



HAL
open science

Synergy of canonical control and unfalsified control concept to achieve fault tolerance

Tushar Jain, Joseph Julien Yamé, Dominique Sauter

► **To cite this version:**

Tushar Jain, Joseph Julien Yamé, Dominique Sauter. Synergy of canonical control and unfalsified control concept to achieve fault tolerance. 18th IFAC World Congress, IFAC WC'2011, Aug 2011, Milan, Italy. pp.CDROM. hal-00570348v2

HAL Id: hal-00570348

<https://hal.science/hal-00570348v2>

Submitted on 26 Mar 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Synergy of canonical control and unfalsified control concept to achieve fault tolerance

Tushar Jain* Joseph J. Yamé** Dominique Sauter***

* *Centre de Recherche en Automatique de Nancy (CRAN) UMR 7039,
Nancy Université (e-mail: tushar.jain@cran.uhp-nancy.fr)*

** *e-mail: joseph.yame@cran.uhp-nancy.fr*

*** *e-mail: dominique.sauter@cran.uhp-nancy.fr*

Abstract: A behavioral theoretical explanation of active fault tolerant control (FTC) problem is proposed in this note. Precisely, we present the concept of canonical controller and unfalsified control in behavioral context. The synergy of two control concepts re-configures the controller such that the system is tolerant to unknown faults. The main feature of the resulting FTC system is that it does not utilize any model-based fault detection and isolation procedure on-line. Therefore, it relies solely on the time-valued trajectories generated by the unknown plant in the closed-loop environment. In our approach to FTC, these trajectories formulate the control specifications that characterize certain *desired behavior*. Consequently, the controller re-designing process commence when this desired behavior is not satisfied.

Keywords: Fault tolerant control, canonical controller, unfalsified control

1. INTRODUCTION

In the past few years, the use of switching control is highly experienced in the field of adaptive control. The important trend in research is focused on the use of multi-model techniques and switching supervisory control. Therefore, a decision block and a bank of controller are designed. The block adjudicates which controller is most suitable for the current situation so that the closed loop can achieve the performance specifications. Usually it is seen that this concept pre-assumes the presence of at least one right controller in the controller bank [Morse (2008)]. Generally, this controller bank is constructed in the analysis and development phase before the system goes online. Switching control provides a descent formalization of control problem when no information about the plant is available. A widely used method which utilizes this concept since a decade is referred as *unfalsified control* [Safonov et.al. (1997)]. The underlying idea is to generate a fictitious reference, which assist in choosing the right controller for the current working mode of the plant. A much similar kind of feedback control can be seen in [Campi et.al. (2000)] but with an iterative procedure. Further, the stability and convergence issue concerned with adaptive switching control is discussed in [Stefanovic et.al. (2008)].

The theory of logic based switching control relies heavily on a bank of controllers. After its wider acceptability for adaptive control, it has recently been used to deal fault tolerant control [Blanke et.al. (2006)] problem. The preliminary work [Yamé et.al. (2008)] made an analysis of controller switching, which suggests an adaptive method to re-configure the control law for the current system state (faulty/normal). Generally, a fault is an unknown variable that changes the behavior of the system such that it no longer satisfies its purpose. The occurrence of fault

might lead to a degraded system performance or even the complete loss of the system function. Therefore, the goal of fault tolerant control is to respond to the after-effects of faults such that the faulty system is still able to achieve the desired behavior. In general, a fault tolerant control system is composed of two parts: a diagnosis part and a controller reconfiguration part. Various design methods exist in the literature that achieves the fault tolerance, see [Blanke et.al. (2006)] and the references therein. The *strong dynamical interaction* between the fault-diagnosis module and fault-accommodation module motivates further to adopt the switching control [Yamé et.al. (2008)]. The other issues are *unavailability of exact model, time constraints* etc.

Considering the restriction of limited controllers in switching approach to FTC, the prime issue is to answer what will happen if the right corrective controller is not available in the controller bank. In this case, will it be possible to maintain the fault tolerance property for the system? This paper deals with these questions. We assume that the plant is linear time-invariant and the mathematical model (representation based on state-space or transfer function) of the plant is unknown. We collect the input-output trajectory generated by the system in the finite horizon. It is also assumed that a set of specification is given that characterizes the desired behavior with its kernel representation. Thus, the purpose is to design a right controller directly based on these collected trajectories. Proceeding with the unfalsified switching control, one can now easily relieved the presence condition of right corrective correct controller in the bank while utilizing our approach. Since any representation of the plant is not known and we do not carry any estimation process for plant, the proposed model-free approach to FTC guarantees that the reconfiguration is fast and reliable. The theory behind our methodology

is based on the mathematical framework of behavioral systems. Here we make use of the canonical control [van de Schaft (2003)] concept in behavioral setting.

In the last decade, behavioral system theoretic approach has received an increasingly broader acceptance for modeling dynamical systems [Polderman et.al. (1997)]. One of the convincing reasons is that it does not start with the input/output distinction for describing how a system interacts with the environment rather, focuses on the set of system trajectories. In behavioral approach, a mathematical model is simply viewed as any relation among these time-valued trajectories (or system variables). Dynamically, these relations are constrained with the evolution of time. Therefore, the collection of time trajectories that model declares possibly called the *behavior* of the dynamical system.

2. CONTROL IN BEHAVIORAL CONTEXT

In this section, we brief the basic concepts of canonical controller and unfalsified control followed by definitions for robust feedback control and fault tolerant control problem. The definition [Trentelman et.al (1996); Yamé et.al. (2008)] of behavioral approach to dynamical systems is given as

Definition 1: A dynamical system is a triple, $\Sigma = (\mathcal{T}, \mathcal{W}, \mathfrak{B})$ with $\mathcal{T} \subset \mathbb{R}$ the time axis, \mathcal{W} , a set called the signal space, and $\mathfrak{B} \subset \mathcal{W}^{\mathcal{T}}$ the behavior ($\mathcal{W}^{\mathcal{T}}$ is the set of all \mathcal{W} -valued time trajectories). The behavior consists of functions $w : \mathcal{T} \rightarrow \mathcal{W}$.

In behavioral setting, the control problem is viewed as an interconnection of two dynamical sub-systems such that the interconnected system must follow certain restrictions. If $\Sigma_1 = (\mathcal{T}, \mathcal{W}, \mathfrak{B}_1)$ and $\Sigma_2 = (\mathcal{T}, \mathcal{W}, \mathfrak{B}_2)$ are two dynamical systems with the same time axis and the same signal space, then the interconnection of Σ_1 and Σ_2 shared by variable c , denoted as $\Sigma_1 \wedge_c \Sigma_2$, is defined as $\Sigma_1 \wedge_c \Sigma_2 := (\mathcal{T}, \mathcal{W}, \mathfrak{B}_1 \wedge_c \mathfrak{B}_2)$, where c is called the latent variable of the system. Thus, the behavior defined by $\Sigma_1 \wedge_c \Sigma_2$ consists simply of those trajectories $w : \mathcal{T} \rightarrow \mathcal{W}$ which are compatible with both the laws of Σ_1 (i.e., w belongs to \mathfrak{B}_1) and of Σ_2 (i.e., w belongs to \mathfrak{B}_2), where w is the manifest variable. The manifest and latent variables are also termed as to-be-controlled and control variables respectively.

Here, we consider an example to illustrate the control problem in behavioral setting. Taking the advantage in behavioral context, we omit the direction for the flow of information as shown in Fig. 1 where symbols have their usual meanings as in a classical feedback control.

For the brevity of explanation, consider the transfer function representation of plant is given by $G(\xi)$ with its co-prime polynomial R_y and R_u such that $G = R_y^{-1}R_u$. The behavior of plant is described as

$$\mathcal{P}_f = \{(w, c) \in \mathbb{R}^{q+p} \mid R(\xi)w = M(\xi)c\} \quad (1)$$

with appropriate matrices $R \in \mathbb{R}^{\bullet \times q}(\xi)$ and $M \in \mathbb{R}^{\bullet \times p}(\xi)$ where $\mathbb{R}^{m \times n}(\xi)$ denote the set of polynomial matrices of size $p \times q$ with indeterminate ξ , $w := (r, y)$ and $c := (e, u)$, and

$$R = \begin{bmatrix} 1 & -1 \\ 0 & R_y \end{bmatrix}, M = \begin{bmatrix} 1 & 0 \\ 0 & R_u \end{bmatrix} \quad (2)$$

Similarly, the controller is given by its transfer function $C(\xi) = C_e^{-1}C_u$ where C_e and C_u are the co-prime polynomials. It is described as

$$\mathcal{C} = \{c \in \mathbb{R}^p \mid H(\xi)c = 0\} \quad (3)$$

with $H \in \mathbb{R}^{\bullet \times p}(\xi)$. Now we will show how the interconnection between \mathcal{P}_f and \mathcal{C} results in a manifest controlled behavior, $\mathcal{K} = (\mathcal{P}_f \wedge_c \mathcal{C})_w$ defined as

$$\mathcal{K} = \{w \in \mathbb{R}^q \mid \exists c \in \mathcal{C} \text{ such that } (w, c) \in \mathcal{P}_f\} \quad (4)$$

In that case, we say that \mathcal{K} is implemented by \mathcal{C} which in connection with the hidden behavior, \mathcal{N} gives the implementability condition [Belur et.al. (2002)]:

$$\mathcal{N} \subset \mathcal{K} \subset \mathcal{P} \quad (5)$$

where \mathcal{P} is the manifest plant behavior and is given by

$$\mathcal{P} = \{w \in \mathbb{R}^q \mid \exists c \in \mathbb{R}^p \text{ such that } (w, c) \in \mathcal{P}_f\} \quad (6)$$

$$\mathcal{N} = \{w \in \mathbb{R}^q \mid (w, 0) \in \mathcal{P}_f\} \quad (7)$$

Example Given a plant $G(s) = \frac{s-1}{s(s+1)}$ and the controller is given by $C = -\frac{s+1}{s+2.6}$. Find the controlled behavior \mathcal{K} ?

$$\begin{aligned} \frac{y}{u} &= \frac{\xi - 1}{\xi(\xi + 1)} = \frac{R_u(\xi)}{R_y(\xi)} \\ [\xi(\xi + 1)]y + (-\xi + 1)u &= 0 \\ e &= r - y \end{aligned}$$

$$\mathcal{P}_f := \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & \xi^2 + \xi & 0 & -\xi + 1 \end{bmatrix} \begin{bmatrix} r \\ y \\ e \\ u \end{bmatrix} = 0$$

For controller \mathcal{C}

$$\begin{aligned} \frac{u}{e} &= -\frac{\xi + 1}{\xi + 2.6} = \frac{C_u}{C_e} \\ (\xi + 1)e + (\xi + 2.6)u &= 0 \end{aligned}$$

$$\mathcal{C} := \begin{bmatrix} 0 & 0 & \xi + 1 & \xi + 2.6 \end{bmatrix} \begin{bmatrix} r \\ y \\ e \\ u \end{bmatrix} = 0$$

$$\mathcal{K} := \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & \xi^2 + \xi & 0 & -\xi + 1 \\ 0 & 0 & \xi + 1 & \xi + 2.6 \end{bmatrix} \begin{bmatrix} r \\ y \\ e \\ u \end{bmatrix} = 0 \quad (8)$$

Since \mathcal{K} is a restricted behavior in \mathcal{P} and $\mathcal{K} = (\mathcal{P}_f \wedge_c \mathcal{C})_w$, there must exist [(Polderman et.al., 1997, Sec 6.2.3)] an unimodular matrix. The greatest common divisor between $R_u(\xi)$ and $C_e(\xi)$ is 1. Using Bezout identity, there exists polynomials $a(\xi)$, $b(\xi)$ such that $a(\xi).R_u(\xi) + b(\xi).C_e(\xi) = 1$. Define unimodular matrices $U_1(\xi)$, $U_2(\xi)$ as follows:

$$U_1(\xi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \xi + 1 & 0 & 1 \end{bmatrix}, U_2(\xi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & a(\xi) & b(\xi) \\ 0 & \xi + 2 & \xi - 1 \end{bmatrix}$$

Here $a(\xi) = -\frac{1}{3.6}$, $b(\xi) = \frac{1}{3.6}$. One easily checks that

$$\begin{aligned} U_2(\xi)U_1(\xi) &= \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & \xi^2 + \xi & 0 & -\xi + 1 \\ 0 & 0 & \xi + 1 & \xi + 2.6 \end{bmatrix} \\ &= \begin{bmatrix} 1 & -1 & -1 & 0 \\ \xi + 1 & -(\xi + 2)^2 & 0 & \xi + 0.8 \\ \xi^2 - 1 & (\xi + 1)(\xi^2 + 1.6\xi + 1) & 0 & 1.8 \end{bmatrix} \quad (9) \end{aligned}$$

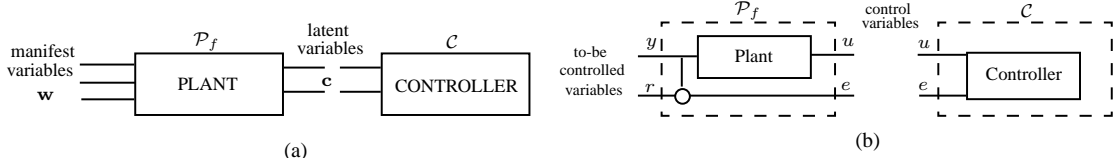


Figure 1. (a) Interconnection of subsystem; (b) Feedback control

$$\mathcal{K} := \begin{bmatrix} 1 & -1 & -1 & 0 \\ \xi + 1 & -(\xi + 2)^2 & 0 & \xi + 0.8 \\ \xi^2 - 1 & (\xi + 1)(\xi^2 + 1.6\xi + 1) & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ y \\ e \\ u \end{bmatrix} = 0 \quad (10)$$

It follows that the relation between r and y is given by the third row of above equation.

2.1 Canonical Controller

Here, we give a brief overview of the achievable behavior and the canonical controller discussed in [van de Schaft (2003)]. Given the manifest plant behavior $\mathcal{P} \in \mathbb{R}^q$ obtained by eliminating the variable c from \mathcal{P}_f , let $\mathcal{D} \in \mathbb{R}^q$ be the desired behavior, which the interconnected system is required to achieve, i.e. $\mathcal{K} \subseteq \mathcal{D}$. Here, we assume that the desired behavior is always implementable. The kernel representation of the desired behavior is represented by a polynomial matrix $D \in \mathbb{R}^{* \times q}(\xi)$ such that $w \in \mathcal{D}$ is given as

$$D(\xi)w = 0 \quad (11)$$

Given a kernel representation of desired behavior, we can construct a canonical controller such that the interconnected system satisfies the desired behavior \mathcal{D} and is defined as

$$\mathcal{C}_{can} = \{c \in \mathbb{R}^p \mid \exists \tilde{w} \text{ such that } (\tilde{w}, c) \in \mathcal{P}_f, \tilde{w} \in \mathcal{D}\} \quad (12)$$

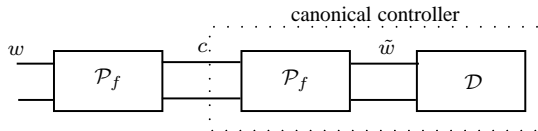


Figure 2. Canonical controller

We see in previous example that the controlled behavior is a restricted behavior in the manifest behavior, \mathcal{P} . Consequently, for \mathcal{D} to be implementable, it also satisfies $\mathcal{D} \subset \mathcal{P}$. Therefore, there must exist a polynomial matrix, L such that

$$L(\xi)R(\xi) = D(\xi) \quad (13)$$

Putting (1) in (13), we conclude

$$L(\xi)M(\xi)c(t) = 0 \quad (14)$$

Eq.(14) induces the kernel representation of a canonical controller acting only on the variable c .

2.2 Unfalsified Control Concept

The terminology *unfalsified control* was proposed in [Safonov et.al. (1997)] to the application in adaptive control system. The formal definition of *unfalsification* and *falsification* is given as:

Definition 2: A controller is said to be falsified by measurement information if this information is sufficient to deduce that the performance specification $(r, y, u) \in T_{spec}, \forall r \in \mathbb{R}$ would be violated if that controller were in

the feedback loop. Otherwise, the controller is said to be unfalsified.

Given $\mathbf{C} = \{C_1, C_2, \dots, C_N\}$, a finite, initial set of causally-left-invertible control laws, the goal of unfalsified control is to determine a $C \in \mathbf{C}$ such that for any unknown plant $G(\xi)$, representing the variations of the real system, the closed-loop system response satisfies the performance specifications involving the command signal $r(t)$, the command input $u(t)$, and the measured output $y(t)$. Unfalsified control concept uses real test data to determine $C \in \mathbf{C}$ instead of any representation of plant. Let (\bar{y}, \bar{u}) denote a set of closed-loop test data. Since a controller $C_i \in \mathbf{C}$ is causally-left-invertible, it is possible to determine a unique set of error signals, represented by \bar{e}_{C_i} that will result in the control signals \bar{u} if C_i is in the loop. The condition on \mathbf{C} to be causally-left-invertible is not very restrict [Stefanovic et.al. (2008)]. It follows that the corresponding command signals \bar{r}_{C_i} are given by

$$\bar{r}_{C_i} = \bar{e}_{C_i} + \bar{y} \quad (15)$$

Hence, \bar{r}_{C_i} is the set of command signals that would have yielded the signals \bar{u} and \bar{y} if the controller C_i were in the loop. Therefore, associated with each $C_i (i = 1, 2, \dots, N)$ is a triple $(\bar{r}_{C_i}, \bar{u}, \bar{y})$. Based upon this triple, we assume a cost functional $J_i : \mathbb{R}^3 \rightarrow \mathbb{R}, i = 1, \dots, N$ and associate performance specifications

$$J_i(\bar{r}_{C_i}, \bar{u}, \bar{y}) \leq \bar{\sigma}, \quad \forall i = 1, \dots, N \quad (16)$$

such that a controller is falsified if there exists $\bar{i} \in \{1, 2, \dots, N\}$, which satisfies

$$J_{\bar{i}}(\bar{r}_{C_{\bar{i}}}, \bar{u}, \bar{y}) > \bar{\sigma} \quad (17)$$

where i and \bar{i} are the indices of connected controller and to-be-connected controller respectively. Note, performance specifications are the control objectives that the interconnected system should satisfy.

2.3 Feedback control and Fault tolerant control

As we have seen above that the desired behavior or the to-be-achieved performance specifications play a very crucial role to guarantee FTC for the system. For re-designing the controller following section 2.1 and for selecting the right controller from the bank as seen in section 2.2, the information about the desired behavior is mandatory. Earlier, we deal with only one performance functional that characterizes a particular desired behavior. However, in real-time environment, we often deal with a set of specification so that if one of them is satisfiable or implementable in behavioral sense, then the fault tolerance is guaranteed. Here, we analyze the importance of desired behavior from the viewpoint of feedback control and fault tolerant control. We broadly categorized the desired behavior as the *stable* desired behavior and the *performance* desired behavior, thus, (16) is modified as

$$J_j(\bar{r}_{C_i}, \bar{u}, \bar{y}) \leq \bar{\sigma}_j, \quad j = 1, \dots, q, \quad i = 1, \dots, N \quad (18)$$

where j is the indices for a set of specifications. The definitions and problem statements are given as follows:

Definition 3: Consider a plant, \mathcal{P}_f connected with another subsystem, controller \mathcal{C} such that the interconnection results in a controlled behavior, \mathcal{K} , given by $\mathcal{K} = \mathcal{P}_f \wedge_c \mathcal{C}$. The controlled behavior must satisfy some constraints imposed by performance specification known as desired behavior, \mathcal{D} such that $\mathcal{K} \subseteq \mathcal{D}$.

It is possible to combine a number of design specifications as e.g. $\mathcal{K} \subseteq \mathcal{D}_{stable} \cap \mathcal{D}_{performance}$ which means that the interconnected subsystem must satisfy both, the stability specification given by \mathcal{D}_{stable} and the performance specification given by $\mathcal{D}_{performance}$. Here we say $\mathcal{D}_{performance}$ is strictly stronger or more restricted behavior than \mathcal{D}_{stable} as if the interconnected system satisfy $\mathcal{D}_{performance}$, it will also satisfy \mathcal{D}_{stable} . Therefore, $\mathcal{D}_{performance} \subset \mathcal{D}_{stable}$.

Now we can define some problems concerned with feedback control and fault tolerant control specifically in relation with desired behavior.

Problem 1: The feedback control design problem for closed loop stability of the system is given by

$$\mathcal{K} = \mathcal{D}_{stable}$$

where \mathcal{D}_{stable} is the stability specification.

Problem 2: The feedback control design problem for performance of the system is given by

$$\mathcal{K} = \mathcal{D}_{stable} \cap \mathcal{D}_{performance,1}$$

where \mathcal{D}_{stable} is the stability specification, $\mathcal{D}_{performance,1}$ is the performance specification.

Problem 3: The feedback control design problem for robust stability is given by

$$\mathcal{K}_\Delta = \mathcal{D}_{stable}$$

where \mathcal{K}_Δ is the controlled behavior as a function of the model uncertainty given by $\Delta \in \underline{\Delta}$.

Problem 4: The feedback control design problem for robust performance is given by

$$\mathcal{K}_\Delta = \mathcal{D}_{stable} \cap \mathcal{D}_{performance,1}$$

Problem 5: The *fault tolerant control* problem for stability is given by

$$\mathcal{K}_\Theta = \mathcal{D}_{stable}$$

where \mathcal{K}_Θ is the controlled behavior, which is subjected to change when an unknown fault occurs as function of some parameter given by $\Theta \in \underline{\Theta}$.

Problem 6: The *fault tolerant control* problem for performance is given by

$$\mathcal{K}_\Theta = \mathcal{D}_{stable} \cap \mathcal{D}_{performance,2}$$

In problem 6, it is assumed that $\mathcal{D}_{performance,1}$ is stronger than $\mathcal{D}_{performance,2}$. It should be noted that the first objective in fault tolerant control problem in connection with problem 5 and 6 is the stability of the interconnected system. The performance comes later, however, it might be possible to design an FTC system such that any degradation in performance can also be minimized, which evokes to incorporate some redundancies.

The solution to the above problems can now be defined as:

Definition 4: The feedback controller is said to be stabilizing if and only if the interconnected system $\mathcal{P}_f \wedge_c \mathcal{C}$ satisfies the stability specification of problem 1.

Definition 5: The feedback controller is said to achieve cost functional objective if and only if the interconnected system $\mathcal{P}_f \wedge_c \mathcal{C}$ satisfies the stability and performance specification of problem 2.

Definition 6: The feedback controller is said to be robustly stabilizing if and only if the interconnected system $\mathcal{P}_\Delta \wedge_c \mathcal{C}$ satisfies the stability specification of problem 3.

Definition 7: The feedback controller is said to satisfy robust performance if and only if the interconnected system $\mathcal{P}_\Delta \wedge_c \mathcal{C}$ satisfies the stability and performance specification of problem 4.

Definition 8: The feedback controller is said to be fault tolerant if and only if the interconnected system $\mathcal{P}_\Theta \wedge_c \mathcal{C}$ satisfies the stability specification of problem 5.

Definition 9: The feedback controller is said to be fault tolerant with performance specification if and only if the interconnected system $\mathcal{P}_\Theta \wedge_c \mathcal{C}$ satisfies the stability and performance specification of problem 6.

3. FAULT TOLERANT CONTROL AND CANONICAL CONTROLLER

The unfalsified control provides a powerful concept for handling the fault tolerant control problem. However, the scheme is effective if and only if a suitable controller exists in the controller bank for every faulty situation. This scenario seems to be impractical in many cases where the faulty condition is hard to reckon. Here, we provide the formulation of canonical control in terms of the system trajectories and the desired behavior. Mathematically, the design of canonical controller is proposed in [van de Schaft (2003)]. The basic formulation uses the idea of internal model principle (IMP). One of the applications of canonical controller considering IMP is shown in [Jain et.al. (2010)]. Later, in [Kaneko (2008)], the design is explicitly formulated for the given input-output data and the desired behavior.

3.1 Canonical Control using input-output data

Here, we assumed a plant to be single input single output for the brevity of explanation. As we have seen earlier, re-consider the co-prime polynomials of the plant given by $R_y(\xi)$ and $R_u(\xi)$. Given the control objectives, we can deduce the kernel representation of the desired behavior [Weiland et.al. (1997)]. Therefore, we write the kernel representation (11) as $[D_r \ -D_y] \begin{bmatrix} r \\ y \end{bmatrix} = 0$ Since $D_r(\xi)$

and $D_y(\xi)$ are polynomials with indeterminate ξ , they can also be represented as an operator that maps the reference trajectory to the output trajectory as, $y = T_d(\xi)r$ where $T_d = D_y^{-1}D_r$. Considering the kernel representations for plant (2), controller (3) and the desired behavior (11), we can directly write

$$C_u(D_r - D_y)R_u + D_r C_e R_y = 0 \quad (19)$$

Here we introduce a *latent* variable ℓ , and rearranging (19), it gives

$$R_u(D_r - D_y)(C_u \cdot \ell) + D_r R_y(C_e \cdot \ell) = 0 \quad (20)$$

Simplifying (20), it can be written as

$$[R_y D_r \quad R_u(D_r - D_y)] \begin{bmatrix} C_e \\ C_u \end{bmatrix} \ell = 0 \quad (21)$$

From (3) we write $u = C_u C_e^{-1} e$ and on using the latent variable ℓ , it gives

$$\begin{bmatrix} e \\ u \end{bmatrix} = \begin{bmatrix} C_e \\ C_u \end{bmatrix} \ell \quad (22)$$

On simplifying (21) and (22), it results in

$$[R_y D_r \quad (D_r - D_y)] \begin{bmatrix} 1 & 0 \\ 0 & R_u \end{bmatrix} \begin{bmatrix} e \\ u \end{bmatrix} = 0 \quad (23)$$

In section 2.1 we see that the canonical controller is parameterized using $R(\xi)$ and $M(\xi)$ from (2). From implementability condition (5), we see that $w \in \mathcal{N}$ is equivalent to $r = y$ and $R_y(\xi)y = 0$ which implies $D(\xi)w = (D_r(\xi) - D_y(\xi))y = 0$. This conditional implication also implies that $[D_r - D_y]$ includes $R_y(\xi)$ as a left factor. From this observation, we assign the desired behavior \mathcal{D} with the kernel representation $D'w = R_y[D_r - D_y] = 0$. This representation corresponds to the same desired behavior as (11). Therefore, a polynomial matrix, $L(\xi)$ satisfying (13) is described by

$$L = [R_y D_r \quad (D_r - D_y)] \quad (24)$$

Eq. (21) can be modified as

$$[C_e D_r \quad C_u(D_r - D_y)] \begin{bmatrix} R_y \\ R_u \end{bmatrix} \ell = 0 \quad (25)$$

re-arranging

$$[C_e \quad C_u] \begin{bmatrix} D_r & 0 \\ 0 & (D_r - D_y) \end{bmatrix} \begin{bmatrix} R_y \\ R_u \end{bmatrix} \ell = 0 \quad (26)$$

Since, we assumed that the plant is controllable anytime, therefore [Polderman et.al. (1997)], there exist an image representation of the plant behavior given by

$$\begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} R_y \\ R_u \end{bmatrix} \ell \quad (27)$$

From (26) and (27), we see that

$$[C_e \quad C_u] \begin{bmatrix} D_r & 0 \\ 0 & (D_r - D_y) \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix} = 0 \quad (28)$$

This represents the relationship of canonical controller for the given input-output trajectories and the desired behavior.

For realizing the canonical controller, we assigned a structure to the polynomials, $C_u(\theta) = \sum_{i=0}^m \theta_i \sigma^i$ and $C_e(\rho) = \sum_{i=1}^n \rho_i \sigma^i$ with unknown parameters $\theta := (\theta_0 \theta_1 \dots \theta_m^T)$ and $\rho := (\rho_0 \rho_1 \dots \rho_n^T)$ respectively. Let n_a and n_b be the degree of D_r and $(D_r - D_y)$. Consider a finite interval data of length $N + 1$, then (28) can be written as

$$C_e(\rho) D_r u_{[k, k+N]} = C_u(\theta) (D_r - D_y) y_{[k, k+N]} \quad (29)$$

where k is the current sample time. Denote $\bar{u}_{[k, k+N-n_a]} := D_r u_{[k, k+N]}$ and $\bar{y}_{[k, k+N-n_b]} := (D_r - D_y) y_{[k, k+N]}$. In the matrix form, it can be described as

$$\Gamma \begin{bmatrix} \theta \\ \rho \end{bmatrix} = \begin{bmatrix} \bar{u}_k \\ \vdots \\ \bar{u}_{k+N-n_a-m-1} \end{bmatrix} \quad (30)$$

where

$$\Gamma := \begin{bmatrix} \bar{y}_k & \cdots & \bar{y}_{k+n} & -\bar{u}_{k+1} & \cdots & -\bar{u}_{m+k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \bar{y}_{k+N-n_b-n-1} & \cdots & \bar{y}_{k+N-n_b-1} & -\bar{u}_{k+N-n_a-m} & \cdots & -\bar{u}_{k+N-n_a-1} \end{bmatrix}$$

The constructed canonical controller is an approximated feedback controller as we perform the least-square method. However, the approximation does not affect the closed-loop system to achieve the desired behavior. Since the approach lies into the category of *data-driven control synthesis*, we do not experience any exact data matching issues as seen in [Fujisaka et.al. (2005)]. Usually, in data-driven control estimating the plant model in any sense (Markovian parameters, state-space parameters, or finite impulse responses) cause these issues. In addition, the controller can be designed *bi-proper*, since we have assigned the fixed structure to the polynomials. The bi-properness condition is imposed on the newly designed controllers, keeping in mind the unfalsification procedure.

3.2 Fault Tolerant Control

Here, we pose our FTC problem in behavioral setting. Fault tolerant control is concerned with the control of faulty system [(Blanke et.al., 2006, Ch.7)]. This implies that we can consider any FTC problem as a control problem subjected to the state of the system (healthy/faulty). The solution of a control problem is completely defined by the triple $\langle \mathfrak{D}, \mathfrak{C}, \mathfrak{U} \rangle$ where

- \mathfrak{U} : is the set of admissible control laws. It defines the algorithm that can be implemented (a mapping from the time domain to the control space).
- \mathfrak{C} : is the set of constraints that a controlled system must satisfy over time. This represents the state and measurement equations of actual plant model in state-space representation.
- \mathfrak{D} : is the set of control/performance objectives. The system is expected to achieve these objectives, when controlled by a control law from the set \mathfrak{U} .

In *projection-based approach* to active FTC, generally the set \mathfrak{U} is finite. Subsequently, with an appropriate switching mechanism the right controller is selected. Effectively, the unfalsified control provides a powerful concept for handling the fault tolerant control problem that lies into the category of projection-based approach. However, the scheme in the present state is effective if and only if a suitable controller exists in the controller bank for every faulty situation. The concept of canonical controller provides a decent parameterization of controllers that achieves the desired objective. Using this method, the idea of switching control can be effectively extended from certain number of control laws in the sense if there is no control law available in \mathfrak{U} for any pre-defined faulty situation.

To obtain a perfect fault tolerant control system, it has been seen in the literature that one of among the triple must be known. In model based FTC, the constraints \mathfrak{C} are known anytime (healthy/post-fault mode). Later the control law is designed and appended to the set \mathfrak{U} . Here, we do not consider any estimation of constraints that implicitly involves *fault detection and isolation (FDI)*

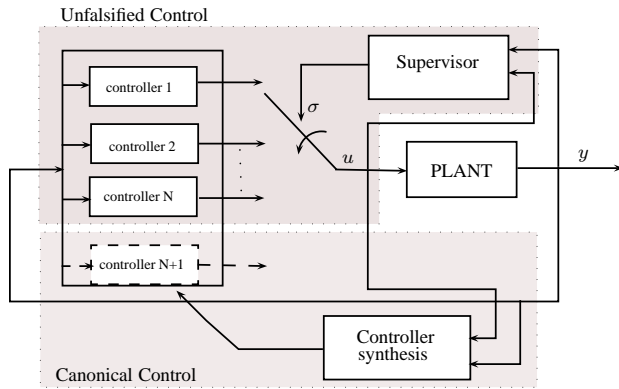


Figure: 3 Proposed FTC Scheme : Synergy of Unfalsified control and Canonical control

delay. In [Yamé et.al. (2008)], without determining \mathfrak{C} a pre-designed controller set is embedded in \mathfrak{U} off-line that guarantees the fault-tolerance property to the system for the anticipated set of faults. In this approach, we define only the control objectives (without determining \mathfrak{C} as well) that characterize a set of desired behaviors that is achievable/implementable anytime.

The extended unfalsified control approach to active FTC is depicted in Fig. 3. The synergy of canonical control and unfalsified control is shown by two individual shaded sub-systems. Here, we use the same structure as explained in [Yamé et.al. (2008)], therefore, we do not discuss the design of supervisor. The control objective is given by the performance functional in (16). In a finite controller bank, the supervisor, based on falsification/unfalsification inequality, efficiently switch the right controller in the loop on detecting the unpermitted behavior. Therefore, the performance functional must be *Cost-detectable*. This property does not take into account the plant model [Stefanovic et.al. (2008)], rather governed by the system trajectories only. Since we assumed that for a set of pre-determined faults at least one right corrective controller exist in the bank; consequently, the unfalsified control concept maintains the fault-tolerance property for the system.

Subsequently, we add a controller synthesis block to broaden the horizon of projection-based FTC. Eventually, the desired behavior is characterized by the same control objectives. Therefore, the approach does not conflict with the cost-functional chosen for the pre-designed controller and the on-line designed controller. Following the method discussed in section 2.1, the controller is designed on-line such that the fault-tolerance property for the system is still maintained. Our proposed scheme assumes that a large set of measurements is required to design a controller in comparison to measurements required for falsifying the controller from the bank. Therefore, on detecting the unpermitted behavior, the supervisor first evolves the controller bank later a new controller is designed if the right controller is not found in the bank.

4. CONCLUSION

In this preliminary note, we studied a fault tolerant control problem in the behavioral theoretic framework. We propose a real-time architecture to active FTC that does not require any knowledge of plant parameters. Thus, our

approach lies in the broad category of data-drive FTC that does not require an explicit FDI module, unlike to existing data-driven approach to FTC in the literature. Since we treat the FTC problem as any control system problem, which is subjected to change on the occurrence of fault. However, in data-driven control it is generally seen that the first step is to determine any representation (state-space parameters, Markovian parameters, finite impulse response) of the plant that implicitly introduce the exact data matching issues and FDI delays as well. Consequently, we do not carry any estimation process, thus, this approach seems promising to handle *in-exact* data-driven control to FTC.

Here, we propose our preliminary approach to real-time active FTC. Therefore, our future work is to study each shaded sub-system in Fig. 3 rigorously, and their integration issues as well.

REFERENCES

- M.N. Belur, H.L. Trentelman. Stabilization, Pole Placement, and Regular Implementability. *IEEE Trans. Automat. Control*, vol.47, no.5, pp. 735-744, 2002.
- M.Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki. Diagnosis and Fault-Tolerant Control. 2nd Edition, *Springer-Verlag*, Berlin, 2006.
- M.C. Campi, A. Lecchini, S.M Savaresi. Virtual Reference Feedback Tuning (VRFT): a new direct approach to the design of feedback controllers. *Proc. 39th Conf. on Decision and Control*, Sydney, pp. 623-628, 2000.
- Y. Fujisaka, Y. Duan, M. Ikeda System Representation and Optimal Tracking in Data Space *16th IFAC World Congress*, 2005.
- T. Jain, J. J. Yamé, D. Sauter A Real Time Router Fault Accommodation *Conference on Control and Fault Tolerant Systems, SysTol'10*, pp. 899-903, 2010
- O. Kaneko On Linear Canonical Controllers Within The Unfalsified Control Framework *17th IFAC World Congress*, Korea, pp. 12279-12284, 2008.
- A.S. Morse. Lecture Notes on Logically Switched Dynamical Systems. in P.Nistri and G.Stefani (Eds)- Non-linear and Optimal Control Theory, Lecture Notes in Mathematics-Springer, 2008.
- J.W.Polderman, J.C.Willems. Introduction to Mathematical Systems Theory: A Behavioral Approach. *New York:Springer-Verlag*, 1997.
- M.G. Safonov, T. Tsao. The unfalsified control concept and learning. *IEEE Trans. Automat. Control*, vol.42, no.6, pp. 843-847, 1997.
- M. Stefanovic, M.G. Safonov. Safe Adaptive Switching Control: Stability and Convergence. *IEEE Trans. Automat. Control*, vol.53, no.9, pp. 2012-2021, 2008.
- H.L. Trentelman, J.C. Willems. Control in Behavioral Setting. *35th IEEE CDC*, Japan, pp. 1824-1829, 1996.
- A.J. van der Schaft. Achievable behavior of general systems. *Systems & Control Letters*, vol.49, no.2, pp. 141-149, 2003.
- S. Weiland, A.A. Stoorvogel, B. de Jager A behavioral approach to the \mathcal{H}_∞ optimal control problem *Systems and Control Letters*, vol. 32, no.5, pp. 323-334, 1997.
- J.J. Yamé, D. Sauter. A Real-Time Model-Free Reconfiguration Mechanism for Fault-Tolerance. *10th Intl. Conf. on Control, Automation, Robotics and Vision*, pp. 91-96, 2008.