Proceedings of the Estonian Academy of Sciences,
2018, **67**, 4, 325–336
https://doi.org/10.3176/proc.2018.4.01
Available online at www.eap.ee/proceedings

CONTROL THEORY

Observer-based residual generation for nonlinear discrete-time systems

Arvo Kaldmäe* and Ülle Kotta

Department of Software Science, School of Information Technologies, Tallinn University of Technology, Akadeemia tee 21, 12618 Tallinn, Estonia

Received 19 March 2018, accepted 15 May 2018, available online 5 October 2018

© 2018 Authors. This is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/).

Abstract. The paper studies the possibility of constructing observer-based residuals to detect faults in a nonlinear discrete-time system. The residuals are generated in such a manner that they detect one specific fault and are not affected by other faults and disturbances. Thus, a bank of residuals has been found to detect and isolate different faults in the system. An algebraic method called functions' algebra is used to construct an algorithm which computes the residuals. The key fact in residual generation is that any discrete-time observable system can be taken into the extended observer form. This form is used to construct the observer to estimate the system states under the assumption that there are no faults in the system. The state estimates are then compared to the measured values of the states. An example is added to illustrate the theoretical results. In the example it is also demonstrated how to combine the fault detection with the plant reconfiguration step of fault tolerant control.

Key words: nonlinear control, fault detection, residual generation, observability, algebraic methods.

1. INTRODUCTION

Fault tolerant control (FTC) is a branch of control engineering that aims at developing control strategies which can handle possible faults in the system. It means that the system is able to preserve stability, thus avoiding possible system breakdown and ensuring safety, and at the same time keeping also a satisfactory level of performance. Generally, FTC methods are classified into two types: passive and active. Passive controllers are designed to be robust against a class of presumed faults or are adaptive. Active FTC methods react to faults directly and depend heavily on fault detection and isolation (FDI).

A huge number of publications are available on different aspects of active FTC; see [23] for an overview. Since FDI is the first and essential part of active FTC, it is the most studied part of the FTC scheme; see [2,15,22,23] and references therein. There are many different approaches for FDI: model-based and data-based, quantitative and qualitative. The goal of model-based quantitative FDI methods is to construct signals, called residuals, which detect faults. The main approaches to the generation of residuals are the parameter estimation approach [7], parity space approach [20], and observer-based approach [15,17]. In this paper the observer-based approach is studied for nonlinear discrete-time systems.

In [17] a geometric approach was used to study the so-called fundamental problem of residual generation for nonlinear continuous-time systems, extending the solution from linear systems. The problem was

* Corresponding author, arvo@cc.ioc.ee

to construct a residual, which is sensitive to one specific fault and not influenced by other faults and disturbances affecting the system behaviour. Later, this solution was rewritten in [1] using an algebraic approach called functions' algebra, which allowed the generalization of the class of systems from control affine to general nonlinear systems.

In this paper the fundamental problem of residual generation is studied for nonlinear *discrete-time* systems using the functions' algebra approach. An algorithm is given (similar to the results of [1,17]) to construct a residual to detect a given fault in the system. The solutions in [1,17], as observer-based FDI schemes in general, depend heavily on the possibility of constructing an observer for a certain subsystem of the given system, which is, in general, a very difficult problem for nonlinear systems. In [17] additional restrictive assumptions were introduced in order to be able to construct the observer. The paper [1] only cites other articles, where observer construction is studied for nonlinear systems. The algorithm suggested in this paper does not depend on any restrictions. This is the key aspect and main novelty of our approach. Here we use the fact that for any observable *discrete-time* nonlinear system one can always construct an extended observer (depending also on the past values of outputs and inputs) to find the observer with *linear error dynamics* and construct a residual, which solves the fundamental problem of residual generation. That is, in principle, we do not have to make any additional restrictive assumptions to estimate system outputs, based on which the residual is constructed. An example of a three-tank system is used to demonstrate the usefulness of the approach for residual generation. In the example we also briefly talk about plant reconfiguration, which is the next possible step after FDI. In [8] the plant reconfiguration problem was studied, which can be connected with the FDI step described in this paper, since the same methodology is used. Connecting FDI and the reconfigurable control steps has been one of the main obstacles to applying different FDI approaches in practice [23].

Unlike the well-known differential geometric approach, used in [17], functions' algebra suits better for studying generic, not local properties. This means that we do not fix a point and work in the neighbourhood of this point; instead, the results of this paper are valid locally around every point where the computations can be done and all the necessary transformations can be defined uniquely. This allows us to simplify the presentation of the results compared to the local case, where a working point and its neighbourhood have to be fixed.

The paper is organized as follows. In Section 2 the problem statement is described. Section 3 gives an overview of functions' algebra and Section 4 of observability and observer construction. The main results are provided in Section 5, where the algorithm for residual generation is presented. The paper ends with an example and conclusions.

2. PROBLEM STATEMENT

Consider a nonlinear discrete-time system of the form

$$\begin{aligned} x(t+1) &= f(x(t),u(t),w(t)), \\ y(t) &= h(x(t)), \end{aligned} \tag{1}$$

where $x(t) \in X \subset \mathbb{R}^n$ is the state vector, $u(t) \in U \subset \mathbb{R}^m$ is the input vector, $w(t) \in W \subset \mathbb{R}$ is the fault signal, and $y(t) \in Y \subset \mathbb{R}^p$ is the output vector. The functions f and h are assumed to be analytic. The goal of fault detection is to construct a signal $r(t)$, called a residual, which is equal to zero when there is no fault ($w(t) = 0$) and becomes non-zero when the fault occurs ($w(t) \neq 0$).

The basic idea of observer-based residual generation is the following. Based on the system equations and the outputs/measurements, one can construct an observer for estimating the states of the system in the fault-free case. Then the residual can be constructed as difference between the estimated and measured functions. For instance, when we want to detect the fault $w(t)$ in the system

$$\begin{aligned} x_1(t+1) &= x_2(t)+w(t), \\ x_2(t+1) &= u(t), \\ y(t) &= x_1(t), \end{aligned}$$

it is enough to construct an observer for the case $w(t) = 0$ and define the residual as $r(t) = y(t) - \hat{x}_1(t)$, where $\hat{x}_1(t)$ is the estimate of the state variable $x_1(t)$. The observer together with $r(t) = y(t) - \hat{x}_1(t)$ is called a residual generator.

In this paper we assume additionally that the system equations depend also on some disturbances $d(t) \in D \subset \mathbb{R}^q$:

$$\begin{aligned} x(t+1) &= f(x(t), u(t), w(t), d(t)), \\ y(t) &= h(x(t)). \end{aligned} \tag{2}$$

Now the situation is more complex, since to be able to construct an observer, one has to find a subsystem, which does not depend explicitly on the disturbances. Finding a residual $r(t)$ that detects the fault but does not depend *explicitly* on the disturbances, and converges asymptotically to zero in the fault free-case, is called the fundamental problem of residual generation.

Moreover, when there are multiple faults, it is also important to isolate them, i.e., to determine which fault affects the system. Let $\bar{w}(t) \in \bar{W} \subset \mathbb{R}^s$ be a vector of faults. Then it may be possible to construct a vector of residuals $\bar{r}(t) \in \mathbb{R}^s$ such that every residual $\bar{r}_i(t)$ detects exactly one fault $\bar{w}_i(t)$ and is not affected by other faults. For that purpose one has to take $w(t) = \bar{w}_i(t)$ and $d(t) = (\bar{w}_1(t), \ldots, \bar{w}_{i-1}(t), \bar{w}_{i+1}(t), \ldots, \bar{w}_s(t))^{\mathrm{T}}$ for all i in the problem statement above and check the solvability conditions (Theorem 2 below) for every i.

Finally, note that the results of this paper are not global, but we study the generic case.

3. FUNCTIONS' ALGEBRA

In this section the method of functions' algebra is briefly described. In what follows, the notations x, $x^{[k]}$, $k \in \mathbb{Z}$ are used, instead of $x(t)$ and $x(t+k)$. Similar notations are used for the other variables as well as for functions. Compared to $x(t+k)$ the element $x^{[k]}$ must be understood as a variable and not as a function of time t.

The approach is developed for a discrete-time system given by its state-space equations, such as (1) (to describe the approach, we take $w(t) = 0$). Consider an infinite set of variables

$$\mathscr{V} = \{x, u, u^{[1]}, \ldots, u^{[k]}, \ldots\}$$

and denote by $\mathscr{F}$ a set of analytic functions[1] in a finite number of variables from the set $\mathscr{V}$. A specific subset of $\mathscr{F}$ is the set $\mathscr{F}_x$, which is used to denote the set of analytic functions depending only on the variables x. In functions' algebra we work with the vectors (of any finite dimension) whose elements are functions from $\mathscr{F}$ or $\mathscr{F}_x$. Denote the corresponding sets of vectors by $S_{\mathscr{F}}$ and $S_{\mathscr{F}_x}$, respectively. The elements of $S_{\mathscr{F}}$ or $S_{\mathscr{F}_x}$ are called vector functions. On the set $S_{\mathscr{F}}$ we define a preorder $\leq$.

Definition 1. *Given $\alpha, \beta \in S_{\mathscr{F}}$, one says that $\alpha \leq \beta$ if there exists a function γ such that $\beta = \gamma(\alpha)$.*

Based on the preorder $\leq$, one defines the equivalence relations $\cong$: the vector functions $\alpha, \beta \in S_{\mathscr{F}}$ satisfy the relation $\cong$ if $\alpha \leq \beta$ and $\beta \leq \alpha$.

The equivalence relation $\cong$ divides the elements of $S_{\mathscr{F}}$ into the equivalence classes. Let $S_{\mathscr{F}} \backslash \cong$ be the set of the equivalence classes. The relation $\leq$ was defined on the set $S_{\mathscr{F}}$, but can also be understood[2] as a relation on $S_{\mathscr{F}} \backslash \cong$, where it becomes a partial order. Then the pair $(S_{\mathscr{F}} \backslash \cong, \leq)$ becomes a lattice, since $\mathbf{0} := [x, u, \ldots, u^{[k]}] \leq \alpha \leq \mathbf{1}$ for all $\alpha \in S_{\mathscr{F}} \backslash \cong$, where k is chosen high enough. The vector function $\mathbf{1}$ corresponds to the equivalence class containing constant vector functions.

Remark 1. In the rest of the paper the equivalence classes are identified by their representatives, which are vector functions from $S_{\mathscr{F}}$ or $S_{\mathscr{F}_x}$ and thus, a vector function should always be understood as an equivalence

[1] Here we limit ourselves to analytic functions, but in principal one can consider a more general class of functions, such as smooth or even some non-smooth functions.

[2] When an equivalence class is represented by an element (a vector function) of this equivalence class.

class. The operations $\times$, $\oplus$, $\mathbf{m}$, $\mathbf{M}$ and the relation Δ, below, are defined on the set of equivalence classes, although in terms of the representatives. Also, since we work with equivalence classes, the sign "$=$" should be understood as "$\cong$".

Since $(S_{\mathscr{F}} \setminus \cong, \leq)$ is a lattice, we can define the binary operations $\times$ and $\oplus$ as

$$\alpha \times \beta := \inf(\alpha, \beta), \quad \alpha \oplus \beta := \sup(\alpha, \beta) \tag{3}$$

for all $\alpha, \beta \in S_{\mathscr{F}} \setminus \cong$. In (3) the infimum and supremum are considered with respect to the partial-order relation $\leq$. Similarly, the notions of maximal/minimal vector function mean maximality/minimality with respect to $\leq$.

The lattice $(S_{\mathscr{F}} \setminus \cong, \leq)$ will be connected to the system dynamics $f(\cdot)$ through the binary relation Δ. Note that Δ is defined only on $S_{\mathscr{F}_x} \setminus \cong$.

Definition 2. *Given $\alpha, \beta \in S_{\mathscr{F}_x} \setminus \cong$, one says that the ordered pair (α, β) satisfies the binary relation Δ, denoted as $\alpha\Delta\beta$, if for all $x \in X$ and $u \in U$ there exists a function f_* such that*

$$\beta(f(x,u)) = f_*(\alpha(x), u). \tag{4}$$

The binary relation Δ is used for the definition of the operators $\mathbf{m}$ and $\mathbf{M}$.

Definition 3. *(i) $\mathbf{m}(\alpha)$ is a minimal vector function $\beta \in S_{\mathscr{F}_x} \setminus \cong$ that satisfies $\alpha\Delta\beta$;*
(ii) $\mathbf{M}(\beta)$ is a maximal vector function $\alpha \in S_{\mathscr{F}_x} \setminus \cong$ that satisfies $\alpha\Delta\beta$.

By $\mathbf{M}^k$ we denote the consecutive application of the operator $\mathbf{M}$, i.e., $\mathbf{M}^k(\beta) = \mathbf{M}(\mathbf{M}^{k-1}(\beta))$ for $k \geq 1$, where $\mathbf{M}^0(\beta) := \beta$.

Important concepts in functions' algebra are the invariant vector functions.

Definition 4. *The vector function δ is said to be invariant with respect to the system dynamics $f(\cdot)$ or, said alternatively, f-*invariant *if $\delta\Delta\delta$. The vector function δ is said to be (h, f)-*invariant *if $[\delta \times h]\Delta\delta$.*

4. OBSERVER CONSTRUCTION

In this section we study the observability property and observer construction for a nonlinear discrete-time system. An algorithm that computes the maximal observable subspace of the state space is given. Also, a method to construct an observer for an observable system is described. Observer construction is a key element in the solution of the residual generation problem in Section 5.

Consider a nonlinear discrete-time system of the form

$$\begin{aligned} x(t+1) &= f(x(t), u(t)), \\ y(t) &= h(x(t)), \end{aligned} \tag{5}$$

where $x(t)$, $u(t)$, $y(t)$, f, and h are as above.

Assumption 1. System (5) is reversible.

The use of analytic functions in (5) and Assumption 1 guarantee that single-experiment observability of system (5) (necessary for observer construction) is equivalent to the multiple-experiment observability (easily studied by the methods of functions' algebra), see [21]. Therefore, in what follows, we just speak about the observability of system (5).

4.1. Observability

Some preliminary results on observability, using functions' algebra, are given in [12].

In plain words, observability means a possibility of recovering the state x of system (5) from the knowledge of the output y, the input u, and a finite number of their forward-shifts $y^{[k]}$, $u^{[k]}$, $k \in \mathbb{N}$. A formal definition is given via the observable space $O(x)$, which, in terms of functions' algebra, can be defined as

$$O(x) := x \oplus [h \times h^{[1]} \times \cdots \times h^{[n-1]} \times u \times \cdots \times u^{[n-2]}]^{\mathrm{T}}. \tag{6}$$

Now the observability can be defined.

Definition 5. *System (5) is called observable if $O(x) = x$. A vector function $\lambda(x) \in S_{\mathscr{F}_x}$ is said to be observable if $O(x) \leq \lambda(x)$.*

By Definition 5 the observable space $O(x)$ is a minimal vector function, in terms of $\leq$, which is observable.

Lemma 1. *The vector function $\lambda(x) \in S_{\mathscr{F}_x}$ is observable if and only if*

$$\lambda(x) \geq h(x) \times \mathbf{M}(h(x)) \times \cdots \times \mathbf{M}^k(h(x))$$

for some k.

Proof. By the definition of operator $\mathbf{M}$

$$h^{[k]} \geq \mathbf{M}^k(h) \times u \times u^{[1]} \times \cdots \times u^{[k-1]}.$$

The definition of the observable space and the observable vector function $\lambda(x)$ yield

$$\lambda \geq h \times h^{[1]} \times \cdots \times h^{[n-1]} \times u \times \cdots \times u^{[n-2]}. \tag{7}$$

Therefore, the observability of a vector function $\lambda(x) \in S_{\mathscr{F}_x}$ is equivalent to

$$\lambda \geq h \times \mathbf{M}(h) \times \cdots \times \mathbf{M}^{n-1}(h) \times u \times \cdots \times u^{[n-2]},$$

which, in turn, is equivalent to

$$\lambda(x) \geq h(x) \times \mathbf{M}(h(x)) \times \cdots \times \mathbf{M}^k(h(x)),$$

since $\lambda(x)$, $h(x)$, and $\mathbf{M}^k(h(x))$, $k \geq 1$, do not depend on $u^{[j]}$, $j \geq 0$. □

Algorithm 1 below computes the observable space of system (5).

Algorithm 1. Compute the sequence of vector functions θ_i, for $i \geq 0$, as

$$\begin{aligned} \theta_0 &= h(x), \\ \theta_{i+1} &= \mathbf{M}(\theta_i) \times h. \end{aligned}$$

Theorem 1. *The limit of Algorithm 1 is the observable space of system (5).*

Proof. As proved in [24], $\mathbf{M}(\delta_1 \times \delta_2) = \mathbf{M}(\delta_1) \times \mathbf{M}(\delta_2)$ for any two vector functions $\delta_1, \delta_2 \in S_{\mathscr{F}_x} \setminus \cong$. Thus, Algorithm 1 defines a decreasing (in terms of $\leq$) sequence of vector functions

$$\begin{aligned} \theta_0 &= h(x), \\ \theta_1 &= [h(x), \mathbf{M}(h(x))]^{\mathrm{T}}, \\ \theta_2 &= [h(x), \mathbf{M}(h(x)), \mathbf{M}^2(h(x))]^{\mathrm{T}}, \\ &\vdots \\ \theta_i &= [h(x), \mathbf{M}(h(x)), \ldots, \mathbf{M}^i(h(x))]^{\mathrm{T}}, \\ &\vdots \end{aligned}$$

Since $x \leq \theta_i$ for all $i \geq 0$, at some point j one has $\theta_{j+1} = \theta_j$, i.e.,

$$h(x) \times \mathbf{M}(h(x)) \times \cdots \times \mathbf{M}^j(h(x)) \leq \mathbf{M}^{j+1}(h(x)).$$

Define $O(x) = \theta_j$. Now, by Lemma 1, clearly all vector functions θ_i are observable and thus also $O(x)$ is observable. □

An obvious consequence of Theorem 1 is

Corollary 1. *The vector function $O(x)$ is f-invariant.*

4.2. Observer

Observer construction with linear error dynamics is in general a difficult problem for nonlinear discrete-time control systems. However, the problem becomes relatively easy when the state equations are in the observer form. Unfortunately, the conditions for transforming system equations (5) into such a form by state transformation are extremely restrictive [3,14]. One possibility of weakening the conditions is to search for an extended state transformation $\varphi(\cdot, \xi_1, \ldots, \xi_{2N}) : X \to X$, parametrized by $(\xi_1, \ldots, \xi_{2N})$ and defined by

$$z(t) = \varphi(x(t), y(t-1), \ldots, y(t-N), u(t-1), \ldots, u(t-N)), \tag{8}$$

such that in the new coordinates system (5) is transformed into the so-called extended observer form with buffer $N \in \{1, \ldots, n-1\}$:

$$\begin{aligned} z(t+1) &= Az(t) + \Phi(y(t), \ldots, y(t-N), u(t), \ldots, u(t-N)), \\ y(t) &= Cz(t), \end{aligned} \tag{9}$$

where the pair (C, A) is in dual Brunovsky form as defined in [13] and Φ is an n-dimensional column vector. For system (9) one can construct an extended observer with buffer N

$$\begin{aligned} \hat{z}(t+1) &= A\hat{z}(t) + K(y(t) - \hat{y}(t)) + \Phi(y(t), \ldots, y(t-N), u(t), \ldots, u(t-N)), \\ \hat{y}(t) &= C\hat{z}(t), \end{aligned} \tag{10}$$

where matrix K is chosen such that the eigenvalues of $A + KC$ are inside the unit circle. It has been proven in [5] that any single-output observable discrete-time system without inputs can be taken into the extended observer form with buffer $n-1$. The proof carries over to the input-dependent and multi-input multi-output case (MIMO). Thus, in principle, one can always construct an extended observer for an observable nonlinear discrete-time system with linear error dynamics that converges asymptotically to zero. In some cases it is possible to choose the buffer N smaller than $n-1$. The existence of the extended state transformation (8) to transform a given system (5) into the extended observer form (9) with buffer $N \in \{1, \ldots, n-1\}$ is studied for special cases in [4,6,9,11] and the general solution for MIMO systems is presented in [10].

For continuous-time systems the extended observer form depends not on the past values of $y(t)$ and $u(t)$ but on their derivatives [18]. In this case the conditions for the existence of the extended state transformation remain restrictive (though less restrictive than in the case when derivatives are not allowed), even when the derivatives up to the order $n-1$ are chosen.

5. RESIDUAL GENERATION

The method described below for finding residual generators consists roughly of three steps:
1. Find a subsystem of (2), not depending explicitly on disturbances d;
2. Find the observable space of the subsystem, computed in the previous step;
3. Construct an observer.

Step 1. First, let $\alpha^0(x)$ be a minimal vector function, such that $\alpha^0(f(x,u,w,d))$ does not depend on the disturbance d. Define also the vector function $\beta^0(x)$, similar to $\alpha^0(x)$, except that now we require that $\beta^0(f(x,u,w,d))$ should not depend on the fault w.

The next algorithm computes the minimal (h,f)-invariant vector function α, which satisfies $\alpha^0 \leq \alpha$.

Algorithm 2. Given α^0, compute the sequence of non-decreasing vector functions

$$\alpha^0 \leq \alpha^1 \leq \cdots \leq \alpha^i \leq \cdots$$

by using the formula

$$\alpha^{i+1} = \alpha^i \oplus \mathbf{m}(\alpha^i \times h) \tag{11}$$

for $i \geq 0$.

The sequence α^i, $i \geq 0$, of vector functions converges, i.e., for some j, $\alpha^{j+1} = \alpha^j$. It has been proven in [19] that $\alpha := \alpha^j$ is the minimal (h,f)-invariant vector function that satisfies $\alpha^0 \leq \alpha$.

Now, based on the vector function α and assuming that $h(x) \oplus \alpha(x) \neq \mathbf{1}$, one can construct the subsystem of (2), by defining $z = \alpha(x)$:

$$\begin{array}{rcl} z(t+1) & = & F(z(t),u(t),y(t),w(t)), \\ \tilde{y}(t) & = & \tilde{h}(z(t)), \end{array} \tag{12}$$

where $\tilde{h}(z(t))$ is the vector function $h(x) \oplus \alpha(x)$, written in terms of $z = \alpha(x)$. In (12) $u(t)$, $y(t)$, and $w(t)$ are considered input variables. Note that the function $F(\cdot)$ in (12) does not depend on the disturbance d, since by definition, the forward shift of α^0 does not depend on the disturbance and, because $\alpha^0(x) \leq \alpha(x) = z$, the forward shift of z does not depend on the disturbance either.

Step 2. The next step is to compute the observable space of system (12). This can be done via Algorithm 1. Let β be the result of the application of Algorithm 1 to system (12). By Corollary 1 the vector function β is F-invariant and one can construct the observable subsystem of (12):

$$\begin{array}{rcl} \eta(t+1) & = & G(\eta(t),u(t),y(t),w(t)), \\ \bar{y}(t) & = & \bar{h}(\eta(t)), \end{array} \tag{13}$$

where $\eta = \beta(z)$ and $\bar{h}$ is the function $\tilde{h}$ rewritten in η variables.

Step 3. The final step is to construct an observer for system (13) in the case when $w(t) = 0$. In order to be able to construct a residual for system (2), the condition $\beta^0 \times \beta = x$ must be true. Note that the dimension of vector β^0 is always $n-1$. Therefore, if β depends on the fault w, then the condition $\beta^0 \times \beta = x$ has to be satisfied. If the condition $\beta^0 \times \beta = x$ is violated, then system (13) does not depend on the fault w and one cannot construct a residual. The residual is defined as

$$r(t) = \bar{y}(t) - \hat{\bar{h}}(\hat{\eta}(t)),$$

where $\hat{\eta}(t)$ and $\hat{\bar{h}}$ are the estimates of $\eta(t)$ and $\bar{h}$, respectively, in the case when $w(t) = 0$. Clearly, in the fault-free case $r(t)$ converges asymptotically to zero, since $r(t)$ is defined as difference between measured and estimated functions and the error dynamics converges to zero. In the case when fault $w(t)$ occurs, the residual $r(t) \neq 0$ since (13) depends on the fault $w(t)$. Also, since system (13) does not depend explicitly on the disturbances $d(t)$, the residual $r(t)$ is not influenced by changes in $d(t)$.

The previous discussion is now concluded in Algorithm 3 below, which can be used to construct the residual $r(t)$ to detect the fault $w(t)$ in (2).

Algorithm 3. Given system (2), compute the following:

1. Find α^0 and β^0.
2. Compute the vector function α by Algorithm 2. Check whether $h(x) \oplus \alpha(x) \neq \mathbf{1}$. If not, then stop, the method cannot be used to compute the residual.
3. Construct system (12), where $z = \alpha(x)$ and $\tilde{h}(z(t))$ is the vector function $h(x) \oplus \alpha(x)$, written in terms of $z = \alpha(x)$.
4. Check whether system (12) satisfies Assumption 1. If not, then stop, otherwise compute the observable space β of system (12) as the limit of Algorithm 1.
5. Check whether $\beta \times \beta^0 = x$. If not, then stop, the method cannot find a residual to detect the fault w.
6. Construct system (13), where $\eta = \beta(z)$ and $\bar{h}$ is the function $\tilde{h}$ rewritten in η variables.
7. Find an observer for system (13), taking $w = 0$.
8. Define $r(t) = \bar{y}(t) - \hat{\bar{h}}(\hat{\eta}(t))$, where $\hat{\eta}(t)$ and $\hat{\bar{h}}$ are the estimates of $\eta(t)$ and $\bar{h}$, respectively.

The section can be concluded by the following theorem.

Theorem 2. *Under the assumption that subsystem (12) satisfies Assumption 1, one can construct a residual to detect the fault $w(t)$ in system (2) if $h \oplus \alpha \neq \mathbf{1}$ and $\beta \times \beta^0 = x$.*

Proof. When the conditions $h \oplus \alpha \neq \mathbf{1}$ and $\beta \times \beta^0 = x$ are satisfied, then, under the assumption that subsystem (12) is reversible, Algorithm 3 yields always a residual to detect the fault w. □

5.1. Comparison

A similar method for residual generation as described above has been used before for continuous-time input-affine systems of the form

$$\begin{aligned} \dot{x} &= f(x) + g(x)u + \ell(x)w + p(x)d, \\ y &= h(x) \end{aligned}$$

by using a differential geometric approach [17] or for general nonlinear continuous-time systems by functions' algebra [1]. Below, some comparisons are made with these papers.

All the specific vector functions and algorithms, described in this section, have direct counterparts in terms of differential geometry, as shown in Table 1, where the notations from [17] are used. In Table 1, $P = \text{span}\{p_1, \ldots, p_d\}$, where $p = (p_1, \ldots, p_d)$.

Note that in [17] there is no condition corresponding to $\alpha \oplus h \neq \mathbf{1}$. This condition is checked while computing $o.c.a.((\Sigma_*^P)^\perp)$, which stands for the maximal observability codistribution contained in $(\Sigma_*^P)^\perp$. That is, if $\alpha \oplus h = \mathbf{1}$, then the algorithm in [17] gives $o.c.a.((\Sigma_*^P)^\perp) = 0$, which corresponds to $\beta = \mathbf{1}$.

The results of [17] were rewritten in terms of functions' algebra in [1]. Moreover, in [17] and [1] not much is said about observer construction, which is a difficult problem for nonlinear systems, but necessary for residual construction. In [17] a residual is constructed under additional restrictive assumptions on the analogue of system (13).

6. EXAMPLE

Consider the control system

$$\begin{aligned} x_1^+ &= (u_1 + w_1)/\vartheta_1 - \theta_1(w_2)\sqrt{x_1 - x_2} + x_1, \\ x_2^+ &= u_2/\vartheta_2 + \theta_1(w_2)\sqrt{x_1 - x_2} - \theta_2\sqrt{x_2 - x_3} + x_2, \\ x_3^+ &= \theta_2\sqrt{x_2 - x_3} - \theta_3\sqrt{x_3 - \vartheta_7} + x_3, \\ y_1 &= x_1, \\ y_2 &= x_2, \end{aligned} \tag{14}$$

where the coefficients are: $\theta_1(w_2) = \vartheta_4(w_2)\sqrt{2\vartheta_8}/\vartheta_1$, $\theta_2 = \vartheta_5\sqrt{2\vartheta_8}/\vartheta_2$, $\theta_3 = \vartheta_6\sqrt{2\vartheta_8}/\vartheta_3$.

Table 1. Comparison between the vector functions used in this paper and corresponding objects, used in [17]

Functions' algebra	Differential geometry
α^0	$P^\perp$
β^0	$(\text{span}\{\ell\})^\perp$
α	$(\Sigma_*^P)^\perp$
β	$o.c.a.((\Sigma_*^P)^\perp)$
$\beta^0\times\beta = x$	$(\text{span}\{\ell\})^\perp +$ $o.c.a.((\Sigma_*^P)^\perp) = T^*X$

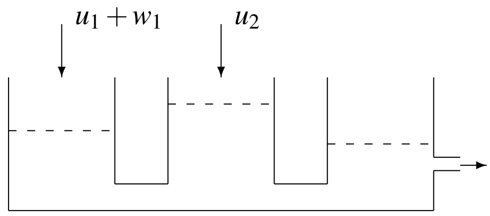

Fig. 1. Three-tank system.

Equations (14) constitute a modified sampled-data model of the well-known example of the three-tank system ([16], see Fig. 1). The system consists of three consecutively united tanks with areas of the cross-section ϑ_1, ϑ_2, and ϑ_3. The tanks are linked by pipes with areas of the cross-section ϑ_4 and ϑ_5. The liquid flows into the first and the second tank and out of the third one through the pipe with the area of the cross-section ϑ_6 located at height ϑ_7; ϑ_8 is the gravitational constant. The levels of liquid in the tanks are x_1, x_2, and x_3, respectively. The liquid levels in the first and the second tank are measured. Assume that two faults may occur. The first fault w_1 is an actuator fault that results in inaccurate addition of liquid into the first tank. The second fault w_2 is in the plant, which results in ϑ_4 being a function of w_2.

6.1. Fault detection

Our goal is to construct a residual to detect the actuator fault w_1. Thus, we look at w_2 as disturbance d. To construct the residual $r(t)$, we apply Algorithm 3 to system (14).

1. The straightforward computations yield $\alpha^0 = [x_1+x_2, x_3]^{\mathrm{T}}$ and $\beta^0 = [x_2, x_3]^{\mathrm{T}}$.
2. Compute by (11)

$$\alpha^1 = \alpha^0 \oplus \mathbf{m}(\alpha^0\times h) = [x_1+x_2,x_3]^{\mathrm{T}} \oplus \mathbf{m}([x_1,x_2,x_3]^{\mathrm{T}}) = [x_1+x_2,x_3]^{\mathrm{T}} \oplus [x_1,x_2,x_3]^{\mathrm{T}} = [x_1+x_2,x_3]^{\mathrm{T}} = \alpha^0$$

   and therefore, $\alpha = \alpha^0 = [x_1+x_2, x_3]^{\mathrm{T}}$.
3. The first condition of Theorem 2 is satisfied since $\alpha\oplus h = x_1+x_2 \neq \mathbf{1}$. Thus, one can construct system (12):

$$\begin{aligned} z_1^+ &= (u_1+w_1)/\vartheta_1 + z_1 + u_2/\vartheta_2 - \theta_2\sqrt{y_2-z_2},\\ z_2^+ &= \theta_2\sqrt{y_2-z_2} - \theta_3\sqrt{z_2-\vartheta_7} + z_2,\\ \tilde{y} &= z_1, \end{aligned} \tag{15}$$

   where $z = \alpha(x) = [x_1+x_2, x_3]^{\mathrm{T}}$ and $\tilde{y} = \alpha\oplus h = x_1+x_2$.
4. System (15) is obviously reversible and thus one can utilize Algorithm 1 to compute the observable space of system (15) when $w_1 = 0$:

$$\begin{aligned} \theta_0 &= \tilde{h} = z_1,\\ \theta_1 &= \tilde{h}\times\mathbf{M}(\theta_0) = z_1\times\mathbf{M}(z_1) = z_1\times[z_1,z_2]^{\mathrm{T}} = [z_1,z_2]^{\mathrm{T}}. \end{aligned}$$

   Thus, $\beta = [z_1,z_2]^{\mathrm{T}} = [x_1+x_2,x_3]^{\mathrm{T}}$.
5. The condition $\beta\times\beta^0 = [x_1+x_2,x_3]^{\mathrm{T}}\times[x_2,x_3]^{\mathrm{T}} = [x_1,x_2,x_3]^{\mathrm{T}}$ is satisfied and therefore, by Theorem 2, one can construct a residual to detect the fault w_1.
6. Since system (15) is observable, system (13) is equal to (15) for this example, i.e., $\eta = z$ and $\bar{y} = \tilde{y}$.
7. Following the work in [10], system (15), where $w_1 = 0$, can be transformed into an extended observer form with buffer 1 and thus an extended observer can be constructed to estimate the variables z_1 and z_2:

$$\begin{aligned} \hat{z}_1(t+1) &= \hat{z}_2(t) + K_1(\tilde{y}(t)-\hat{y}(t)) + \varphi(\cdot),\\ \hat{z}_2(t+1) &= K_2(\tilde{y}(t)-\hat{y}(t)),\\ \hat{y}(t) &= \hat{z}_1(t), \end{aligned} \tag{16}$$

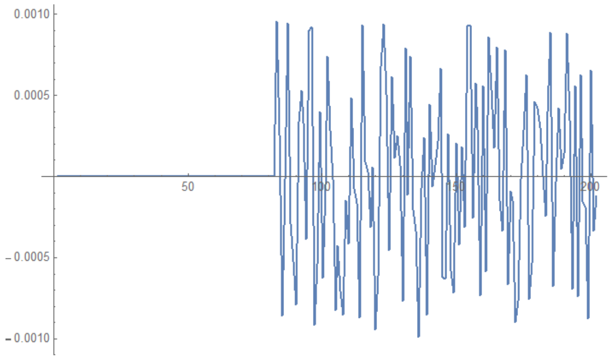

Fig. 2. Fault signal in simulations.

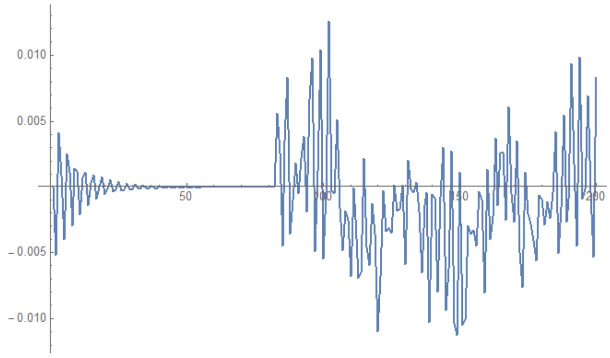

Fig. 3. Residual signal in simulations.

where

$$\begin{aligned} \varphi(\cdot) &= \tilde{y}(t)+u_1(t)/\vartheta_1+u_2(t)/\vartheta_2-\theta_2\sqrt{y_2(t)-\phi(\cdot)}, \\ \phi(\cdot) &= \theta_2\sqrt{y_2(t-1)-\psi(\cdot)}-\theta_3\sqrt{\psi(\cdot)-\vartheta_7}+\psi(\cdot), \\ \psi(\cdot) &= y_2(t-1)-\frac{1}{\theta_2^2}\left(\tilde{y}(t)-\tilde{y}(t-1)-u_1(t-1)/\vartheta_1-u_2(t-1)/\vartheta_2\right)^2. \end{aligned}$$

8. The residual to detect the fault w_1 can be written as $r(t)=\tilde{y}(t)-\hat{z}_1(t)$ or in terms of original system outputs $r(t)=y_1(t)+y_2(t)-\hat{z}_1(t)$.

Simulation results are presented in Figs 2 and 3, which show the effectiveness of the residual. Throughout the simulation the constant inputs $u_1=0.003$, $u_2=0$ and constant disturbance $w_2=0.001$ are used. Initial values are chosen as: $x_1(0)=3$, $x_2(0)=2$, $x_3(0)=1$, $\hat{z}_1(0)=5$, and $\hat{z}_2(0)=0$. The fault w_1 appears at the time instant $t=80$ and is generated as a random sequence of values between –0.001 and 0.001 (see Fig. 2). At first, the residual (Fig. 3) converges to zero as the estimate $\hat{z}_1(t)$ converges to the measured value of $y_1(t)+y_2(t)$. After the occurrence of the fault w_1 the residual clearly becomes nonzero. It is common that a fault is detected if the residual signal exceeds some threshold. This adds more robustness to fault detection. In our setting we have decoupled system disturbances from the residual signal, which means that we can possibly take this threshold smaller than with other methods.

6.2. Plant reconfiguration

In active FTC the FDI part is followed by fault accommodation or plant reconfiguration steps. Linking these parts together is one of the main challenges in active FTC [23]. Thus we shortly demonstrate how to use functions' algebra for the plant reconfiguration part in this example.

After detecting the fault w_1 in system (14), one can use the results of [8] on plant reconfiguration to find a maximal subsystem of (14), which can be decoupled from the effects of the fault w_1. Thus one can control the subsystem normally even when the fault w_1 appears. The paper [8] uses also functions' algebra to achieve its goal and therefore can be combined with the results of this paper. Assume now that the fault w_2 is not present and apply the results of [8] (more precisely Algorithm 3 in [8]) to show that the subsystem corresponding to the state variables x_2 and x_3 can be decoupled from the fault w_1. Therefore we show that one can control the liquid levels of the second and third tanks even when the fault w_1 occurs. Note that the starting point in [8] is the vector function β^0.

Now one can check that the vector function $\xi=[x_2,x_3]^{\mathrm{T}}$ can be decoupled from the fault w_1 by a static output feedback

$$u_2=\vartheta_2 v_2+\vartheta_2\theta_1\sqrt{y_1-y_2}. \tag{17}$$

After applying the feedback (17), system (14), where the fault w_2 is not present, takes the form

$$\begin{array}{rcl} x_1^+ & = & (u_1+w_1)/\vartheta_1-\theta_1\sqrt{x_1-x_2}+x_1, \\ x_2^+ & = & v_2-\theta_2\sqrt{x_2-x_3}+x_2, \\ x_3^+ & = & \theta_2\sqrt{x_2-x_3}-\theta_3\sqrt{x_3-\vartheta_7}+x_3, \\ y_1 & = & x_1, \\ y_2 & = & x_2. \end{array} \tag{18}$$

7. CONCLUSIONS

The problem of fault detection and isolation has been studied in this paper for nonlinear discrete-time systems. An algorithm has been derived to compute a residual, which detects exactly one fault and is not affected by other faults or disturbances. The solution is based on the construction of an observer for the subsystem, which depends on the specific fault but not on the other faults and disturbances. Since one can always construct an extended observer for any observable nonlinear discrete-time system, it is enough to check whether it is possible to construct subsystem (13), depending on the fault. The same mathematical approach (i.e., functions' algebra) has previously been used to study the plant reconfiguration problem [8], which possibly allows unifying the problems in the future studies.

ACKNOWLEDGEMENTS

The work of Arvo Kaldmäe was supported by the Estonian Centre of Exellence in IT (EXCITE) funded by the European Regional Development Fund. The publication costs of this article were covered by the Estonian Academy of Sciences.

REFERENCES

1. Berdjag, D., Christophe, C., Cocquempot, V., and Jiang, B. Nonlinear model decomposition for robust fault detection and isolation using algebraic tools. *IJICIC*, 2006, **2**(6), 1337–1354.
2. Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M. *Diagnosis and Fault-Tolerant Control*. Springer, Berlin, 2016.
3. Califano, C., Monaco, S., and Normand-Cyrot, D. Canonical observer forms for multi-output systems up to coordinate and output transformations in discrete time. *Automatica*, 2009, **45**(11), 2483–2490.
4. Huijberts, H. J. C. On existence of extended observers for nonlinear discrete-time systems. In *New Directions in Nonlinear Observer Design* (Nijmeijer, H. and Fossen, T. I., eds). Springer, London, 1999, 73–92.
5. Huijberts, H. J. C., Lilge, T., and Nijmeijer, H. Synchronization and observers for nonlinear discrete time systems. In *Proceedings of the European Control Conference*. Karlsruhe, Germany, 1999, 1–9.
6. Huijberts, H. J. C., Nijmeijer, H., and Pogromsky, A. Y. Discrete-time observers and synchronization. In *Controlling Chaos and Bifurcations in Engineering Systems*. CRC Press, Boca Raton, FL, USA, 1999, 439–456.
7. Isermann, R. Fault diagnosis of machines via parameter estimation and knowledge processing. *Automatica*, 1993, **29**(4), 815–835.
8. Kaldmäe, A., Kotta, Ü., Jiang, B., Shumsky, A., and Zhirabok, A. Faulty plant reconfiguration based on disturbance decoupling methods. *Asian J. Control*, 2016, **18**(3), 858–867.
9. Kaparin, V. and Kotta, Ü. Extended observer form for discrete-time nonlinear control systems. In *The 9th IEEE International Conference on Control and Automation*. Santiago, Chile, 2011, 507–512.
10. Kaparin, V. and Kotta, Ü. Transformation of nonlinear MIMO discrete-time systems into the extended observer form. *Asian J. Control*, 2018, DOI 10.1002/asjc.1824.
11. Kaparin, V., Kotta, Ü., and Mullari, T. Extended observer form: simple existence conditions. *Int. J. Control*, 2013, **86**(5), 794–803.
12. Kaparin, V., Kotta, Ü., Shumsky, A. Y., and Zhirabok, A. N. A note on the relationship between single- and multi-experiment observability for discrete-time nonlinear control systems. *Proc. Estonian Acad. Sci.*, 2011, **60**(3), 174–178.
13. Krener, A. J. and Respondek, W. Nonlinear observers with linearizable error dynamics. *SIAM J. Control Optim.*, 1985, **23**(2), 197–216.

14. Lee, W. and Nam, K. Observer design for autonomous discrete-time nonlinear systems. *Syst. Control Lett.*, 1991, **17**(1), 49–58.
15. Li, L. *Fault Detection and Fault-Tolerant Control for Nonlinear Systems*. Springer, 2016.
16. Patton, R., Clark, R., and Frank, P. *Fault Diagnosis in Dynamic Systems: Theory and Applications*. Prentice-Hall, NJ, USA, 1989.
17. Persis, C. D. and Isidori, A. A geometric approach to nonlinear fault detection and isolation. *IEEE Trans. Autom. Control*, 2001, **46**(6), 853–865.
18. Plestan, F. and Glumineau, A. Linearization by generalized input-output injection. *Syst. Control Lett.*, 1997, **31**(2), 115–128.
19. Shumsky, A. and Zhirabok, A. Unified approach to the problem of full decoupling via output feedback. *European J. Control*, 2010, **16**(4), 313–325.
20. Staroswiecki, M. and Comtet-Varga, G. Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 2001, **37**(5), 687–699.
21. Wang, Y. and Sontag, E. D. Orders of input/output differential equations and state-space dimensions. *SIAM J. Control Optim.*, 1995, **33**(4), 1102–1126.
22. Witczak, M. *Fault Diagnosis and Fault-Tolerant Control Strategies for Non-Linear Systems*. Springer, 2014.
23. Zhang, Y. and Jiang, J. Bibliographical review on reconfigurable fault-tolerant control systems. *Ann. Rev. Control*, 2008, **32**(2), 229–252.
24. Zhirabok, A. and Shumsky, A. *The Algebraic Methods for Analysis of Nonlinear Dynamic Systems*. Dalnauka, Vladivostok, Russia, 2008 (in Russian).

Olekutaastajal põhinev veatuvastussignaali arvutamine diskreetaja mittelineaarsetele süsteemidele

Arvo Kaldmäe ja Ülle Kotta

On uuritud võimalusi veatuvastussignaalide leidmiseks mittelineaarses diskreetaja süsteemis, kasutades selleks süsteemi olekute hinnanguid. Kirjeldatud signaalid konstrueeritakse nii, et nende väärtused oleksid nullid, kui süsteemis puudub konkreetne viga, sõltumata teistest vigadest ja häiringutest, ning nende väärtused muutuvad antud vea tekkimisel nullist erinevaks. See võimaldab automaatselt tuvastada ka vea asukoha ja liigi. Veatuvastusalgoritmi tuletuskäik kasutab võreteoorial põhinevat matemaatilist aparatuuri. Tulemuste uudsus seisneb selles, et algoritmis kasutatakse ära diskreetaja süsteemide omadust, mis võimaldab alati ja suhteliselt lihtsa vaevaga hinnata vaadeldava süsteemi olekuid eeldusel, et süsteemis puuduvad vead. Neid hinnanguid võrreldakse olekute mõõdetud väärtustega, mis erinevad hinnangutest juhul, kui viga mõjutab süsteemi. Artiklis kirjeldatud meetodit on kasutatud populaarse kolme anuma näite puhul. Lisaks on näites demonstreeritud, kuidas ühildada veatuvastuse protsess kontrolleri ümberarvutamisega, mis on veakindla juhtimise üks võimalikest järgmistest sammudest pärast vea tuvastamist.