transmitting this information to each individual actuator in the network. The individual components of the system are described below.

2.1 Plant and feedback controller

Assume the plant is a linear time-invariant system,

$$\dot{x}(t) = Ax(t) + B\Gamma u(t), \tag{1}$$

with a state $x(t) \in \mathbb{R}^n$ and input $u(t) \in \mathbb{R}^m$ given by a state-feedback

$$u(t) = -Kx(t), \tag{2}$$

where K is designed to render the closed-loop system (1) asymptotically stable. The closed-loop system matrix is denoted by $\bar{A} = A - BK$. From now on we drop the time argument in the variables x and u. The matrix $B \in \mathbb{R}^{n\times m}$ is assumed to have $\text{rank}(B) = k < m$, i.e., B does not have full column rank. The rank-deficiency of B is an expression of the redundancy in the actuation network. The matrix $\Gamma \in \mathbb{R}^{m\times m}$ is diagonal positive semi-definite with $[\Gamma]_{ii} = \gamma_i \in \{0, 1\}$. In fact, γ_i can be interpreted as a measure of the i-th actuator's effectiveness, where $\gamma_i = 1$ would mean that the actuator is functioning under nominal conditions, i.e., healthy actuator, while $\gamma_i = 0$ indicates otherwise. Under nominal conditions one has $\Gamma = I$. Each column of the matrix B is denoted as b_i.

2.2 Actuator network

The control system is assumed to have redundant actuators with sufficient computational capabilities to perform local computations and communicate with neighboring actuator nodes. The group of such actuators is denoted as an actuator network.

Let the actuator network be represented by the undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ with n vertices and edge set $\mathcal{E}$ with m edges. Each vertex $i \in \mathcal{V}$ represents an actuator, and an edge $e_k = (i, j) \in \mathcal{E}$ means that actuator nodes i and j can exchange information. Denote $\mathcal{N}_i = \{j | j \neq i, (i, j) \in E\}$ as the neighbor set of node v_i. The adjacency matrix $\mathcal{A}$ is defined as $\mathcal{A}_{ij} = 1$ for $i \neq j$ and $\mathcal{A}_{ii} = 0$. The degree matrix D is characterized by $D_{ii} = \sum_{j\in\mathcal{N}_i} \mathcal{A}_{ij}$. The Laplacian $\mathcal{L}$ of the actuator network graph is defined as $\mathcal{L} = D - \mathcal{A}$. Define $\mathcal{C}$ as the span of real symmetric matrices, $\mathcal{S}^n$, with sparsity pattern induced by the communication graph Laplacian $\mathcal{L}$, i.e., $\mathcal{C} = \{S \in \mathcal{S}^n | S_{ij} = 0 \,\text{if}\, \mathcal{L}_{ij} = 0\}$. Furthermore, denote $\mathcal{V}_f \subseteq \mathcal{V}$ as the set of faulty nodes such that $\gamma_i = 0$ if and only if $i \in \mathcal{V}_f$. Let the set of healthy nodes, $\gamma_i = 1$, be $\mathcal{V}_h = \{i \in \mathcal{V} | \, i \notin \mathcal{V}_f\}$ and consider $\mathcal{E}_h = \{(i, j) \in \mathcal{E} | \, i, j \in \mathcal{V}_h\}$. The subgraph $\mathcal{G}_h(\mathcal{V}_h, \mathcal{E}_h)$ corresponds to the graph of the healthy nodes. The number of healthy and faulty nodes are denoted as $n_h = |\mathcal{V}_h|$ and $n_f = |\mathcal{V}_f|$, respectively.

2.3 Fault model

An actuator failure is modeled as a change in the Γ matrix, where the column b_i representing the faulty actuator is multiplied by the coefficient γ_i. Denoting the new matrix by $\tilde{\Gamma}$, the system after a failure is represented as

$$\dot{x} = Ax + B\tilde{\Gamma}u. \tag{3}$$

We assume that actuators are able to detect and diagnose failures instantaneously in their own components and can notify their neighbors. Individual detection can be done through fault detection hardware ([Isermann and Raab, 1993, Isermann, 2011]).

2.4 Problem formulation

Assume that a fault has disabled several actuators in the network. Right after the fault occurs, the nodes in the network are able to know that a fault has occurred and reconfiguration must take place. After a failure, if the actuation signal is not modified, a loss in closed-loop performance may occur as well as the possible destabilization of the system. Therefore, reconfiguration from the failure is required. In particular, denote $\tilde{u} \in \mathbf{R}^m$ as the reconfigured control signals after a fault and let the reconfigured controller be

$$\tilde{u}(t) = -\tilde{K}x(t), \tag{4}$$

and $\tilde{A} = A - B\tilde{\Gamma}\tilde{K}$ as the respective closed-loop system matrix. The aim of the reconfiguration is to achieve model matching [Gao and Antsaklis, 1991, Staroswiecki and Cazaurang, 2008] as defined next.

Definition 1. Denote the nominal closed-loop system matrix as $\bar{A} = A - BK$ and consider the actuator faults described by $\tilde{\Gamma}$. A reconfiguration method computing a new controller $\tilde{K}$ so that $\tilde{A} = A - B\tilde{\Gamma}\tilde{K} = \bar{A}$ is denoted as an exact *model matching* method.

Throughout the paper we assume that exact model matching is always feasible. Since the system is over-actuated, there may exist different controllers achieving model matching. To reduce the possible choices of controllers attained by the reconfiguration scheme, we introduce the convex function $f(\tilde{K})$ to assess the cost of the controller $\tilde{K}$. The controller cost function $f(\tilde{K})$ may be interpreted as, for instance, the performance loss, the actuator degradation, or the number of active actuators. In this paper, we consider that $f(\tilde{K})$ corresponds to the performance loss induced by the change in the control law $\tilde{K}$ after a fault occurs.

The actuator network reconfiguration problem can be posed as follows.

Problem 1. How can each healthy actuator modify its own control input after a fault has occurred, so that exact model matching is ensured while minimizing the performance loss?

The reconfiguration scheme solving Problem 1 is designed to compute a controller minimizing the cost $f(\tilde{K})$ while achieving model matching, as described by

$$\begin{aligned} &\underset{\tilde{K}}{\text{minimize}} \quad f(\tilde{K}) \\ &\text{subject to } A - BK = A - B\tilde{\Gamma}\tilde{K}. \end{aligned} \tag{5}$$

Next we describe the centralized approach to solve Problem 1. Later, the centralized approach is modified so that it can be implemented in a distributed fashion based on local information exchange among actuators in the actuator network.

3. CENTRALIZED ACTUATOR RECONFIGURATION

In our work, we assume that the controller (2) is an optimal linear-quadratic (LQ) regulator for system (1). This controller is obtained as the minimizer of the following criterion

$$J_0 \triangleq \min_u \int_0^\infty x^TQx + u^TRu \; dt \tag{6}$$

where $Q \succeq 0$ and $R \succ 0$ and we assume that R is a diagonal matrix ([Kwakernaak and Sivan, 1972]). Moreover, we denote the elements of R^{-1} as β_i, $i = 1, ..., m$. Supposing the system is in nominal conditions, i.e. $\Gamma = I$, the optimal LQ controller is

$$u = -R^{-1}B^TPx \tag{7}$$

where P is the solution of the Riccati equation

$$A^TP + PA - PBR^{-1}B^TP + Q = 0. \tag{8}$$

It is well known that if (7) is applied to system (1) continuously, the optimal control cost obtained from (6) is $J_0 = x_0^TPx_0$ for an initial condition $x(0) = x_0$. Assuming the reconfiguration takes place instantly, the corresponding control cost is

$$\tilde{J}_0 \triangleq \int_0^{t_f} x^TQx + u^TRu \;\; dt + \int_{t_f}^\infty x^TQx + \tilde{u}^TR\tilde{u} \;\; dt, \tag{9}$$

and the performance loss induced by the fault and the controller $\tilde{K}$ is a convex function defined as

$$\begin{aligned} f(\tilde{K}) \triangleq \tilde{J}_0 - J_0 &= \int_{t_f}^\infty x^TQx + \tilde{u}^TR\tilde{u} \; dt \\ &- \int_{t_f}^\infty x^TQx + u^TRu \; dt \\ &= \int_{t_f}^\infty \tilde{u}^TR\tilde{u} - u^TRu \; dt \end{aligned} \tag{10}$$

The optimal centralized reconfiguration solving Problem 1 through the reconfiguration scheme (5) is now presented.

Proposition 1. The optimal controller $\tilde{K}^\star$ minimizing the loss in performance $f(\tilde{K})$ after a fault $\tilde{\Gamma}$ while ensuring exact model matching is

$$\tilde{K}^\star = {\tilde{R}^\star}{}^{-1}\tilde{\Gamma}B^TP = {\tilde{R}^\star}{}^{-1}\tilde{\Gamma}RK, \tag{11}$$

where $\tilde{R}^\star$ is the solution to the convex optimization problem

$$\begin{aligned} \underset{\tilde{R}}{\text{minimize}} \qquad & x_f^T\tilde{P}_o(\tilde{R})x_f \\ \text{subject to} \qquad & \bar{A} = A - BK \\ & \tilde{B}\tilde{R}^{-1}\tilde{B}^T = BR^{-1}B^T \\ \bar{A}^T\tilde{P}_o + \tilde{P}_o\bar{A} + PB\tilde{C}B^TP = 0, & \end{aligned} \tag{12}$$

where $\tilde{C} = \tilde{\Gamma}\tilde{R}^{-1}R\tilde{R}^{-1}\tilde{\Gamma} - R^{-1}$. Moreover, the centralized actuator network reconfiguration is attained by computing $\tilde{K}^\star$ and having $\tilde{u}^\star = \tilde{K}^\star x$ as the control signal after fault.

In order to prove Proposition 1, we need to derive the following lemmas. We begin by deriving the condition that guarantees model matching, followed by the performance loss in order to prove the above proposition.

Lemma 1. For $\tilde{K} = \tilde{R}^{-1}\tilde{\Gamma}B^TP$, exact model matching is achieved if

$$B\tilde{\Gamma}\tilde{R}^{-1}\tilde{\Gamma}B^T = BR^{-1}B^T. \tag{13}$$

Proof. Following Definition 1, model matching is guaranteed if the closed-loop matrix before fault is the same as after the fault, i.e.,

$$A - BK = A - B\tilde{\Gamma}\tilde{K} \tag{14}$$

where $\tilde{K} = \tilde{R}^{-1}\tilde{\Gamma}B^TP$. Recalling that $K = R^{-1}B^TP$ and rewriting (14) as (13) concludes the proof.

Under the assumption that model matching is achieved, the performance loss is characterized as follows.

Lemma 2. Assuming that model matching holds, the performance loss $f(\tilde{K}) = \tilde{J}_0 - J_0$ is given by

$$f(\tilde{K}) = x_f^T\tilde{P}_ox_f, \tag{15}$$

where $\tilde{P}_o$ is the solution to the Lyapunov equation

$$\begin{aligned} &(A - BK)^T\tilde{P}_o + \tilde{P}_o(A - BK) + \\ &\quad + PB\left(\tilde{\Gamma}\tilde{R}^{-1}R\tilde{R}^{-1}\tilde{\Gamma} - R^{-1}\right)B^TP = 0 \end{aligned} \tag{16}$$

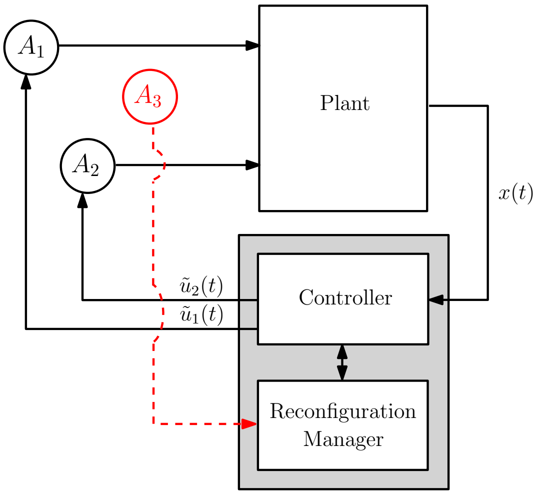


Fig. 2. Networked control system with centralized reconfiguration. Faults are reported by the actuators to the centralized controller. The reconfiguration manager is responsible for performing the reconfiguration. Dashed arrow represents the transmission of information related to faults.

Proof. Since model matching is achieved, the state trajectory of the closed-loop system before and after the fault is the same. Therefore the loss in performance between the non-faulty and the reconfigured system is given by

$$\begin{aligned} f(\tilde{K}) &= \int_{t_f}^{\infty} \tilde{u}^T R \tilde{u} - u^T R u \; dt \\ &= \int_{t_f}^{\infty} x^T P B \underbrace{\left(\tilde{\Gamma} \tilde{R}^{-1} R \tilde{R}^{-1} \tilde{\Gamma} - R^{-1}\right)}_{\tilde{C}} B^T P x \; dt \\ &= \int_{t_f}^{\infty} x_f e^{\bar{A}t^T} P B \tilde{C} B^T P e^{\bar{A}t} x_f \; dt, \end{aligned} \tag{17}$$

where $\bar{A} = A - BR^{-1}B^T P$ is the closed-loop system matrix and $\tilde{C} = \tilde{\Gamma}\tilde{R}^{-1}R\tilde{R}^{-1}\tilde{\Gamma} - R^{-1}$. Recalling the resemblance of $f(\tilde{K})$ to the observability Gramian of the pair $\left(A + BK, \, \tilde{D}\right)$ with $\tilde{D} = \tilde{C}^{\frac{1}{2}} B^T P$ ([Zhou et al., 1996]) concludes the proof.

We are now ready to derive the proof of Proposition 1.

Proof. [Proposition 1] Recall that the optimal controller after a fault is the solution to (5). We now show that (5) can be rewritten as (12).

The first constraint in (12) is the model matching constraint derived in Lemma 1.

As we aim at minimizing the loss in performance, the objective function in (12) is the one given in Lemma 2 as (15) with $\tilde{P}_o$ given by (16).

Fig. 2 depicts an example of a centralized reconfiguration that is performed by a system component denoted as reconfiguration manager. A fault occurs at actuator A_3, which detects that it is faulty and reports its faulty state to the reconfiguration manager which now knows $B\tilde{\Gamma}$. The reconfiguration manager then solves (12) in Proposition 1 to derive the new controller $\tilde{K}^\star$, which allows the calculation of $\tilde{u}_1^\star$ and $\tilde{u}_2^\star$. This value is then transmitted to the healthy actuators A_1 and A_2.

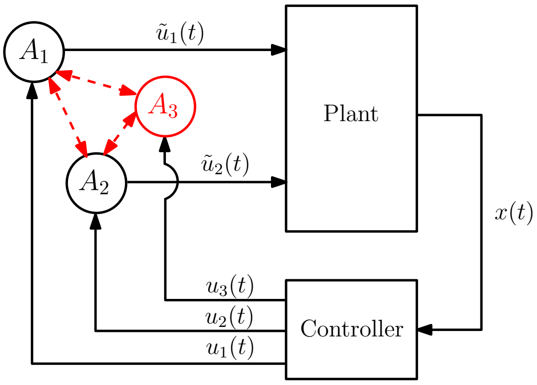


Fig. 3. Networked control system with distributed reconfiguration. Faults are detected by the actuators, which are also responsible for the reconfiguration. Reconfiguration is achieved through the communication among actuators in a distributed manner through the actuator network.

In the next section we propose the distributed reconfiguration solution to Problem 1.

4. DISTRIBUTED ACTUATOR RECONFIGURATION

To achieve distributed reconfiguration, the solution proposed in Proposition 1 should be implemented in a distributed fashion. We are now ready to introduce and solve the distributed version of Problem 1 for the class of scalar systems with $n = 1$.

Assumption 1. Each actuator i has b_i, b_j and β_j in memory for all $j \in \mathcal{N}_i$ and receives u_i from the controller.

An illustration of a distributed reconfiguration is shown in Fig. 3 where a fault occurs at actuator A_3. The actuators locally infer that actuator A_3 is no longer functioning, and so actuators A_1 and A_2 reconfigure themselves, computing and applying $\tilde{u}_1$ and $\tilde{u}_2$, respectively.

Here we propose a distributed scheme that implements (12) in a distributed manner for a scalar system. For the sake of notation, let R^{-1} ($\tilde{R}^{-1}$) be a diagonal with non-negative diagonal entries $[R^{-1}]_{ii} = \beta_i$ ($[\tilde{R}^{-1}]_{ii} = \tilde{\beta}_i$), $i = 1, ..., m$. Moreover, without loss of generality assume $\Gamma = I$ before the faults have occurred.

Proposition 2. Let $\tilde{\beta} = \left(\tilde{\beta}_1 \; \ldots \; \tilde{\beta}_m\right)$. For a scalar system with $n = 1$, the reconfiguration problem (12) can be rewritten as

$$\begin{aligned} &\min_{\tilde{\beta}} f(\tilde{\beta}) \\ &\text{s.t.} \;\; H\tilde{\Gamma}\tilde{\beta} = \omega \end{aligned} \tag{18}$$

where

$$f(\tilde{\beta}) = -\frac{(Px_f)^2}{2\bar{A}} \sum_{i=1}^{m} \left(\gamma_i^2 \tilde{\beta}_i^2 - \beta_i^2\right) b_i^2 \beta_i^{-1}, \tag{19}$$

$$H = \left({b_1}^2 \; \ldots \; {b_m}^2\right) \tag{20}$$

$$\omega = \sum_{i=1}^{m} \beta_i b_i^2. \tag{21}$$

Proof. Recall that the optimal reconfiguration ensuring model-matching and minimizing the loss in performance is given by (12). We begin by addressing the model-matching constraint.

Consider the equality constraint enforcing model-matching (13). Given that R and $\tilde{R}$ are diagonal, this expression can be rewritten as

$$\sum_{i=1}^{m} \tilde{\beta}_i (\gamma_i b_i)^2 = \sum_{i=1}^{m} \beta_i {b_i}^2. \tag{22}$$

As the term $\sum_{i=1}^{m} \beta_i b_i$ is invariant to the fault, we denote it by $\sum_{i=1}^{m} \beta_i b_i = \omega$. Thus (13) can be rewritten as $H\tilde{\Gamma}\tilde{\beta} = \omega$.

As for the objective function, recall that the performance loss under model-matching is given by Lemma 2. Using the Lyapunov equation (16) for the scalar case, the observability Gramian $\tilde{P}_o$ can be rewritten as $\tilde{P}_o = -\frac{P^2}{2\bar{A}} B \left(\tilde{\Gamma}\tilde{R}^{-1} R \tilde{R}^{-1}\tilde{\Gamma} - R^{-1} \right) B^T$. Since $\tilde{\Gamma}$, R, and $\tilde{R}$ are diagonal, $\tilde{P}_o$ can be rewritten as

$$\tilde{P}_o = -\frac{P^2}{2\bar{A}} \sum_{i=1}^{m} {b_i}^2 (\tilde{\beta}_i^2 \gamma_i^2 \beta_i^{-1} - \beta_i). \tag{23}$$

Replacing $\tilde{P}_o$ in the objective function concludes the proof.

Note that the objective function can be rewritten as $f(\tilde{\beta}) = c_1 \tilde{\beta}^T \Omega \tilde{\beta} + c_0$ with $\Omega = M\tilde{\Gamma}^2 R$, $M = \mathrm{diag}(H)$, $c_1 = -\frac{(Px_f)^2}{2\bar{A}} > 0$, and $c_0 = c_1 \sum_{i=1}^{m} -\beta_i^2 b_i^2 \beta_i^{-1}$. For simplicity, we replace the objective function (19) by $f(\tilde{\beta}) = \tilde{\beta}^T \Omega \tilde{\beta}$, which does not change the optimal solution of (18) since c_0 and c_1 are constants with respect to the decision variable $\tilde{\beta}$ and c_1 is positive.

4.1 Distributed optimization approach

Optimization problems of the form (18) are known as resource allocation problems [Xiao and Boyd, 2006, Ghadimi et al., 2011]. In order to efficiently solve this problem in a distributed manner, one can use a gradient method as proposed in [Xiao and Boyd, 2006]. Recalling that $\mathcal{C}$ is the span of real symmetric matrices with sparsity pattern induced by the communication graph Laplacian $\mathcal{L}$, we formulate the solution of the optimal distributed reconfiguration in Proposition 2 through the following theorem.

Theorem 1. Recall the objective function $f(\tilde{\beta}) = \tilde{\beta}^T \Omega \tilde{\beta}$ with $\Omega = M\tilde{\Gamma}^2 R$ and $M = \mathrm{diag}(H)$. The actuator network reconfiguration as described by (18) is achieved by running the following algorithm

$$\tilde{\beta}(k+1) = \tilde{\beta}(k) - WM\tilde{\Gamma}^2 R\tilde{\beta}(k), \tag{24}$$

where $\tilde{\beta}(0)$ is such that $H\tilde{\Gamma}\tilde{\beta}(0) = \omega$ and $W \in \mathcal{C}$ satisfies $H\tilde{\Gamma}W = 0$, $W(H\tilde{\Gamma})^T = 0$, and

$$\begin{bmatrix} W + W^T + \tilde{\Gamma}H^T(H\tilde{\Gamma}\tilde{\Gamma}H^T)^{-1}H\tilde{\Gamma} & W \\ W^T & \Omega^{-1} \end{bmatrix} \succ 0.$$

Moreover, the algorithm is distributed since W has the sparsity structure of the communication graph and $f(\tilde{\beta})$ is separable.

Proof. The proof that (24) converges to the optimal solution to (18) under the stated conditions follows directly from [Xiao and Boyd, 2006]. The distributed nature of the algorithm follows from the fact that (24) can be computed with only local information, given the sparsity of $W \in \mathcal{C}$ and the separability of $f(\tilde{\beta})$.

The distributed algorithm (24) requires the initial condition $\tilde{\beta}(0)$ to be a feasible solution of the constraint of problem (18). Such feasible solution is readily available by the following method.

Lemma 3. Let j 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. The initial condition $\tilde{\beta}(0)$ can be computed as

$$[\tilde{\beta}(0)]_i = \begin{cases} [\beta]_i, & i \notin \mathcal{J} \\ [\beta]_i + \nu_i b_i^{-2} b_j^2 [\beta]_j, & i \in \mathcal{J} \end{cases}, \tag{25}$$

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

Proof. Note that the computations are done locally, since by construction only the neighbors of the faulty node j are involved in the computations. The coefficient ν_i indicates how much i compensates for the control effort of the faulty node j before the fault. Moreover, having $[\tilde{\beta}(0)]_i = [\beta]_i + \nu_i {b_i}^{-2} b_j^2 [\beta]_j$, $\forall i \in \mathcal{J}$ and $\sum_{i \in \mathcal{J}} \nu_i = 1$ ensures that $H\tilde{\Gamma}\tilde{\beta}(0) = H\beta(0)$, and so $\tilde{\beta}(0)$ is a feasible solution. Hence, each healthy actuator i in the neighborhood of the faulty node must solely exchange and agree on the set of parameters ν_i.

Besides the need for a feasible initial condition $\beta(0)$, the distributed algorithm (24) also requires a suitable matrix W satisfying the conditions in Theorem 1. The following result provides a suitable matrix for before the occurrence of faults.

Lemma 4. Let $\lambda_i(I - W\Omega)$ denote the set of eigenvalues of $I - W\Omega$ ordered so that $|\lambda_1| \leq \cdots \leq |\lambda_{N-1}| \leq \lambda_N = 1$ and assume $\tilde{\Gamma} = I$. Then a suitable candidate for W is the normalized Laplacian of the network graph $W = -\delta M^{-1} \mathcal{L} M^{-1}$ where $M = \mathrm{diag}(H)$ and $\delta < 0$ is chosen sufficiently small in magnitude so that $|\lambda_{N-1}| < 1$.

Proof. The proof follows from Xiao and Boyd [2006, Section 4.2], by observing that $H\tilde{\Gamma} = \mathbf{1}^T \tilde{\Gamma} M$.

The optimal value of δ maximizing the convergence speed of (24) can be computed by solving an SDP problem proposed by Xiao and Boyd [2006]. The authors also propose a conservative heuristic to determine δ so that (24) converges, which in our setting corresponds to

$$\delta = \min_{i \in \mathcal{V}_h} \frac{\beta_i b_i^2}{D_{ii}}. \tag{26}$$

Note that such heuristic can be implemented in a distributed fashion by first electing as a leader the healthy node with the smallest value $\frac{\beta_i b_i^2}{D_{ii}}$ and then distribute this value for all the other nodes.

Given the original communication graph $\mathcal{G}$, assume W is chosen as stated in Lemma 4. Furthermore, denote $\tilde{\mathcal{G}}$ as the graph obtained from $\mathcal{G}$ by disconnecting the faulty nodes. The next result provides a suitable matrix $\tilde{W}$ for the distributed algorithm (24) after the occurrence of faults.

Lemma 5. Assume $\tilde{\Gamma} \neq I$ and that the subgraph of healthy nodes $\mathcal{G}_h$ is connected. A matrix $\tilde{W}$ compliant with Theorem 1 can be chosen as $\tilde{W} = -\tilde{\delta} M^{-1} \tilde{\mathcal{L}} M^{-1}$, where $\tilde{\mathcal{L}}$ is the Laplacian of $\tilde{\mathcal{G}}$.

Proof. Without loss of generality, assume the nodes are ordered so that the graph Laplacian is of the form

$$\tilde{\mathcal{L}} = \begin{bmatrix} \mathcal{L}_h & 0 \\ 0 & I_{n_f} \end{bmatrix},$$

where $\mathcal{L}_h$ is the Laplacian of the $\mathcal{G}_h$. Note that the corresponding matrix $\tilde{\Gamma}$ is given by

$$\tilde{\Gamma} = \begin{bmatrix} I_{n_h} & 0 \\ 0 & 0 \end{bmatrix}.$$

Recall the necessary conditions for convergence $H\tilde{\Gamma}\tilde{W} = 0$ and $\tilde{W}\tilde{\Gamma}H^T = 0$. Since $H\tilde{\Gamma} = \mathbf{1}^T\tilde{\Gamma}M$, we have $\mathbf{1}^T\tilde{\Gamma}\tilde{\mathcal{L}}M^{-1} = 0$ and $M^{-1}\tilde{\mathcal{L}}\tilde{\Gamma}\mathbf{1} = 0$, which concludes the proof.

Lemma 5 indicates that, once a fault occurs and the faulty nodes are removed, the healthy nodes do not need to recompute $\tilde{W}$. Instead, $\tilde{W}$ naturally arises from the communication graph $\tilde{\mathcal{G}}$, and the actuators can simply run the distributed algorithm from Theorem 1 by communicating with their healthy neighbors.

The results in this section are summarized in Algorithm 1, which describes the distributed actuator reconfiguration scheme.

Algorithm 1 Distributed Actuator Reconfiguration

```
(1) Detect and isolate actuator faults and disconnect the faulty actuators;
(2) Compute $\tilde{\beta}(0)$ by locally updating $\beta$ as per Lemma 3;
(3) Locally update $W$ according to Lemma 5;
(4) Run the distributed iterations from Theorem 1 to compute $\tilde{\beta}^\star$;
(5) For $t \geq t_f$, each healthy node $i$ computes and applies $u_i^\star = \tilde{\beta}_i^\star \beta_i^{-1} u_i$.
```

Remark 1. By design, Lemma 3 achieves a distributed reconfiguration solution where model matching is guaranteed. However, there is no minimization of the performance loss. This lemma can be used if model matching is a requirement of the reconfiguration and there is an interest on reducing the amount of communication among actuators. In fact, one can set $\nu_i = \frac{1}{|\mathcal{J}|}$, $i \in \mathcal{J}$ in (25) for the local neighborhood of the faulty nodes. By doing so, each healthy actuator in the neighborhood of the faulty actuator, only requires the knowledge of the total number $|\mathcal{J}|$ of healthy actuators in that neighborhood. Several methods to calculate the number of healthy nodes in a network have been recently proposed in the literature [Shames et al., 2012, Terelius et al., 2012, Cichon et al., 2012].

The convergence speed of the distributed optimization will depend on the actuator network connectivity as it directly influences the computation of $\tilde{\beta}_i^\star$ (step 4) for each actuator. In the case that a decision must be taken within a fixed number of steps and the algorithm has not yet converged to the optimal solution, the performance loss will be higher. However, model matching will always be guaranteed as per remark 1.

5. NUMERICAL EXAMPLES

5.1 Small example

Consider a scalar system with three actuators where none of the actuators has a fault in normal conditions, and characterized by

$$A = a, \quad B = (b_1 \ b_2 \ b_3), \ \Gamma = \mathrm{diag}(1, 1, 1),$$

in (1).

We now evaluate the reconfiguration of this system in a case of a complete failure of actuator 3 ($\gamma_3 = 0$, $\gamma_1 = \gamma_2 = 1$). We start by presenting the optimal centralized reconfiguration of the problem presented in Proposition 2, followed by the optimal distributed reconfiguration solution to Proposition 2 proposed in Algorithm 1.

Optimal centralized reconfiguration: As shown in Proposition 2, the performance loss $f(\tilde{\beta})$ is a quadratic function of $\tilde{\beta}$. Therefore one can easily derive analytically the optimal values of $\tilde{\beta}_1^\star$ and $\tilde{\beta}_2^\star$ as the solution to (18), while guaranteeing model matching [Boyd and Vandenberghe, 2004].

The optimal solution is given by

$$\begin{cases} \tilde{\beta}_1^\star = \dfrac{\omega}{b_1^2}\left(1 - \dfrac{\beta_2 b_2^2}{b_1\left(b_2^2\beta_2 + \beta_1\right)}\right) \\ \tilde{\beta}_2^\star = \dfrac{\omega\beta_2}{b_1(b_2^2\beta_2 + \beta_1)} \end{cases} \tag{27}$$

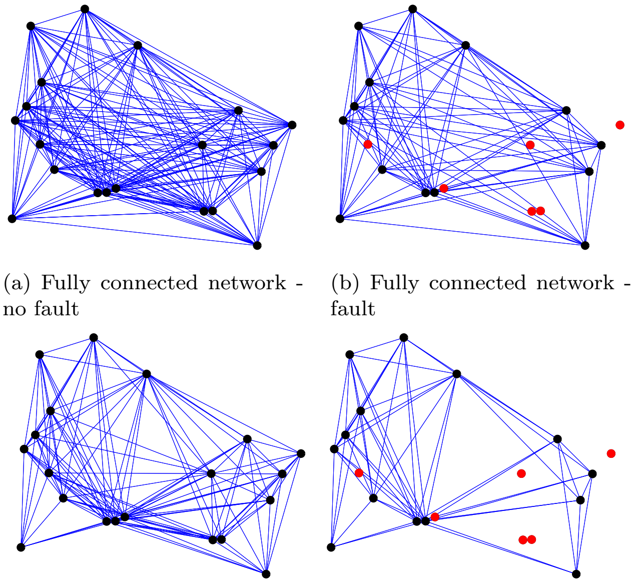
The control input applied by each actuator after the reconfiguration is given by

$$\begin{cases} \tilde{u}_1^\star = \tilde{\beta}_1^\star b_1 p x = \dfrac{\tilde{\beta}_1^\star}{\beta_1} u_1 \\ \tilde{u}_2^\star = \tilde{\beta}_2^\star b_2 p x = \dfrac{\tilde{\beta}_2^\star}{\beta_2} u_2 \\ \tilde{u}_3^\star = 0 \end{cases}$$

Optimal distributed reconfiguration: We assume that all actuators in the network can communicated directly and so the graph is fully connected. After the fault occurs, the nodes detect, isolate and disconnect the faulty actuator (step 1). Hence, actuators 1 and 2 exchange information among themselves and stop communicating with the faulty actuator 3. As per Lemma 3, nodes compute $\tilde{\beta}(0)$ locally based on β. In this case, we assume that the healthy actuators 1 and 2 select $\nu_1 = \nu_2 = \frac{1}{n_h} = \frac{1}{2}$ (step 2). The weight matrix is selected as $\tilde{W} = -\tilde{\delta}M^{-1}\tilde{\mathcal{L}}M^{-1}$, where $\tilde{\delta}$ is computed according to (26) (step 3). The values of $\tilde{\beta}(0)$ and W satisfy the conditions of Theorem 1, thus the optimal solution for $\tilde{\beta}^\star$ (step 4) is given by (27). Each actuator then applies the optimal control inputs to the plant (step 5).

5.2 Large example

We now evaluate the optimal distributed reconfiguration applied to a scalar system with $A = 1$ and where twenty actuators form an actuator network and six actuators completely fail ($n_h = 14$). The input matrix B is randomly generated with $0 < b_i \leq 10$. The fault occurs at time step $t = 0.1$s.



(a) Fully connected network - no fault

(b) Fully connected network - fault

(c) Sparse network - no fault

(d) Sparse network - fault

Fig. 4. Actuator networks for large scalar example. Fully connected (a and b) and sparse network (c and d). The healthy nodes are colored black and the faulty nodes are colored red.

In order to solve this problem we use algorithm 1. We evaluate our solution when applied to two different network topologies. In the first case, we deal with a fully connected network of actuators, which are depicted in Figs. 4a and 4b in case of no fault and with fault, respectively. In second case we consider a sparse actuator network which is shown in Figs. 4c and 4d for no fault and faulty situation, respectively. The initial values for each of the healthy nodes is defined using the Lemma 3 (step 2). In the first case, since the network is fully connected we can design $\nu_i = \frac{1}{|\mathcal{J}|} = \frac{1}{14}$, $i \in \mathcal{J}$ and $\mathcal{J} = \mathcal{V}_h$. For the second case, the parameter $\nu_i = \frac{1}{|\mathcal{J}|}$, $i \in \mathcal{J}$ which is the local neighborhood of the faulty nodes. In step 3, the values of W are achieved by Lemma 5 through the exchange of information among neighboring nodes.

The results for this example are presented in Figs. 5 and Fig. 6 for the first and second cases, respectively. The plots depict the evolution of the state $x(t)$, the control input $u(t)$, the performance loss $f(t)$ in (15). Additionally, we show the convergence of $\tilde{\beta}_i$, $i = 1, \ldots, 20$ which is calculated using Lemma 3. As expected, the same trajectory is obtained with faults and without faults. However, when the fault occurs, the control input values are increased in order to compensate for the actuator faults. The convergence to the optimal values $\tilde{\beta}^\star$ is faster for a fully connected graph than for the sparse network, as it is expected. In this example, for a fully connected graph, the convergence to a neighborhood of the optimal solution defined as $\|\tilde{\beta}(k) - \tilde{\beta}^\star\|_2 \leq 10^{-4}$ takes 9 steps, while for the sparse graph it takes 129 steps.

6. CONCLUSIONS AND FUTURE WORK

In this work, we developed a distributed reconfiguration method for scalar networked control systems under actuator faults. The proposed approach is able to minimize the loss in performance that occurs from the actuator faults, while guaranteeing that the same state trajectory is obtained. The optimal distributed reconfiguration is guaranteed to achieve the same solution as the optimal centralized reconfiguration, while only requiring local cooperating among healthy actuators. A large-scale numerical example demonstrates the effectiveness of our approach. Particularly, we show that the minimization of the loss in performance as well as the same state trajectory is achieved. Additionally, the speed of convergence of the distributed algorithm depends on the network connectivity.

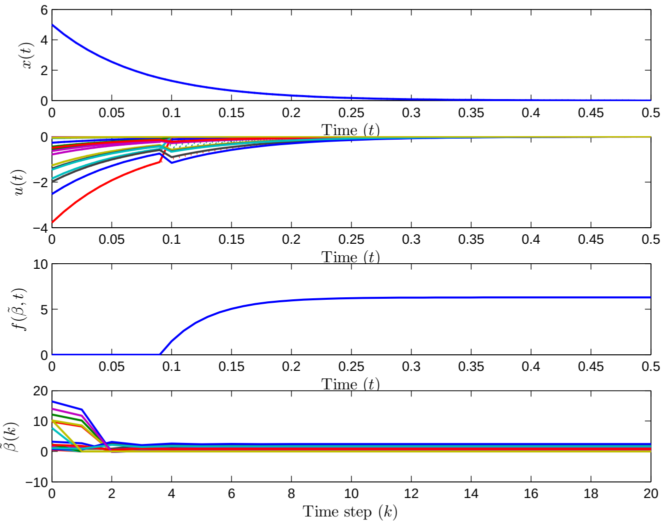


Fig. 5. Reconfiguration of a network of 20 actuators controlling a scalar system when 6 faults occur. The fault takes place at $t = 0.1$ s. Results for a fully connected graph show in Figs. 4a and 4b.

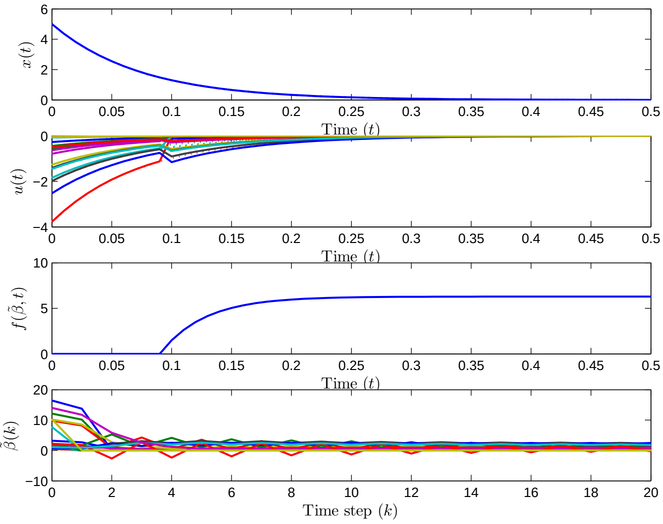


Fig. 6. Same setup as in Figure 5 but with a sparse graph depicted in Figs. 4c and 4d. Slightly slower convergence occurs due to a less connectivity among actuators.

As future work, we aim at analyzing the distributed reconfiguration problem with non-instantaneous and asynchronous detection and reconfiguration. Moreover, we will target the extension of the method developed in this paper to higher-order systems.

REFERENCES

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.

J. Bendtsen, K. Trangbaek, and J. Stoustrup. Plug-and-play control - modifying control systems online. *Control Systems Technology, IEEE Transactions on*, 21(1):79–93, 2013.

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

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

J.C. Campelo, F. Rodriguez, A. Rubio, R. Ors, P.J. Gil, L. Lemus, J.V. Busquets, J. Albaladejo, and J.J. Serrano. Distributed industrial control systems: a fault-tolerant architecture. *Microprocessors and Microsystems*, 23(2):103 – 112, 1999.

Jacek Cichon, Jakub Lemiesz, and Marcin Zawada. On size estimation protocols for sensor networks. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 5234–5239. IEEE, 2012.

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

M.J. Ellsworth, L.A. Campbell, R.E. Simons, and R.R.S. Iyengar. The evolution of water cooling for ibm large server systems: Back to the future. In *Thermal and Thermomechanical Phenomena in Electronic Systems, 2008. ITHERM 2008. 11th Intersociety Conference on*, pages 266–274, 2008.

Zhiquiang Gao and Panos J. Antsaklis. Stability of the pseudo-inverse method for reconfigurable control systems. *International Journal of Control*, 53(3):717–729, 1991.

Euhanna Ghadimi, Mikael Johansson, and Iman Shames. Accelerated gradient methods for networked optimization. In *American Control Conference (ACC), 2011*, pages 1668–1673, 2011.

Rolf Isermann. Supervision, fault-detection and diagnosis methods: a short introduction. In *Fault-Diagnosis Applications*, pages 11–45. Springer Berlin Heidelberg, 2011.

Rolf Isermann and Ulrich Raab. Intelligent actuators: Ways to autonomous actuating systems. *Automatica*, 29(5):1315 – 1331, 1993.

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.*, pages 2910–2915, 2009.

Huibert Kwakernaak and Raphael Sivan. *Linear optimal control systems*. Wiley Interscience, New York :, 1972.

K.J. Morrisse, G.F. Solimini, and U.A. Khan. Distributed control schemes for wind-farm power regulation. In *North American Power Symposium (NAPS), 2012*, pages 1–6, 2012.

R. Poovendran, K. Sampigethaya, S. K. S. Gupta, I. Lee, K. V. Prasad, D. Corman, and J. Paunicka. Special issue on cyber - physical systems [scanning the issue]. *Proceedings of the IEEE*, 100(1):6 –12, jan. 2012.

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*, 47(4):678–691, April 2011.

Iman Shames, Themistoklis Charalambous, Christoforos N Hadjicostis, and Mikael Johansson. Distributed network size estimation and average degree estimation and control in networks isomorphic to directed graphs. In *Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing, Allerton House, IL*, 2012.

M. Staroswiecki and F. Cazaurang. Fault recovery by nominal trajectory tracking. In *American Control Conference, 2008*, pages 1070–1075, 2008.

Marcel Staroswiecki and Denis Berdjag. A general fault tolerant linear quadratic control strategy under actuator outages. *International Journal of Systems Science*, 41 (8):971–985, 2010.

Marcel Staroswiecki, Hao Yang, and Bin Jiang. Progressive accommodation of parametric faults in linear quadratic control. *Automatica*, 43(12):2070 – 2076, 2007.

Hakan Terelius, Damiano Varagnolo, and Karl Henrik Johansson. Distributed size estimation of dynamic anonymous networks. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 5221–5227. IEEE, 2012.

P.G. Voulgaris and Shengxiang Jiang. Failure-robust distributed controller architectures. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 1, pages 513–518 Vol.1, 2004.

N.Eva Wu, Kemin Zhou, and Gregory Salomon. Control reconfigurability of linear time-invariant systems. *Automatica*, 36(11):1767 – 1771, 2000.

L. Xiao and S. Boyd. Optimal scaling of a gradient method for distributed resource allocation. *Journal of Optimization Theory and Applications*, 129:469–488, 2006.

Inseok Yang, Donggil Kim, and Dongik Lee. Fault-tolerant control strategy based on control allocation using smart actuators. In *Control and Fault-Tolerant Systems (SysTol), 2010 Conference on*, pages 377–381, 2010.

Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2):229 – 252, 2008.

K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.