Ecole doctorale IAEM Lorraine
Faculté des Sciences et Technologies

THÈSE

pour l'obtention du

Doctorat de l'Université de Lorraine, France
Spécialité : Automatique, Traitement du Signal et Genie Informatique

présentée par

Tushar JAIN

Contribution à la synthèse de commandes tolérantes aux défauts par l'approche comportementale

"Behavioral system-theoretic approach to fault-tolerant control"

Soutenue publiquement le 20 Novembre 2012

Membres du jury :

Rapporteurs:

Prof. Vincent COCQUEMPOT	-	Université Lille1 : Sciences et Technologies, France
Prof. Ron PATTON	-	The University of Hull, UK

Examinateurs:

Prof. Abdellah EL MOUDNI	-	Université de Technologie de Belfort-Montbéliard, France
Prof. Didier MAQUIN	-	Université de Lorraine, France
Prof. Dominique SAUTER	-	Université de Lorraine, France (Directeur de thèse)
Dr. Joseph Julien YAMÉ	-	Université de Lorraine, France (Co-directeur de thèse)

Centre de Recherche en Automatique de Nancy UMR - 7039 - CNRS - UDL
Boulevard des Aiguillettes, 54506 Vandœuvre-Les-Nancy
Tel. +33(0)3 83 68 44 19 - Fax +33(0)3 83 68 44 37

To my parents,
To Akshita

Acknowledgment

I sincerely and wholeheartedly acknowledge contribution of lots of people covertly, overtly, or in many ways they could not even imagine, to this work. First and foremost, I would like to express my gratitude to my supervisors Prof. Dominique Sauter and Dr. Joseph Julien Yamé for giving me an opportunity to join their research group SURFDIAG. They were a constant source of inspiration and support in the scientific endeavors. To make it short, I would thank Dominique for showing his constant confidence in what I was doing and Joseph for teaching me a lot about *Research* and *Science*. I benefited much from the PAPYRUS European 7th framework research project during the last phase of my stay in CRAN, thanks to Dominique for making this possible. Special thanks to Joseph for polishing my knowledge with many informal discussions and brainstorming sessions that have taught me how to do research. His engineering intuition and mathematical rigorousness have been an outstanding example to me. I was in awe and appreciated his ingenious way of seeing a mathematical problem with some satirical examples like

"shooting the mosquitoes with a gun"

In the research that I conducted during these three years, I greatly profited from numerous interactions with several other researchers. Some of them are Prof. Jan Willems to whom I met during the graduate school in Paris, Prof. Jin Jiang, Prof. Youmin Zhang, and Prof. Sirkka-Lissa Jämsä-Jounela during their visit to CRAN in 2010, 2011, and 2012 respectively.

Additionally, I want to thank all the members of my Ph.D. evaluation committee: Prof. Vincent Cocquempot, Prof. Abdellah El Moudni, Prof. Didier Maquin, Prof. Ron J Patton. Special thanks to Vincent for the very thorough review of the draft version of my thesis, for inviting me to the GTS3 meeting in Paris, for providing me with many constructive comments, and for giving interesting interpretations on some point of the thesis. In addition, thanks to Ron for the discussions on fault-tolerant systems, which took place at the dining table during ROCOND'12, SysTol'10. Thanks to Didier for his constant interaction during SysTol'10 and Journée des doctorants.

In CRAN, I would like to thank all the people I have met, which made my stay here interesting and enjoyable. Without giving an exhaustive list, some of them are Junbo, Adriana, Gabriela, Manal, Sinuhe, Amine, Jingwen, Ronay, Lukasz, Shaik. I also would like to thank Abdou who helped me with French related works in my initial years and Ahmed, Ghassane during my last years. Nonetheless, I would like to thank Sabine for her constant and quick help with the administration.

Finally, coming to life in Nancy, I rarely felt I was out of India because of the Indian circle I was surrounded with. It is my pleasure to thank the people who made me feel home out of home. To start with, I thank Ashish and Neeraj for making the transition from New Delhi to Nancy easy for me. I thank to my housemate Ritesh for never concluding debates starting from preparing the Indian food to Indian politics. I thank Shashi, Shaik, and Mehwish for all the memorable evening we spent together by playing some board games, watching movies and long cooking adventures. There were many others, who were not mentioned here, yet helped me in making my stay in Nancy memorable all along. I would like to express my deep gratitude to all of them.

Contents

Acknowledgment i

List of Figures ix

Nomenclature xi

Abstract xii

I Motivations and Objectives 1

1 Fault-Tolerant Control Systems 3

1.1 Need of Fault Tolerant Control Systems . . . 4
1.2 Classification of Faults . . . 5
1.3 Faults Modeling . . . 7
1.3.1 Multiplicative faults . . . 7
1.3.2 Component faults . . . 8
1.3.3 Additive faults . . . 9
1.4 Fault-tolerant Control System . . . 10
1.4.1 Classification of FTC systems . . . 12
1.4.2 Role of FDD in Active Fault-Tolerant Control . . . 14
1.5 Fault-tolerant Control Methodology . . . 16
1.5.1 Passive Methods . . . 17
1.5.2 Active Methods . . . 18
1.5.3 Analysis of Fault-tolerant systems . . . 26
1.6 Scope of the thesis . . . 28
1.7 Outline of the thesis . . . 32
1.8 Organization of the thesis . . . 33

2 Behavioral Paradigm 35

2.1 Dynamical systems . . . 35
2.2 Linear Differential systems . . . 39
2.3 Equivalent and minimal representations . . . 41
2.4 Latent variables and elimination . . . 43
2.5 Observability and detectability . . . 46
2.6 Controllability and stabilizability . . . 48
2.7 Autonomous behavior . . . 51
2.8 Input/Output Representation . . . 53

2.9 Input/State/Output Representation 56

II Novel active fault-tolerant control approaches 59

3 Interconnection of Behaviors 61
3.1 Interconnection of dynamical systems 61
3.2 Regular Interconnection 63
3.3 Implementability 66
3.3.1 Full interconnection 67
3.3.2 Partial interconnection case 68

4 Projection based approach to FTC in behavioral context 73
4.1 Introduction 74
4.2 Fault-Tolerant Control in the behavioral framework 75
4.2.1 Feedback Interconnection 75
4.2.2 Problem Formulation 77
4.3 Design and Implementation of Real-time FTC 80
4.3.1 The Behaviors 80
4.3.2 Set-up of the FTC Architecture 81

5 Online redesign approach to FTC in behavioral context 87
5.1 Fault-Tolerant Control: partial interconnection case 88
5.1.1 Feedback Interconnection within the partial interconnection 88
5.1.2 Problem Formulation 90
5.1.3 Design and Implementation of Real-time FTC via partial interconnection 91
5.2 Fault-tolerant Control: full interconnection case 95
5.2.1 Feedback Interconnection within the full interconnection 95
5.2.2 Design and Implementation of Real-time FTC via full interconnection 97

III Practical Aspects 101

6 Real-time Smooth Interconnection in the behavioral context 103
6.1 Limitations in AFTC systems 103
6.2 Basic Cause of Switching Transients 104
6.3 Guaranteeing the Smooth Interconnection 108

7 Simulation Results and Discussion 113
7.1 Aircraft Auto landing mechanism . . . 113
7.1.1 Model Description . . . 114
7.1.2 Fault scenarios . . . 116
7.1.3 Constructing a controller bank . . . 116
7.1.4 Projection-based FTC mechanism . . . 117
7.2 Hydraulic Plant . . . 119
7.2.1 Model of the plant . . . 119
7.2.2 Fault scenario . . . 121
7.2.3 Online redesign based FTC mechanism . . . 121
7.3 Wind Turbine . . . 124
7.3.1 Description of Wind Turbine . . . 125
7.3.2 Fault Scenario . . . 127
7.3.3 Fault-Tolerant Control Objectives . . . 128
7.3.4 Projection-based FTC . . . 130
7.3.5 Online-redesign based FTC . . . 134

8 Conclusions and future research 139

A Dwell-Time Switching 143

B Direct Continuous-time Model Identification 145
B.1 The Traditional SVF Method . . . 145

Bibliography 147

List of Figures

1.1 Fault Classification according to their location 6

1.2 According to their time characteristics. (a) abrupt; (b) incipient; (c) intermittent 7

1.3 Fault Tolerant Control system architecture with the supervision sub-system [Pat97]. 10

1.4 Decomposition of Fault Tolerant Control 14

1.5 Visualization of time-delay in FDD process 16

1.6 Control Switching approach to FTC 24

1.7 Visualization of the time-delay in FTC systems 31

2.1 Permitted trajectories defining the behavior 37

2.2 An illustration of concatenation of trajectories. 39

2.3 Projection of full behavior 44

2.4 Observability 47

2.5 Controllability 50

3.1 Interconnection of systems 62

3.2 Feedback interconnection 66

3.3 Plant 67

3.4 Controller 68

3.5 The plant and controller after interconnection 69

3.6 The relation between $\mathcal{P}$, $(\mathcal{P})_w$, $\mathcal{C}$, and $\mathcal{K}$. 70

3.7 The hidden behavior 71

4.1 Feedback Configuration in the classical sense 76

4.2 Feedback Configuration in the behavioral sense 76

4.3 Output trajectories in Case-1 79

4.4 Output trajectories in Case-2 79

4.5 Projection-based Active FTC in behavioral context 82

4.6 Internal structure of the supervisor 82

5.1 Feedback Configuration in the classical sense 88

5.2 Feedback Configuration in the behavioral sense 89

5.3 The canonical controller 92

5.4 2DOF feedback control in the classical sense 96

5.5 2DOF feedback control in the behavioral sense 96

6.1 Switching from controller 1 to controller 2: controller reconfiguration 105

6.2 Example : Non-smooth Interconnection 109
6.3 Example : Smooth Interconnection 110

7.1 Aircraft during the landing phase 114
7.2 System-level Simulink Architecture of Projection-based Fault-Tolerant Aircraft Control 118
7.3 Closed-loop signals aircraft autopilot landing system 118
7.4 The two-tanks plant 120
7.5 Closed loop signals of two-tank FTC system 123
7.6 System Overview of the Wind-Turbine model 124
7.7 Illustration of the reference power curve for the wind turbine depending on the wind speed 129
7.8 Block diagram showing major elements of the Simulink-based Wind Turbine FTC Systems 129
7.9 Nacelle Yaw Error 130
7.10 Wind profile 132
7.11 System-level Simulink Architecture of Projection-based Fault-Tolerant Wind Turbine Control 132
7.12 Closed-loop Signals of Fault-Tolerant Wind Turbine System: Dashed-dotted line denotes the case of without any FTC mechanism; Dashed line denotes the second faulty case; Solid line denotes the case of using the proposed FTC mechanism 133
7.13 Wind profile used for simulation 135
7.14 System-level Simulink Architecture of Online-redesign based Fault-Tolerant Wind Turbine Control 136
7.15 Closed-loop Signals of Fault-Tolerant Wind Turbine System . 136

Nomenclature

$\mathbb{R}$	field of real numbers
$\mathbb{R}^{\mathtt{n}}$	$\mathtt{n}$-dimensional real Euclidean vector space over $\mathbb{R}$
$\mathbb{R}^{\bullet\times\bullet}[\xi]$	set of polynomial matrices with real coefficients having appropriate dimension and indeterminate ξ
$\mathbb{T}$	time space
$\mathbb{S}$	signal space
$t \in \mathbb{R}$	time
$s(t) \in (\mathbb{T} \times \mathbb{S})$	system trajectories
$\mathfrak{L}^{\mathtt{s}}$	set of all linear differential systems with $\mathtt{s}$ variables
$x(t) \in \mathbb{R}^{\mathtt{n}}$	state vector
$y(t) \in \mathbb{R}^{\mathtt{m}}$	output vector
$u(t) \in \mathbb{R}^{\mathtt{r}}$	input vector
$f(t) \in \mathbb{R}^{\mathtt{s}},\ (\mathtt{s} \leq \mathtt{m})$	fault vector
$e(t) \in \mathbb{R}^{\mathtt{n}}$	tracking error

Abbreviations

FDD, FD	Fault Detection and Diagnosis
FDI	Fault Detection and Isolation, Fault Detection and Identification
(A or P)FTC(S)	(Active or Passive) Fault-Tolerant Control (Systems)
CR	Controller Reconfiguration
FA	Fault Accommodation
LTI	Linear Time Invariant
MIMO	Multi-Input Multi-Output
SISO	Single-Output Single-Input
FMEA	Failure-Modes and Effects Analysis

Abstract

The field of system and control theory has achieved an interdisciplinary status during the past five decades, and we refer to the theory that was developed during this period as the conventional control theory. This mainly relates to the study of automation and the design of controllers. A controller is a device that makes the interconnection with a given system so that the controlled system can behave in a desired way. In this thesis, we deal with the issues when the controlled system becomes faulty. The control of a faulty system addresses the concept of Fault-Tolerant Control System (FTCS). The study of such systems is in response to the demands of large-scale industries since from their viewpoint it is the foremost task to design control systems, which are capable of tolerating potential faults occurring either in the internal closed-loops or from the environmental factors in order to improve the reliability and availability of a system while providing the expected performance.

The work presented in this thesis is mainly focused on synthesizing the online controllers that guarantee the closed-loop system to be fault-tolerant at anytime. Two methodologies are proposed in this work, which rest under the broad classification of FTC systems, namely projection-based approach and online redesign approach. The novelty of these approaches lies in the fact that any *a priori* information about the plant is not available in real-time. In addition, no online identification or estimation of the operating plant is carried out. Rather, the re-configuration procedure of the controllers is solely based on the measurements generated by the unknown plant. This phenomenon is very nicely demonstrated by using the time-trajectory based viewpoint of behavioral theory. Within this mathematical framework, the interconnection between two dynamical systems, namely the plant and the controller, plays the significant role. Consequently, taking the benefits of this behavioral framework, the real-time measurement based solutions are proposed to handle the fault-tolerant control problem.

From the practical implementation viewpoint, the transient management during the controller reconfiguration mechanism is one of the important requirements for active FTCS. The last part of the thesis deals with the online implementation of the controllers within the behavioral framework, which takes care of the transient mechanism. The proposed approach guarantees the "real-time smooth interconnection" between the controller and the unknown plant. Moreover, in this part the application of the theory developed in the thesis is effectively demonstrated on real-world examples, namely the two-tanks system, the aircraft landing mechanism, and the NREL's 5MW wind turbine system.

Part I

Motivations and Objectives

Chapter 1

Fault-Tolerant Control Systems

Contents

1.1 Need of Fault Tolerant Control Systems 4

1.2 Classification of Faults 5

1.3 Faults Modeling 7

1.3.1 Multiplicative faults 7

1.3.2 Component faults 8

1.3.3 Additive faults 9

1.4 Fault-tolerant Control System 10

1.4.1 Classification of FTC systems 12

1.4.2 Role of FDD in Active Fault-Tolerant Control 14

1.5 Fault-tolerant Control Methodology 16

1.5.1 Passive Methods 17

1.5.2 Active Methods 18

1.5.3 Analysis of Fault-tolerant systems 26

1.6 Scope of the thesis 28

1.7 Outline of the thesis 32

1.8 Organization of the thesis 33

Modern technological systems rely on sophisticated control schemes to meet increased performance and safety requirements. These conventionally designed control schemes that are embedded within such complex systems may result in an unsatisfactory performance, or even instability, in the event of malfunctions occurring within the system components. These malfunctions are termed as *faults*. The International Federation of Automatic Control (IFAC) SAFEPROCESS Technical Committee defines a fault as an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition [Ise97]. Therefore, in order to improve the reliability and availability of a system while continually providing a satisfactory performance, it is necessary to design control systems, which are capable of tolerating potential faults. These types of control systems are known as Fault-Tolerant Control (FTC) systems. In fact, fault tolerance is the property of a system reacting against the occurring faults.

1.1 Need of Fault Tolerant Control Systems

The need of fault tolerant control systems varies depending on the type of applications. Generally, systems are classified as non-safety-critical systems, and safety-critical systems. In non-safety-critical systems, it would have sufficed to notify the user that something went wrong with the system which needs attention. Some of the examples are washing machine, coffee machine, etc. As a matter of fact, every system is a compound unit of various sub-systems. These sub-systems are interconnected with each other in some way that whenever a fault occurs in one of the sub-systems, it progressively prevails to other sub-systems as well. Consequently, it forms a chain of faults. Nevertheless, in case a fault is accommodated on the very first stage, then further damages to the system could be avoided. On the other side, in safety-critical systems, the user might not have an option to shut down the system, thereby, to forbid a disaster necessary actions are required at run-time guaranteeing the *fail-safe operation*, since it involves the loss of capital and primarily, the human life. Industrial plants, unmanned aerial vehicles, etc. are some of the examples. Several incidents had occurred that further motivates the system to equip with an effective fault tolerant control strategy, which are as follows:

- On October 4, 1992, the fatal crash of EL AL Flight 1862 of a Boeing 747-200F freighter [ZJ08] : The crew flying out of Schiphol Airport in Amsterdam suffered separation of both engines from the starboard wing. The report states that despite the failure, it continued flying for almost 15 minutes, thus, finally crashed into an apartment building, which caused a considerable loss of life.

- On December 2, 1984, the world's worst industrial catastrophe at the Union Carbide India Limited (UCIL) pesticide plant in Bhopal, Madhya Pradesh, India termed as Bhopal Disaster : A leak of methyl isocyanate gas and other chemicals from the plant resulted in the exposure to hundreds of thousands of people [sit]. This leakage leads to a disaster because the steam boiler, intended to clean the pipes, was out of the action for unknown reasons.

In the first incident above, later studies expressed that the reported available time before the crash was sufficient to design an online "fault correcting" controller. While during the investigation of the second incident, the company admits that the safety system in place was not programmed effectively for handling that particular type of fault. Numerous examples do exist in the history that howl the need of fault-tolerant control systems such as the explosion at the nuclear power plant in Chernobyl, Ukraine on 26 April 1986, the explosion of the Ariane 5 rocket on 4 June 1996, the crash of the American Airlines Flight 191 on 25 May 1979 where the pilot had only 15 seconds to react before the flight crashed, etc. Indeed, the above disasters could have been avoided by means of an effective design of a fault-tolerant control system. Unlike in these unfortunate cases, there are some fortunate cases as well where the tragedy was called-off by taking necessary correctable actions at run-time. Like the case of Delta Flight 1080 (April 12, 1977), the elevator gets jammed at 19° upwards and the pilot hadn't been provided an indication about this malfunction. Fortuitously, the pilot successfully reconfigured the remaining control elements within the aircraft and landed it safely. A theoretical study on the fatal crash of Boeing 747-200F freighter discussed in [MJ03] provides a very strong evidence for the need of a fault-tolerant control in the scientific community. It shows that the available time after the occurrence of fault was appropriate enough to avoid the crash.

Seeing to the above incidents, fault-tolerant control systems drew more attention from a wide range of industrial and academic communities, where the safety of human lives and reliability issues are of utmost priority. Note that fault tolerance is not limited to high-end systems, and consumer products, such as automobiles, etc. Since these products are increasingly dependent on microelectronic/mechatronic systems, on-board communication networks, and software, thus requiring new techniques for achieving tolerant to faults [ZJ08]. Plainly, the purpose of research on FTC systems is to develop generic methods for achieving an increased fault-tolerance by means of synthesizing corrective actions for an operating faulty system.

1.2 Classification of Faults

In the existing FTC community, faults are often classified according to their location of occurrence within the system as (see Fig. 1.1)

Actuator faults represent the partial or the complete loss of control actions. A "stuck" in an actuator is one of the examples of a complete loss of the control action where it is unable to produce any actuation to the system. However, partially failed actuator produces only a part of the normal actuation, e.g. hydraulic leakage.

Sensor faults represent the incorrect reading from the sensors. These faults

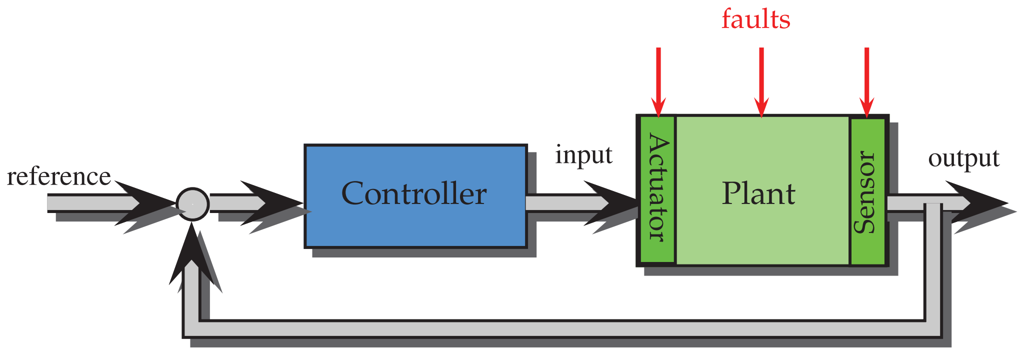


Figure 1.1: Fault Classification according to their location

can also be subdivided into partial and complete faults. Broken wires, losses of contact with the surface, etc. are some of the examples for a complete loss while fractional sensor faults include gain reduction, biased measurements, etc.

Component faults represent all other faults that cannot be characterized into either the sensor faults or the actuator faults. These faults could cause a change in the physical parameters of the system due to a structural damage. The component faults cover a wide class of situations. Therefore, treatment of this class is considered to be the most challenging one.

The set of actuator and sensor as shown in Fig. 1.1 collectively form a "to-be-controlled" system. A trivial solution to handle the aftereffects of an occurring fault within any of these sub-systems is to replace the faulty components by alternative components. However, *duplicating the actuators or sensors* in the system in order to achieve increased fault-tolerance is often not considered a preferred option due to the high running cost and large sizes.

Faults can also be categorized in accordance with their time characteristics as illustrated in Fig. 1.2. A fault is called

hard (or abrupt) if its effects on the system are larger and bring the system very close to the limit of acceptable performance.

intermittent if it appears and disappears repeatedly, where it is difficult to determine whether it is a fault or a disturbance.

soft (or incipient) if its impact on the system is of small magnitude and its effects in the beginning are unnoticeable.

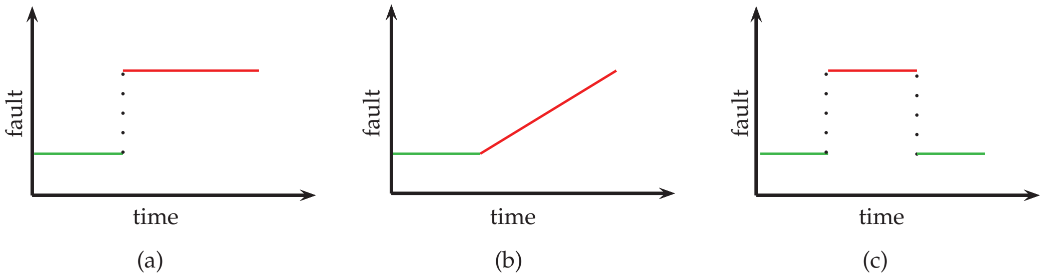


Figure 1.2: According to their time characteristics. (a) abrupt; (b) incipient; (c) intermittent

The above classification depends entirely on the way a fault is modeled. The next section presents the modeling of faults. This, in fact, demonstrates how an occurred fault affects the system.

1.3 Faults Modeling

Faults are often categorized into additive and multiplicative faults according to their way of representation. We use the *state-space representation* for describing a dynamical system, denoted by Σ, so that the relation from the system inputs $u \in \mathbb{R}^{\mathtt{m}}$ to the measured outputs $y \in \mathbb{R}^{\mathtt{p}}$ can be expressed by

$$\Sigma_{nom} : \begin{cases} \dfrac{d}{dt}x(t) = Ax(t) + Bu(t), \\ \quad y(t) = Cx(t) + Du(t) \end{cases} \tag{1.1}$$

where $x \in \mathbb{R}^{\mathtt{n}}$ denotes the state of the system and Σ_{nom} denotes the nominal (nom) $(or$ fault-free) system. In addition, all the matrices have appropriate dimensions whenever unspecified.

1.3.1 Multiplicative faults

Multiplicative $(mult)$ modeling of faults is mostly used to represent actuator faults (af), and sensor faults (sf). Particularly, actuator faults are modeled as an abrupt change in the nominal control action described by

$$u^f(t) = \Gamma_A u(t), \tag{1.2}$$

where $\Gamma_A = \text{diag}(\gamma_1^a, \gamma_2^a, ..., \gamma_m^a) \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}, \gamma_i^a \in \mathbb{R}$. Substituting the nominal control action $u(t)$ in equation (1.1) by a new control action $u(t)^f$ for the faulty (f) operating mode results in the following state-space model

$$\Sigma_{mult,af} : \begin{cases} \dfrac{d}{dt}x(t) = Ax(t) + B\Gamma_A u(t), \\ \quad y(t) = Cx(t) + D\Gamma_A u(t) \end{cases} \tag{1.3}$$

In this way $\gamma_i^a = 0, \gamma_i^a = 1$, and$\gamma_i^a = \varepsilon^a \forall i \in \{1, 2, \ldots, m\}$, with $\varepsilon^a \in (0, 1)$, represents a complete failure, nominal operation, and partial loss respectively of the i-th actuator within the system.

Sensor faults (sf) appearing in the system (1.1) represent incorrect reading from the sensors. As a result, the real output of the system differs from the variable being measured. Similar to the above, they can be modeled as

$$y^f(t) = \Gamma_S y(t) \tag{1.4}$$

where $\Gamma_S = \mathrm{diag}(\gamma_1^s, \gamma_2^s, ..., \gamma_p^s) \in \mathbb{R}^{\mathtt{p}\times\mathtt{p}}, \gamma_j^s \in \mathbb{R}$, so that $\gamma_j^s = 0$ represents a total fault of the j-th sensor, and $\gamma_j^s = 1$ models the normal mode of operation of the j-th sensor. In addition, partial faults are modeled by taking $\gamma_j^s \in (0, 1)$. The model of the system after the appearance of sensor faults is then represented by

$$\Sigma_{mult,sf} : \begin{cases} \dfrac{d}{dt}x(t) = Ax(t) + Bu(t), \\ \quad y^f(t) = \Gamma_S Cx(t) + \Gamma_S Du(t) \end{cases} \tag{1.5}$$

The state-space models (1.3)-(1.5) have been widely used in the literature on FTC systems (see, e.g. [TCJ02], [NSHT00]).

1.3.2 Component faults

Component ($comp$) faults bring changes in any of the elements within a dynamical system. It belongs to the class of faults that cannot be classified as a sensor or an actuator fault. These faults are often modeled in the form of a linear parameter-varying system

$$\Sigma_{comp} : \begin{cases} \dfrac{d}{dt}x(t) = A(f)x(t) + B(f)u(t), \\ \quad y(t) = C(f)x(t) + D(f)u(t) \end{cases} \tag{1.6}$$

where $f \in \mathbb{R}^{\mathtt{n_f}}$ is a time dependent exogenous parameter vector representing the component faults. Suppose there is only one component fault in a dynamical system Σ_{nom}. This transforms the B-matrix in (1.1) to $B(f)$ and has an explicit representation $B(f) = \Gamma_{bf}B$. Assuming matrix-B to be full column rank and there exists a diagonal matrix $\tilde{\Gamma}_{bf}$ such that

$$\Gamma_{bf}B = B\tilde{\Gamma}_{bf} \Longrightarrow \tilde{\Gamma}_{bf} = B^\dagger \Gamma_{bf} B$$

where $B^{\dagger}$ is the pseudo-inverse of matrix-B. The state-space representation of the faulty model is then given by

$$\Sigma_{comp} : \begin{cases} \dfrac{d}{dt}x(t) = Ax(t) + B\tilde{\Gamma}_{bf}u(t), \\ \quad y(t) = Cx(t) \end{cases} \tag{1.7}$$

We see that equation (1.7) is equivalent to (1.3) with $D = 0$. From the above, it is interesting to note that the component faults can be regarded as a special case of multiplicative additive faults. Similarly, this can be expressed for multiplicative sensor faults as well.

1.3.3 Additive faults

The representation of additive (*add*) faults are more general than representing the multiplicative faults. The state-space model with additive faults is given by

$$\Sigma_{add} = \begin{cases} \dfrac{d}{dt}x(t) = Ax(t) + Bu(t) + Ff(t), \\ \quad y(t) = Cx(t) + Du(t) + Ef(t) \end{cases} \tag{1.8}$$

where $f(t) \in \mathbb{R}^{\mathrm{n_f}}$ is a signal describing the fault. This representation may, in principle, be used to model a wide class of faults, including sensor, actuator, and component faults. For instance, for the case of actuator faults, posing (1.2) as

$$u^{f}(t) = (\Gamma_A - I)u(t) + u(t)$$

results in (1.3), which is equivalent to (1.8) with $\begin{bmatrix} F \\ E \end{bmatrix} f(t) = \begin{bmatrix} B \\ D \end{bmatrix} (\Gamma_A - I)u(t)$. This can also be expressed for multiplicative sensor faults in a similar way. To show the component fault as a special case of additive fault, consider a component fault in the matrix-B which is represented as $B(f) = B \pm \Delta B_f$, then $Ff(t) := \pm(\Delta B_f)u(t)$.

One of the advantages of such modeling, as already mentioned, that the additive representation can be used to model a more general class of faults than the multiplicative one. In addition, it is more suitable for investigating the design of FTC schemes because here the faults are represented by intruding a signal rather than by "a change" in the state-space matrices of the system. For this reason, the majority of the fault diagnosis methods is focused on studying the additive faults ([Him78], [Ise84], [TNS98]).

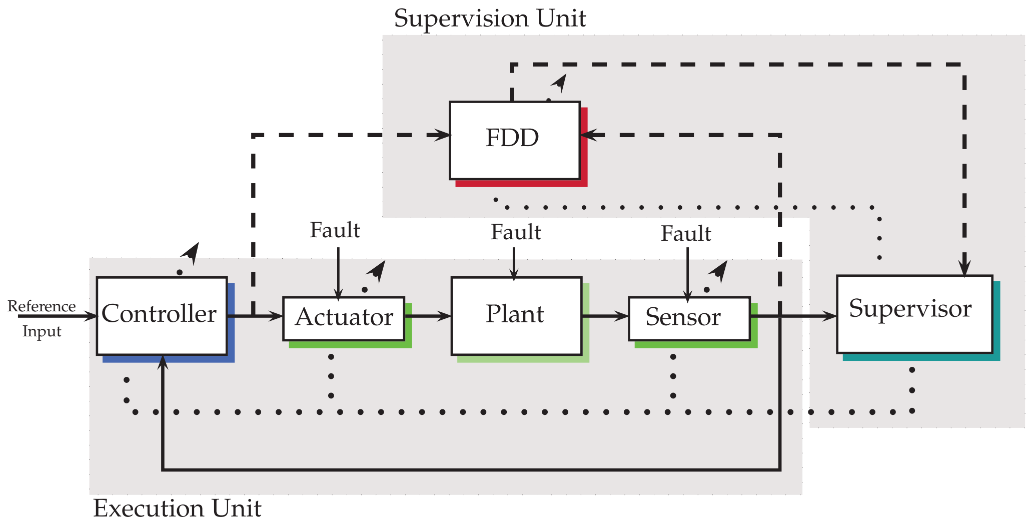


Figure 1.3: Fault Tolerant Control system architecture with the supervision sub-system [Pat97].

1.4 Fault-tolerant Control System

A system is said to be *fault tolerant* whenever under a fault occurrence, the system is able to recover its original task with either the same or a degraded performance. Generally, a fault-tolerant system is composed of two cascaded modules:

Fault Detection and Diagnosis (FDD): is a monitoring module which is used to detect faults, and diagnose their location and significance in a system. Stating precisely, it performs the following tasks:

- fault detection : to indicate whether a fault has occurred or not within the system.
- fault isolation : to determine the location of an occurring fault.
- fault identification : to precisely estimate the size and the nature of a fault.

Supervisor: is a recovery module taking necessary actions so that the faulty system (i.e. the system under a fault occurrence) can achieve the control objectives *at anytime*. These actions can demand to reconfigure the set of actuators, sensors, or the control law. Generally, in the literature, this module is also termed as the Fault Accommodation (FA) module or the Controller Reconfiguration (CR) module.

The general functional scheme of a fault-tolerant control system is shown in Fig.1.3 with four main components: the plant itself (including sensors and actuators), the fault detection and diagnosis unit, the feedback (or feed-forward) controller, and the supervisor sub-system. The main controller activity occurs in the execution unit, and is represented by the solid line, while the dashed line represents the operation of the FDD unit with a dotted line representing the adaptation (tuning, scheduling, accommodation, and reconfiguration). In the illustrated figure, the plant is considered to have potential faults in the sensors, actuators (or other system components). In a faultless case, the FDD unit remains idle. During this course of time, the nominal feedback controller attenuates the disturbances, and ensures the set-point following with satisfying other requirements on the closed-loop system. The job of the FDD unit is to extract the complete information about the onset, location and severity of occurring faults. It is believed that if a fault is tolerable, it must be diagnosed as early as possible since it might lead to serious consequences with the evolving time. On the supervisory level, the diagnosis block simply recognizes that the closed-loop is faultless and no change in (or of) the control law is necessary. On the other hand, whenever a fault occurs, the supervision subsystem together with a precise information received from the diagnosis block makes the closed-loop fault-tolerant. This procedure involves, based on the system inputs and outputs together with the precise information, reconfiguring the sensor set and/or actuators to isolate the faults, and tune or adapt the controller to accommodate the fault effects. This makes the closed loop satisfying the performance specifications at anytime [JYS10a]. In the literature, the terms, namely fault detection and isolation (FDI) or fault detection and identification (again, FDI) are often used. To avoid any confusion, here FDI has been adopted to stand for fault detection and isolation, while FDD will be used when the fault identification function is also added to FDI.

A first look at the structure of the FTC system gives it an impression of adaptive control systems. This can quickly be seen by replacing the diagnosis block by a system identification block identifying the plant parameters, and replacing the supervisor block by a "controller parameter adjustment" block converts an FTC system to an adaptive control system. However, the operating capacity of an FTC system is far ahead than an adaptive system. Unlike to the former system, the latter system is primarily designed to adapt to changes in the process dynamics over a specific operating range and disturbance characteristics [AW95].

In [BKSL03], it is shown that the dependability analysis of a system determines the effectiveness of a fault-tolerant control system. Dependability measures the degree to which a system is operable at a random time during a specific mission profile, given that its services are available at the start of the mission [AKKH09]. It is not a single property measure, but a collection of related measures, including some attributes such as:

Safety : The property that a system does not fail in a manner that causes catastrophic damage during a specified period of operation.

Maintainability : The ease with which a system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment.

Reliability The probability that a system will perform the functions for which it is designed for a given period of operation in the nominal conditions. Note that a fault-tolerant control system cannot change the reliability of the plant components, but it improves the reliability of an overall system [BKSL03].

A dependable system with high availability and reliability is considered as a *fail-safe system*. In fact, the FTC is a control methodology that ensures continual safe or acceptable operation of the system through fault detection and diagnosis (FDD), and controller reconfiguration (CR) in response to occurring faults. See [Wu04a] for a deeper insight of the dependability analysis. Over the last three decades, the growing demand for safety, reliability, maintainability, survivability in technical systems have drawn a significant research within fault diagnosis, like, in [Him78] where fault detection for chemical processes is introduced. One of the first surveys of fault detection is conducted in [Ise84], where some methods based on modeling, and estimations are introduced and in the latest survey [HKKS10], various reconfiguration methods are discussed in parallel to fault diagnosis techniques. Historically, from the point of view of practical applications, a significant amount of research on fault-tolerant control systems was motivated by aircraft flight control system designs [Ste05]. The goal, therein, was to provide “self-repairing” capability in order to ensure a safe landing in the event of severe faults in the aircraft [EWLW85]. Other practical applications, where the research on designing an FTC system has been conducted, are rail traction drive [BPD99], ship propulsion plant [BIZL98], winding machine [NSHT00], automated highway systems [SP97], etc.

1.4.1 Classification of FTC systems

To a certain extent, fault tolerance can also be accomplished without the structure given in Fig. 1.3 by means of well established control methods. Generally, FTC systems are classified in two categories: *passive* (PFTC) and *active* (*A*FTC). Figure 1.4 shows a taxonomy of fault-tolerant control methods, based on either passive or active approaches. In PFTC systems, controller parameters remain unchanged and this fixed controller has the ability to tolerate the changes in the plant dynamics. The closed-loop system satisfies its goals under a very restrictive repertoire of likely occurring faults, thus, maintain the *anytime fail-safe* property. This approach neither requires any FDD schemes nor a controller re-configuration strategy, thereby, makes it a computationally more attractive approach [EWLW85]. The passive approaches make use of robust control techniques without the use of an on-line FDD module to ensure that the closed-loop system remains insensitive to certain faults. Thus, the impaired system continues to operate with the same controller. In order to achieve such a robustness against faults, usually a very restricted subset of possible occurring faults is considered. Moreover, it is popularly known that to achieve the robustness against certain faults using a single controller is only possible at the expense of decreased nominal performance [Pat93]. Since the control law is not reconfigured, and the control objectives associated with the system are fulfilled with a low level of performance, this approach is also known as the *conservative approach* [Pat97].

In contrast to PFTC systems, the AFTC systems are *adaptive* in nature. The controller parameters can change according to the changes in the plant dynamics. This maintains the fail-safe operation, thereby, guarantees a satisfactory performance, not only when all control components are functioning well, but also in the cases when there are malfunctions of the sensors, actuators, or other system components. The reconfigured controller compensates against the impacts of faults either by *selecting* a pre-computed control law among the available control laws or by synthesizing a new *on-line* one. Both of these algorithms rely heavily on *real-time* FDD schemes that provide the most up-to-date information about the current working status of the plant. The structure of an active FDD-based FTC system is illustrated by Fig.1.3. The FDD module in the supervision unit uses the input-output measurements of the system to detect and localize the faults. Subsequently, the information about the estimated faults is passed to the supervisor subsystem together with input-output measurements. This led to the changes in the parameters and/or the structure of the controller to achieve the acceptable post-fault system performance. In other words, all subsystems in an AFTC system should be operating in an on-line and real-time manner. In this regard, “AFTC systems are real-time systems”. Indeed, to achieve a successful control reconfiguration, the FDD scheme should be able to provide an accurate and the most up-to-date information (including post-fault system models) about the system in real-time.

Depending on the way the post-fault performance is achieved, active FTC

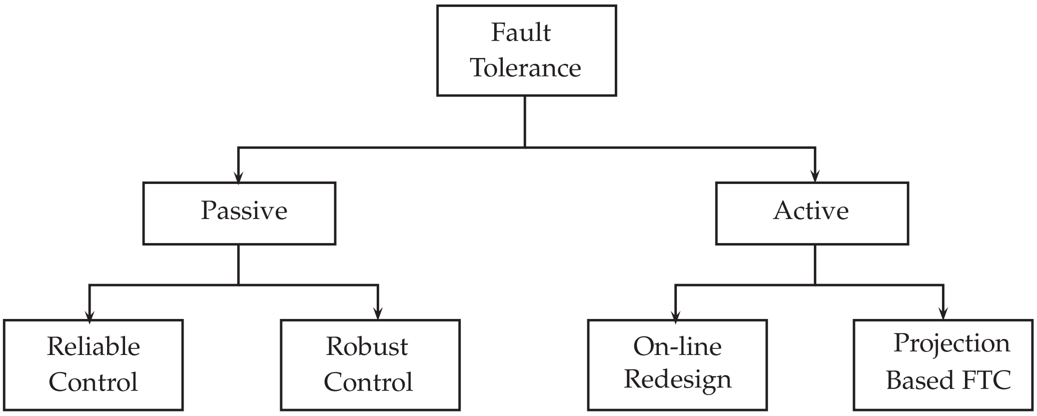


Figure 1.4: Decomposition of Fault Tolerant Control

methods are further subdivided into the projection-based methods [MHRC89] and the on-line redesign methods [LWEB85]. These methods take into account the post-faulty controller switching. The online redesign method involves determining new controller parameters in response to the control impairment. This is often referred to as the reconfigurable control and/or restructurable control. On the contrary, the projection based methods rely on the controller selection from a set of off-line pre-designed controllers. Usually each precomputed controller in the set is designed for a particular class of likely-occur faulty situations. A controller is switched in the closed-loop with the help of a supervision subsystem whenever the corresponding fault pattern has been diagnosed by the FDD scheme.

Other than the conceptual differences amongst AFTC and PFTC systems, the structural differences, i.e. inclusion of both, FDD and reconfigurable controllers within an overall system structure is the main criteria for distinguishing AFTC systems from PFTC systems.

1.4.2 Role of FDD in Active Fault-Tolerant Control

In the process and manufacturing industries, the FDI stage is crucial to improve the production efficiency, quality of the product, and the cost of production. At many times, fault detection, isolation and estimation collectively are *termed as* FDD or *simply* FD (Fault Diagnosis). The FDI algorithm primarily consists of making a binary decision, either that something has gone wrong or that everything is fine with the system. The residual signals play an important role in making these binary decisions, which are the signals that, in the absence of faults, deviate from zero only due to modeling uncertainties, with nominal value being zero, or close to zero under actual working conditions. In this regard, the role of the decision system is to determine whether the residuals differ significantly from zero and, from the pattern of zero and non-zero residuals, to decide which are the most likely fault effects. Subsequently, it determines the location as well as the nature of the fault. There are two main directions for developing a fault diagnosis system: using *hardware redundancy*, and using *analytical redundancy*. The former direction is based on the concept of comparing the duplicate signals generated by different hardwares, such as measurements of the same signal given by two or more sensors. The drawback of this approach is that a significant cost is involved while using necessary extra equipments. On the other hand, the analytical redundancy uses a mathematical model of the system together with an identification algorithm for determining the operating mode of the plant to perform the FDI. Comparatively, the latter direction is a more cost effective approach. Generally, the analytical redundancy approach for FDD is further categorized into *quantitative* model-based methods and *qualitative* model-based methods. The former model-based methods are commonly based on

1. state estimation;
2. parameter estimation;
3. parity space; and
4. combination of the above three.

On the other hand, the qualitative methods use artificial intelligence (AI) techniques, such as pattern recognition, neural networks, etc. An ideal fault detection and diagnosis are expected to pose several characteristics. Venkatasubramanian et al., in [VRYK03a, VRYK03b] emphasized the following characteristics:

1. Quick detection and diagnosis,
2. Isolability,
3. Robustness,
4. Novelty identifiability,
5. Classification error estimate,
6. Adaptability,
7. Explanation facility,

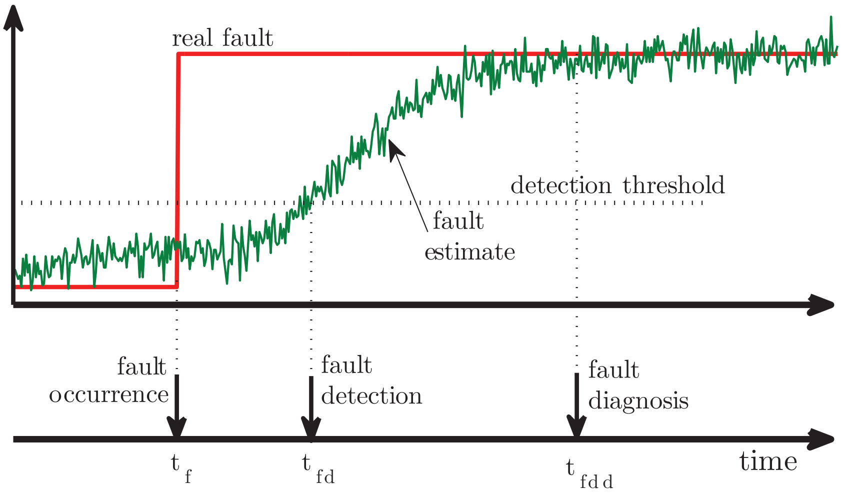


Figure 1.5: Visualization of time-delay in FDD process

8. Model requirements,

9. Storage and computational requirements, and

10. Multiple faults identifiability.

In an active FTC system, FDD based algorithms first provides the complete information about a fault in real-time so that a controller could subsequently be reconfigured. Fig. 1.5 illustrates the time-map in a fault detection and diagnosis process [Kan04]. In an integrated FDD-CR approach to the FTC, the passage of time during the fault diagnosis phase prior to the control reconfiguration phase have a crucial impact in an overall AFTC system. In the visualization of Fig. 1.5, it is assumed that the FDD scheme is based on the direct estimation of a fault signal, thus, the diagnosis delay is often experienced due to "gradual convergence" while determining a precise faulty information.

1.5 Fault-tolerant Control Methodology

In many automatic control system applications, it is important to employ the best appropriate control action to ensure a continuous safe operation of the system against occurring faults. This can be achieved by using various control reconfiguration techniques. There lies a difference in the variable use of the terms "restructurable" and "reconfigurable" control. Reconfigurability implies that the system with a fixed structure can be modified to oppose an unpermitted change. This requires only modifying the controller parameters online in response to faults. On the other hand, restructurability subsuming reconfigurability, implies that not only the parameters, but the system structure itself can be changed to accommodate the unpermitted changes. In this section, an overview of existing reconfigurable FTC strategies is presented.

1.5.1 Passive Methods

Passive FTC systems are based on the robust controller design techniques. The prime aim is to synthesize one controller architecture with fixed parameters that makes the closed-loop system insensitive to certain faults. In this method, the impaired system continues to operate with the same controller and the same system structure. The two main approaches dealing within passive FTC are discussed below.

1.5.1.1 Reliable Control

Reliability is an idealistic goal of fault-tolerant control that requires repeatability of the system stability and its performance. In some particular cases of anticipated faults, a passive fault-tolerance is normally used in connection with a reliable control [VMP92]. In this regard, a fixed (or unreconfigurable) controller is designed such that it optimizes, in some sense, the so-called *worst-fault performance* for a set of anticipated faults. Various design methodologies of the reliable control for passive FTC are discussed in [JZ00], [ZJ98], [YWS00].

1.5.1.2 Robust Control

Within a feedback control system, the robustness against disturbances and modeling errors is difficult to achieve, however, it is one of the basic requirements. Unknowingly intruding signals from sensors, actuators introduce disturbances, while an imperfect matching between a theoretically designed mathematical model, and the real functioning process causes modeling errors. If these modeling errors and disturbances affect the nominal dynamics of the system to an unacceptable level, then they can also be considered, in some sense, faults. Thus, the robustness analysis assists in designing controllers such that the system becomes insensitive to these faults. This is achieved mainly by assuming a restricted repertoire of to-be-likely occurring faults. These approaches are usually based on *quantitative feedback theory* [KPH97] or *robust* $\mathcal{H}_\infty$ *control theory* [ZR01], [NS03].

1.5.2 Active Methods

Despite the well-known existing robust controller design methodology in the literature, it poses new problems and challenges that might not appear in the conventional controller designs for AFTC systems. An important criterion to judge the suitability of any control method is to analyze the real-time implementation of the controller. In this regard, following requirements should be satisfied within an FTC system [ZJ08]

- control reconfiguration must be done under real-time constraints;
- the reconfigurable controller should be designed automatically with minor trail-and-error and human interactions; and
- the methods selected must provide a solution even if the solution is not optimal.

1.5.2.1 Pseudo Inverse

The principle of the pseudo-inverse method (PIM) is to modify the constant feedback gain matrix in the state-feedback control law such that the reconfigured system approximates the nominal model closely. Considers again the nominal linear continuous-time system

$$\Sigma_{nom} : \begin{cases} \dfrac{d}{dt}x(t) = A_n x(t) + B_n u(t) \\ \quad y(t) = C_n x(t) \end{cases} \tag{1.9}$$

with the linear state-feedback control law $u(t) = K_n x(t)$, where the subscript n denotes the nominal parameters of the system, under an assumption that the state-vector $x(t)$ is available for measurement. Under a fault occurrence, the post-fault system model can be represented by

$$\Sigma_{faulty} : \begin{cases} \dfrac{d}{dt}x^f(t) = A_f x^f(t) + B_f u^f(t) \\ \quad y^f(t) = C_f x^f(t) \end{cases} \tag{1.10}$$

with the reconfigured control law $u^f(t) = K_f x^f(t)$, where K_f is the to-be-determined feedback gain matrix. Various methods were presented in the literature to compute this matrix. The method demonstrated in [Ost85] is based on determining the K_f such that the closed-loop state-transition matrix for the faulty system given by $(A_f + B_f K_f)$, approximates the transition matrix of the normal plant $(A_n + B_n K_n)$. This yields the K_f matrix

$$K_f = B_f^{\dagger}(A_n - A_f + B_n K_n) \tag{1.11}$$

where $B_f^\dagger$ denotes the pseudo-inverse of B_f. This is relatively a fast solution and very suitable for the online implementation providing the information received from the FDD module is very accurate. Owing to this computational simplicity, it is one of the most cited methods in AFTC research community. It turns out that the feedback gain of an optimal control law computed by (1.11) does not always guarantee to stabilize the closed-loop system. To overcome with this issue, [GA91] describes the modified pseudo-inverse method (MPIM) in which the difference between the closed loop state-transition matrices of the nominal and the post-fault model is minimized subject to the stability constraints. This enforces the system to recover the specified performance as closely as possible. According to MPIM, the feedback gain matrix is then computed as:

$$K_f = \arg\, \min_{K_f} \|(A_n + B_n K_n) - (A_f + B_f K_f)\| \tag{1.12}$$

This, however, results in a constrained optimization problem that indeed increases the computational burden, whenever evaluated online. The stability issues in PIM are re-visited in [Sta05] where it is proposed to use a set of admissible models of the plant ensuring the guaranteed stability. Consequently, the classical method and the modified pseudo-inverse method have been extended by using a set of admissible models in real-time. The work presented in [TNS98], [NSHT00] also investigated a similar kind of approach, where the reconfigured control action $u^f(t)$ is directly computed from the nominal control law $u(t)$ by $u^f(t) = B_f^\dagger B_n u(t)$.

1.5.2.2 Eigenstructure assignment

Eigenstructure assignment (EsA) methodology to deal with the reconfiguration mechanism is considered to be a powerful technique than PIM and MPIM [KA96]. The principle is to place the eigenvalues of the closed-loop system and their associated eigenvectors, via the feedback control laws, that can meet the closed-loop design specifications. Precisely, the main objective of this reconfigurable control system design is to synthesize a feedback gain matrix so that the closed-loop eigenvalues of the reconfigured system become similar to those of the pre-fault system [ZJ01], i.e.

$$\lambda_i^f = \lambda(A_f + B_f K_f) = \lambda_i = \lambda(A_n + B_n K_n), \forall i = 1, 2, ..., n \tag{1.13}$$

where λ_i denote the eigenvalues of the system while at the same time minimizing the 2-norm of the difference between the corresponding eigenvectors. Consider v_i as the eigenvectors corresponding to the eigenvalues $\lambda_i, i = 1, 2, ..., n$, then using the EsA method we can compute the state-feedback gain matrix

K_f for the faulty model (1.10) as a solution of the following optimization problem

$$\text{EsA}: \begin{cases} \text{Find} & K_f \\ \text{such that} & (A_f + B_f K_f)v_i^f = \lambda_i v_i^f, i = 1, 2, ..., n \\ \text{and} & v_i^f = \text{arg min}_{v_i^f} \|v_i - v_i^f\|_{W_i}^2 \end{cases} \tag{1.14}$$

where $\|v_i - v_i^f\|_{W_i}^2 = (v_i - v_i^f)W_i(v_i - v_i^f)$. Indeed, it is the least-square optimization problem, however, this does not impose any computational burden within this approach. The advantage of EsA is worth noting whenever the performance specifications are given in terms of the system eigen-structure. The eigen-structure of the closed-loop system can be determined precisely to perform the stability analysis and achieving the specified dynamical performance analysis. It turns out that the resulting system performance employing this method might not be optimal, in some sense, since the model mismatching issues and the FDD uncertainties cannot easily be incorporated within this optimization framework.

1.5.2.3 Model Following

The model-following approach for active FTC systems is an attractive candidate to design an online controller. Here, the goal is to emulate the performance characteristics of the reference model, with or without faults or failures. The ideal form of the model following approach is termed as the *Perfect Model-Following* (PMF) approach. Consider the following reference (ref) model

$$\Sigma_{ref}: \begin{cases} \dfrac{d}{dt}x^M(t) = A_M x^M(t) + B_M r(t), \\ \quad y^M(t) = C_M x^M(t) \end{cases} \tag{1.15}$$

where $r(t)$ is the reference signal. Given the open-loop system (1.9), the control actions are composed of the matrices K_r and K_x of the state-feedback given by

$$u(t) = K_r r(t) + K_x x(t). \tag{1.16}$$

With the above control actions, the reference model and the closed-loop system can be written as

$$\begin{aligned} \dot{y}^M(t) &= C_M A_M x^M(t) + C_M B_M r(t), \\ \dot{y}(t) &= (C_n A_n + C_n B_n K_x)x(t) + C_n B_n K_r r(t). \end{aligned}$$

Since the objective is to precisely match the above two models, the PMF can be achieved by selecting the following feedback gain matrices given by

$$\text{PMF:} \begin{cases} K_x &= (C_n B_n)^{-1}(C_M A_M - C_n A_n), \\ K_r &= (C_n B_n)^{-1} C_M B_M \end{cases} \tag{1.17}$$

where it is assumed that the system has an equal number of input-output, and the inverse of the matrix $C_n B_n$ exists. In 1.17, the control design process is demonstrated for the fault-free case (or the nominal system). As a matter of fact, the similar procedure can be employed for the faulty case as well provided the system parameters are perfectly known since the feedback gain matrices depend entirely on the parameters of the working mode of the plant.

Based on the availability of the system matrices in real-time, PMF is categorized into *indirect (explicit) method*, and *direct (implicit) method*. In the former method, system matrices are estimated online while in the latter, the controller gain-matrices are estimated directly provided that a precise information of the system matrices are known *a priori* [Pat97]. In [TCJ02], a direct adaptive-state feedback control scheme is carried out while an indirect approach is presented in [ZJ02] with an emphasis on the implementation issues of an overall AFTC scheme. An advantage of using model following approaches is that they usually do not require an FDD scheme. However, the use of a simple FDD scheme can be seen in an indirect method. In addition, these methods have a limited online fault accommodation capability because of the need of a perfect post-fault model, which introduces difficulties in dealing with model uncertainty issues.

1.5.2.4 Model Predictive Control

The Model Predictive Control (MPC) approach for AFTC systems is an industrially relevant control strategy which has received a lot of attention lately in fault-tolerant community. MPC is an effective technique for solving multivariable constrained optimal control problems. In this strategy, an internal model of the plant is used to predict the system dynamics in a finite-time horizon. Based on these predictions, a cost function capturing the performance of the system is minimized over a sequence of future input commands. As discussed in [AAB$^+$01], the MPC architecture allows fault-tolerance to be embedded in a relatively easy way by: (a) redefining the constraints to represent certain faults (usually actuator faults), (b) changing the internal model, and (c) changing the control objectives to reflect limitations due to a faulty mode in operation.

In [MJ03], the validation of this approach is done by considering a practical example of the Flight 1862. It is shown that the plane crash would have been avoided by using the MPC-based fault-tolerant control system. For instance, the actuator faults within this framework can be represented by modifying the constraints of an optimization problem, and the sensor faults can be modeled by modifying the internal model of the plant. In this way, there is practically no additional optimization process that needs to be executed on-line, and this method can be viewed as having an inherent self-reconfiguration property.

With its self-reconfiguration capability, the MPC is very suitable and an attractive strategy for achieving the fault-tolerance. For an overview of the work on MPC-based FTC, refer to [KM99], [MJ03] and the references therein. The problem with these approaches arises from the fact that an optimization process is executed at each sampling instant, thus, makes the problem computationally intractable. The suitability of various fault-tolerant controllers with online accommodation, namely model predictive control (MPC), linear quadratic controller (LQR) and pseudo-inverse method (PIM) are compared in [MGB08]. In addition, the real-time implementation issues of a control strategy together with considering the time utilized in determining the parameters of the faulty mode of the plant are also addressed.

1.5.2.5 YJBK Fault Tolerant Control

The (primary) YJBK parametrization was first formulated by Youla et al. [YJB76] and independently by Kučera. It has been later used in solving feedback control problems that result in the dual YJBK parametrization [TMM97]. In the former approach, the controllers are parameterized such that it stabilizes the system while in the latter, the plant model is parameterized that can be stabilized by only one controller. The idea of using YJBK parametrization in the fault-tolerant control design is based on co-prime factorization, which mainly consists of two parts: a nominal performance controller design and an additional controller satisfying the robustness property. These controllers work in a way such that whenever a fault is detected, the controller structure is reconfigured by adding a robustness loop that makes the system fault-tolerant. The method allows for the design of passive as well as for the active fault handling. In addition, the related design method can be fitted either to guarantee stability or to achieve graceful degradation in the sense of guaranteed degraded performance [NS05]. The salient feature offered by the Youla-Kucera parametrization is that it offers an elegant and very fast solution to the control re-design problem for some particular class of faults that leave the system stable with the existing controller but unable to fulfill exactly the specified performance due to an occurring fault [BKSL03].

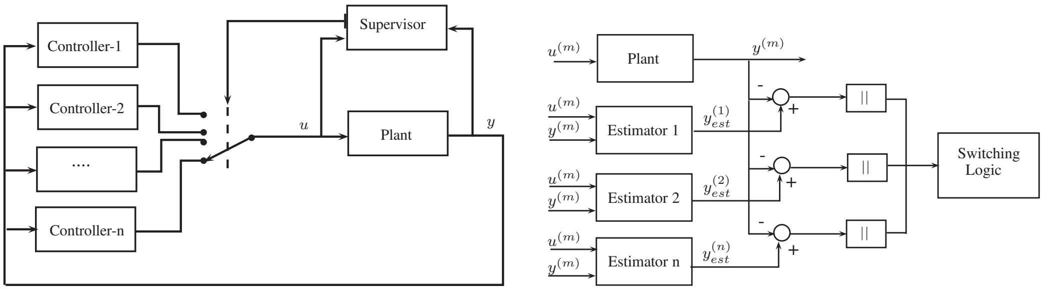
In many similar ways, a feedback fault-tolerant control architecture is proposed in [ZR01] which is also composed of two controller parts: one for the performance and other to deal with robustness issues. An active fault tolerant control strategy utilizing the YJBK parametrization involves the use of two control laws. The nominal or the robust controller is always connected in the closed-loop system irrespective of the occurrence of faults. It is only the other controller which is re-configured by the parameterization with some stable parameter, in case a fault is detected. A similar kind of controller structure is also discussed in [CDZ03], which is based on the *Generalized Internal Model Control* (GIMC) architecture. Both these control structures rely heavily on YJBK parametrization. A new FTC controller implementation structure within the YJBK framework is discussed in [NP09], which includes a residual vector. These residual vectors are used for an internal feedback in the controllers. Further, these residual vectors are also used directly in connection with the fault diagnosis in the FTC architecture. In fact, a FDD module is always embedded within these control structures, which requires the precise information about the nominal model and the post-fault model in real-time. Another drawback is that the additive faults cannot be detected or isolated. Particularly, this approach is based on detecting the closed-loop instability introduced by faults via the so-called dual YJBK matrix.

1.5.2.6 Multiple-Model Approach

The Multiple-Model (MM) approach is another model following technique for AFTC systems. This approach belongs to the class of projection-based methods using the adaptive control framework. Unlike to the previous discussion, here a controller is not re-designed. In this approach, a set of linear models is used to describe the system for different operating modes or under various faulty conditions. In this way, considering a pre-defined bank of models, a bank of controllers is designed and operates in real time. The notion of the multiple model has been used for various applications in the FTC community. In [TSP03], [ZJ01], the same notion is used to characterize the FDD module instead of controller re-configuration. In FDD module characterization [TSP03], it generates the residual signal for a bank of FDD modules. Linear quadratic (LQ) controllers are then pre-designed for each corresponding operating mode of the system. Based on the knowledge of the residual vector, one of the controllers is switched into the loop with the best matching current-operating mode. In [ZJ01], based on the retrieved fault diagnosis information incorporating the multiple-models, the EsA approach is used to design the controllers online. In [KV00], it is used for weighted control allocation to accommodate the fault effects. The key in their design process is to develop an on-line procedure that determines the global control action through the (probabilistically) weighted combination of different control actions.

The multiple-model method also proves to be a very attractive tool for the modeling and the control of non-linear systems [YIZB03]. However, these approaches usually consider a finite number of anticipated faults and proceed by building one local model for each anticipated fault. In this way at each instant one local controller is “active”, namely the one corresponding to the model that is in effect. Moreover, if the current model is not in the pre-designed model set, the control action can still be synthesized by using some convex combination of local models available in a pre-defined model set. In that case, one of the disadvantages is that the synthesized control action might not be optimal. This convex combination is also termed as *control mixer* or *blending*. The use of FDD module is always seen within this approach as well. In addition to the above, the method from [BBM05] is based on multiple model fault detection and identification and fast switching among multiple controllers based on the on-line information obtained from the FDI subsystem. However, because a finite number of models were used and if none of the models coincided with the actual damage, the resulting control system could only assure that the output errors were bounded, but not that they tended to zero asymptotically.

Recently, the logic based switching control (LBSC) has attracted attention within the MM framework. The main idea is to build a supervisor, that comprises a set of estimators followed by the performance evaluation of those estimators, and a switching logic scheme as depicted in Fig. 1.6. Each estimator reconstructs the plant output in either one of the healthy or faulty working modes that yields an indicative signal. This signal is used to determine the current working mode of the plant and the corresponding control law is then switched into the loop. A small difference from the MM method is that no mixing of control actions is performed in LBSC, i.e. only one controller is active at each time instant. However, continuous switching among two or more controllers can be seen to achieve the desired objective [YCJ10]. In [BM98], this indicative signal is generated by comparing the outputs of the local models with the measured system output. Based on some norm of this output estimation error, the corrective decisions are taken in corresponding to the model that best describes the current operating mode of the system.



(a) Logic-based switching architecture

(b) Structure of Supervisor

Figure 1.6: Control Switching approach to FTC

In all these approaches, either a plant model is used, or it is estimated in real-time. A bank of filters/models has to be pre-designed that work in parallel with the plant to identify precisely the current working mode. This introduces the system identification delay, and raises model mismatching issues (see Fig. 1.5). During this time and from the viewpoint of real-time constraints, the current controller controls the (faulty) system which is not a right controller for the closed-loop and can drastically deteriorates the system performance. In [YCJ10], preliminary efforts were made to design an FTC scheme using the classical concept of arbitrary switching theory discussed in [Lib03]. The scheme is developed in a case where a fault is not accommodated by a single controller alone i.e. no right controller is present in the bank, but the whole process can still be controlled via a periodic switching between the other controllers. An interesting real-time approach was proposed in [YK05] where the (constrained) control switching occurs based on the control performance. Recently, an AFTC scheme for the *successive faulty case* is designed in [YJC11].

1.5.2.7 Data-driven approaches

In general, the system faults are grouped into three broad categories as discussed earlier. The purpose of an FDD unit is to use available signals to detect, identify, and isolate the possible faults at different locations in the system. In the response, an overall FTC scheme calculates the required actions so that the system can continue to operate safely. As mentioned above, the model-based FDI approaches to the FTC, though quite effective, fails to fulfill the real-time constraints. A research community is thus formed around the data-driven based FDD and subsequently, the FTC. The information about the fault is extracted from the data generated by the system. Later, the controller reconfiguration module performs its operation to fulfill the *anytime* fail-safe property. Data driven FDD has gone through three main phases in its development. These three phases are referred to as the signal based FDD, multi-variable statistics based FDD, and the knowledge-based FDD [HTYJLM09]. The common feature in these methods is they all use a raw system data and process knowledge to carry out the required FDD.

Recently, in [Don09], the notion of data-driven method is used separately for fault detection and identification, and controller reconfiguration. The suggested algorithms are based on the subspace predictive control (SPC), thus, rely heavily on a sequence of Markov parameters identified from the data. Another subspace-based approach without demonstrating controller reconfiguration is presented in [DZN$^+$09]. A key step in these approaches is to predict the future output (in terms of Markov parameters), which maps the past I/Os and future inputs to the future outputs of the system. The identified predictor then parameterizes a so-called predictive controller. The closed-loop SPC skips the realization of the system model and relies only on the identified Markov parameters. To identify these parameters, the inputs to the plant have to be “informative” enough, i.e., they must persistently excite the system to a sufficient order.

1.5.3 Analysis of Fault-tolerant systems

Fault-tolerant systems may degrade performance whenever a fault occurs, but an occurring fault will not develop into a failure at the system level, if this could be prevented through proper action in the programmable parts of a control loop. The performance of an overall FTC system (FTCS) depends on many factors [ZJ08], such as

- the speed and accuracy of the online FDD scheme,
- availability of the remaining healthy (functional) actuators or sensors,
- strategy to utilize the hardware or analytical redundancy in the system,
- the type of control strategies adopted in the reconfigurable controller design, and
- the integration of these components to form an overall AFTCS.

Many applications address a set of faulty situations that are known in advance, for example, when actuator outages are considered [YWDC06]. Even when an appropriate reconfiguration strategy is utilized, particularly, sensor fault, and more generally, any type of fault falls in the same category [BKSL03]. Designing the control law, such that the aftereffects of an occurring fault are compensated, is usually carried out through passive or active fault-tolerant schemes, as discussed above. Nevertheless, whatever the (passive or active) FTC strategy is, its feasibility is obviously dependent on the recoverability (or fault coverage [Wu04b]) of each fault [Sta02]. Reconfigurability is concerned with the possibility either to accommodate the faults or to reconfigure the system when faults occur. In this context, system component based Failure Mode and Effects Analysis (FMEA) is considered to be the first step for a systematic design of FTC systems, which includes various measures, namely dependability analysis, control reconfigurability analysis, fault coverage analysis. An FMEA deals with system components, viz. sensors, controllers, actuators, faults and the propagation of fault effects. This preliminary phase determines how fault effects arising from the component relate to faults at inputs, outputs, or internal parts. Based on this, a complete coverage of likely occurring faults within the system and their possible accommodation measures are provided at the outset together with the performance specifications that a system can achieve. A logical boolean mapping of faults is developed in [Bla96] demonstrating the propagation of faults and their effects within the system. Given a set of system faults $f_c \in \mathcal{F}$, and a set of effects $e_c \in \mathcal{E}$, the fault propagation scheme can be expressed by

$$e_{ci} \leftarrow P_i^f \otimes f_{ci}$$

where $P_i^f : \mathcal{F} \times \mathcal{E} \to \{0, 1\}$ is a boolean matrix with i representing the component identifier and $\otimes$ the inner product disjunction operator. When faults effects propagate from other components, we get, at level i

$$e_{ci} \leftarrow P_i^f \otimes \begin{bmatrix} f_{ci} \\ e_{c(i-1)} \end{bmatrix}. \tag{1.18}$$

Equation (1.18) is a surjective mapping from faults to effects, that is to say, there is a unique path from fault to end effects. However, there could be several unlike faults that may cause the same end effects.

Satisfying real-time constraints and considering the dynamical nature of the system, it imposes many challenges while designing an effective FTC scheme. Usually a very limited amount of time is available to carry out the post-fault model construction (estimation) and for controller reconfiguration. The trade-off among the various design objectives and the interaction among different subsystems have to be carried out on-line. These issues are associated with modeling, stability, performance, robustness, non-linearity, simulation, implementation, and applications. Other interesting approaches that are recently developed are based on the use of virtual sensors/actuators [PTA10], progressive accommodation of fault [Sta04]. These approaches rely on the optimal LQR-based controller reconfiguration mechanism.

In addition, due to the historical reasons and taking into account the complexity of the problem, most of the research on FDD and Reconfigurable Control (RC) is carried out as two separate entities. More specifically, most of the FDD techniques are developed as a diagnostic or a monitoring tool, rather than an integral part of the FTCS. As a result, some existing FD methods may not be able to satisfy the need of controller reconfiguration [ZJ08]. On the other hand, most of the research on reconfigurable control is carried out assuming the availability of the perfect FDD. This in turn demands for the availability of complete information about the post-fault plant model which might not be accessible all the time. Few work has been done on reducing the time taken in controller reconfiguration mechanism. However, no treatment for the time taken in the FDD module is seen. In most of the above briefed AFTC methods, the stability of the post-fault system cannot be guaranteed during the period an FDD unit is performing its diagnostic functions. Moreover, in these AFTC approaches, it always requires a perfect FDD unit to carry out controller reconfiguration mechanism. A promising approach is proposed in [YK04a] that does not utilize an explicit FDD module to deal with AFTC systems. The approach is based on the framework of unfalsified adaptive control theory [ST97].

The propagation of transients within an integrated FDD-CR based FTC system also plays an important role in judging the fault-tolerance scheme. When the considered control laws are static in nature, no precaution is required during the reconfiguration process. However, the situation is different for dynamic controllers. Within an AFTC system, undesirable transients may occur, which are harmful to the safe operation of the system. The consequences of these transients may cause saturations in actuators, and damage to components in the system. Therefore, such transients should be minimized as much as possible. This phenomenon is often termed as the bumpless phenomena [HKH87] or smooth switching [SKP00]. The potential solutions in reducing these reconfiguration transients may lie in how to manage the system/controller states or command inputs. A brief comparative study of the bumpless transfer of controllers has been carried out in [PAGB10]. The case of switching between the stabilizing multi-variable controller is discussed in [NSA04] in the YJBK framework. A promising approach for bumpless transfer has been proposed in [YK07] which is based on the parametrization of candidate controllers. More comprehensive treatment in the transition management for reconfigurable control systems can be found in [GCW$^+$03].

1.6 Scope of the thesis

The overview of various active approaches discussed in the previous section shows that a plenteous literature is present in the field of fault-tolerant control. As it is argued in the surveys done by Zhang et al [ZJ08] and Patton [Pat97], there are still certain areas that have not yet received the required attention. Regardless of having a gap of more than a decade between the two surveys, one of the serious prevailing issues is the real-time implementation of an *integrated FDD-CR* approach to deal with FTC systems. Some of the main issues (I) are now highlighted [JYS12d, JYS12c, JYS11c], which laid the foundation for this thesis and will be resolved in the rest of the chapters.

(I-1). Precise knowledge of *a priori* plant model in real time. The problem of fault-tolerant control can often be handled using the following two approaches: model-based approach and model-free approach. In the former approach, the full information of the operating plant is/should be known *a* *priori*. Accordingly, a sub-system (state observer, output observer, Kalman's filter, etc.) is built, which reconstructs the plant output and diagnoses the fault. It is while constructing this sub-system that requires the precise information about the working model of the plant. The aftereffects of any occurring fault are then accommodated using the CR module. On the other hand, in so-called model-free approaches, the information of a fault is extracted using the observed measurements. Nevertheless, *a priori* knowledge of the plant is as well required during the online estimation of a fault [DZN+09]. Hence, the knowledge of any *a priori* plant model is mandatory at any time while dealing with fault-tolerant control systems. This may elicit model mismatching issues that can generate false alarms even in a fault-less situation.

(I-2). Strong dynamical interaction between FDD and CR module. In the schematic representation of FTC systems as illustrated in Fig. 1.3, the FDD module monitors the current working mode of the plant. Under a fault occurrence, the generated residual vector indicates the detection of malfunction. In general, this residual is generated by detecting a mismatch between the model used in the FDD module and the current working mode of the plant. Subsequently, the supervisor reconfigures the controller. However, the particular issue which has not received much attention in the FTC research community is the study of "post-fault FDD module". In this regard, the supervisor together with control reconfiguration has to update the FDD module as well for the new operating mode of the plant such that there does not occur any false alarm after fault accommodation. In the closed-loop environment, this causes a strong dynamical interaction between the FDD module and the supervisor. For an instance, consider the case of *disgraceful performance degradation* where the control objectives are modified to achieve partial tolerance to faults. This implies that after the accommodation of fault, the current operating plant still enfolds some unaccommodated dynamics due to an occurred fault. As a result, the fault diagnosis module must be *adaptive* with respect to the CR module such that the further modeling issue can be averted. Note the flow of two-way information between the supervisor and the FDD module within the supervision unit which is clearly reflected in the Fig. 1.3.

(I-3). Time-delays at various stages in the overall AFTCS. As it is pointed that the classical AFTC schemes require two cascaded distinctive modules. These two modules, no doubt, involve their respective time delays, namely the fault detection and diagnosis delay, and the controller reconfiguration delay [BKSL03], [YJ11]. Collectively, we termed this delay as the fault accommodation delay, i.e., the interval between the occurrence of a fault and the controller reconfiguration. We now illustrate the fault accommodation delay in an active FTC system (refer to Fig. 1.4 for the taxonomy of FTC systems) using the time-map. In an online redesign approach to AFTCS, generally a controller is designed using the LQR-based optimal control [Sta04]. This requires to solve a new, i.e. for the post-fault model, Algebraic Riccati Equation iteratively. This implies that the optimal solution is not available instantly, however, it takes some time to generate a new control law. The passage of time is usually known as the controller reconfiguration delay in the traditional scheme. The dynamics of the system working under these two cascaded modules are distributed in the following time-periods [JYS12f].

1) $t \in [0, t_f[$: the plant is in the nominal operating mode (or fault-less) and the applied controller is the nominal one.

2) $t \in [t_f, t_{fd}[$: the plant is in the faulty mode, but the FDD algorithm has not yet detected, isolated and estimated the fault. Therefore, the current controller is still the nominal one.

3) $t \in [t_{fd}, t_{fdd}[$: the plant is in the faulty mode, and the FDD algorithm has detected the fault but has not yet isolated and estimated the fault. Therefore, still the current controller is the nominal one.

4) $t \in [t_{fdd}, t_{ftc}[$: the plant is in the faulty mode, and the FDD algorithm has detected, isolated and estimated the fault but a new controller for the faulty mode has not yet been computed.

5) $t \in [t_{ftc}, \infty[$: the plant is in the faulty mode, the new controller has been computed. Subsequently, it makes an interconnection with the plant.

In projection-based AFTC schemes, there is no need to design a new controller on-line. A bank of pre-designed controllers are installed, and under the constrained switching, an appropriate control law is switched instantly as soon as the fault is diagnosed. Since no new controller is designed online, the controller reconfiguration delay is not seen in this approach. From the time-map as shown in Fig. 1.7, the FDD delay has been always there regardless of using any approach to AFTC schemes. Considering the real-time constraints, a significant attention has to be entail while handling these time delays. In the interval between an occurrence of a fault and its accommodation, the real-time performance of any FTC scheme become a concern.

Reacting to the above highlighted issues, we deal with real-time fault-tolerant control systems in this thesis. A traditional active FTC system undergoes two cascaded stages. This might be one of the basic causes of various industrial disasters as demonstrated before, since there might not be enough

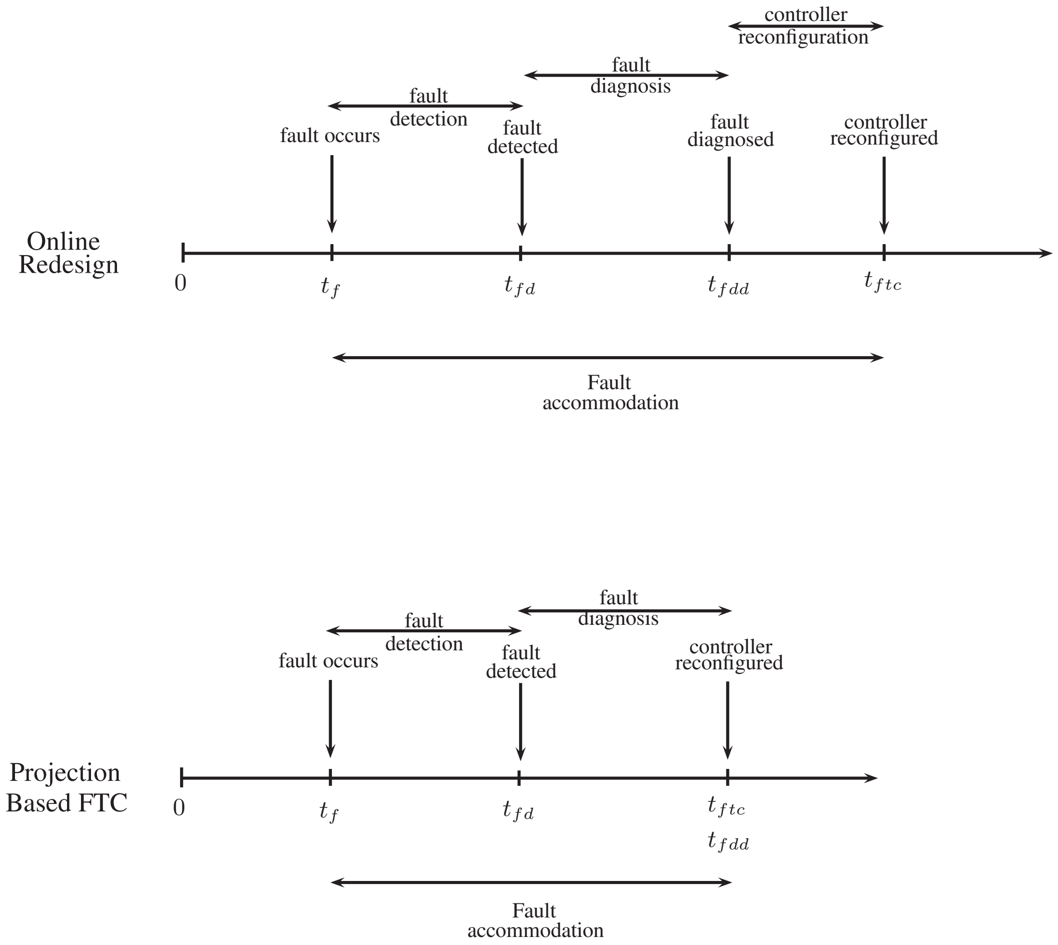


Figure 1.7: Visualization of the time-delay in FTC systems

time available to perform both the stages. Seeing these shortcomings and stating precisely, the objective of this thesis is to develop generic methods for fault-tolerant control based on real-time trajectories generated by the system subject to faults. The main contribution within the work intends to formulate and to demonstrate model-free approaches to AFTC that does not require any *a priori* information about the plant in “real-time”. To perform this task, we use the mathematical framework of behavioral system theory. In the last decades, it has been seen that the behavioral point of view has received an increasingly broader acceptance as an approach for modeling dynamical systems, and now it is generally viewed as a cogent framework for system analysis [Wil91], [Wil97], [VW99], [RW01], [TW02], [vdS03], [BVW06], [Wil07].

1.7 Outline of the thesis

In this thesis, we deal with the issues when the controlled system becomes faulty. The control of a faulty system addresses the concept of fault tolerant system. This implies that an AFTC problem is concerned with the control problem subject to the working mode (healthy/faulty) of the system [BKSL03]. The control problem is completely defined by the triple

$$< \mathfrak{O}, \mathcal{P}^{\mathtt{mode}}, \mathfrak{U} > \tag{1.19}$$

where

- The objective $\mathfrak{O}$ defines what the system is expected to achieve. This implies that the system should satisfy certain closed-loop performance specifications.
- The working modes $\mathcal{P}^{\mathtt{mode}}$ are functional relations that the controlled system satisfies over a time. This represents the state and measurement equations of the working mode of the plant within the state space representation.
- The set $\mathfrak{U}$ represents the admissible control laws. These control laws are designed in such a manner, when implemented satisfies the control objective.

Let us analyze the impact of faults on the control problem. An occurrence of a fault on the system transforms the control problem from $< \mathfrak{O}, \mathcal{P}^{\mathtt{h}}, \mathfrak{U} >$ into $< \mathfrak{O}, \mathcal{P}^{\mathtt{f}}, \mathfrak{U} >, \mathtt{f} \in \mathfrak{F}$ where $\mathfrak{F}$ indexes the set of all considered faults, $\mathcal{P}^{\mathtt{h}}$ is the set of healthy (or nominal) constraints, and $\mathcal{P}^{\mathtt{f}}$ is a set of faulty constraints. Generally, an occurring fault does not change the system objective because the main idea of the fault tolerant control system is to try to reach them even in the presence of faults. However, this would sometimes be possible or not. A case when specified objectives are not able to be achieved, the problem is transformed into finding the new objectives being less restrictive such that the system still manages to satisfy the fault-tolerance property (anytime fail-safe) but with somewhat a lesser performance, i.e. disgraceful degraded performance.

Unlike to the integrated FDD-CR (or model-based) fault-tolerant control strategies, where the prime motive is to determine the constraints $\mathcal{P}^{\mathtt{mode}}$ of the faulty system during the FDD operation, here in the proposed approaches we do not have access to *a priori* knowledge about these constraints in either of the working modes of the plant. In this thesis, we directly deal with time-valued trajectories generated by the system in real-time, which is independent of any representation of the plant. In this regard, the representation-free feature of behavioral theory motivates us to borrow its mathematical framework. More details on the behavioral approach to systems and control are presented in the next chapter. The underlying aim is to make an *interconnection between the controller and the working mode of the plant subject to faults* without utilizing an explicit FDD module. Real-time constraints are thereby relaxed, since reconfiguring an impaired control to a proper fault-tolerant, one only need to analyze the trajectories of faulty dynamics. This avoids any on-line fault estimation and iterative control re-design steps.

1.8 Organization of the thesis

The thesis is divided into three parts. The first part consists of two chapters. In Chapter 1, we have provided an extensive review of the existing fault-tolerant control approaches in the literature. In this, we posed the motivations and objectives behind this research work. Chapter 2 deals with introducing the mathematical tools borrowed from the behavioral system theory. Indeed, no prior work has been seen in the FTC community using this mathematical framework. Therefore, this chapter lays the strong basis to support the posed objectives within this work.

The second part of the thesis consists of three chapters, in which we present the real-time solutions to solve an FTC problem presented in the first part. In the behavioral framework, a control problem is treated as an interconnection of two dynamical systems. In Chapter 3, we demonstrate the significance of these interconnections and the other requirements on the type of interconnections. The first real-time solution based on the projection approach to an FTC problem is demonstrated in Chapter 4. Chapter 5 demonstrates an another approach, namely the online redesign based approach, to deal with an FTC problem.

The last part of the thesis consists of two chapters, which is quite significant from the practical implementation viewpoint within an AFTC system. The transient management is one of the major issues that needs a special attention during the controller reconfiguration process. Chapter 6 presents a methodology to guarantee the *real-time smooth interconnection.* In Chapter 7, the validation of the theory developed in the previous chapters has been done by taking various case-studies.

Chapter 2

Behavioral Paradigm

Contents

2.1 Dynamical systems 35

2.2 Linear Differential systems 39

2.3 Equivalent and minimal representations 41

2.4 Latent variables and elimination 43

2.5 Observability and detectability 46

2.6 Controllability and stabilizability 48

2.7 Autonomous behavior 51

2.8 Input/Output Representation 53

2.9 Input/State/Output Representation 56

In this chapter, we introduce the basic concepts of behavioral approach that laid the foundation to deal with FTC approaches in this thesis. We shall see here how the behaviors are described for a dynamical system. The elementary properties (for example linearity, time/shift invariance) associated with a dynamical system are discussed taking this behavioral point of view. As we mentioned (very briefly) in the last chapter, the real essence of this approach lies in its representation-free description. However, for the brevity of explanation, we shall often utilize kernel representations. These representations consider the polynomial matrices for describing a system, which are more general way of representing dynamical systems since these representations can easily be translated in either external-type or internal-type of representations [Che99, section 6-7], [Wil91]. To motivate further use of behavioral framework, we have considered various follow-up examples.

2.1 Dynamical systems

The starting point of this study is to describe the notion of a dynamical system. In classical control theory, modeling of a dynamical system is the first step. Modeling a system describes the way by which the variables of the system evolve, thus, demands for various representations for the system. In case of unavailability of these representations, they are then determined utilizing the tools borrowed from the system identification community. Interestingly at any stage of the problem formulation (to be introduced in later chapters), we do not deal with identifying the system (the plant). Let s denote a vector-valued variable whose components consist of the system variables. We define the signal space, denoted by $\mathbb{S}$, where the variable s takes its values. Usually, s itself is a function of an independent variable called time, taking its values in a set called the time axis. The symbol $\mathbb{T}$ denotes the time axis. Accordingly, in the behavioral framework a dynamical system is defined as [Wil91]

Definition 2.1. (Dynamical System) *A dynamical system Σ is represented by a triple $\Sigma = (\mathbb{T}, \mathbb{S}, \mathcal{B})$ where $\mathbb{T} \subseteq \mathbb{R}$, called the time axis, $\mathbb{S} \subseteq \mathbb{R}^{\mathtt{s}}$ called the signal space and $\mathcal{B} \subseteq \mathbb{S}^{\mathbb{T}}$ called the behavior. A trajectory is a function*

$$s : \mathbb{T} \to \mathbb{S}, \ t \mapsto s(t). \quad \square$$

As mentioned earlier, the set $\mathbb{S}$ is the space in which the system time-signals take on their values and the behavior $\mathcal{B} \subseteq \mathbb{S}^{\mathbb{T}}$ is a *family* of $\mathbb{S}$-valued time trajectories. $\mathbb{S}^{\mathbb{T}}$ denotes the set of maps from $\mathbb{T}$ to $\mathbb{S}$. Thus, an element of the behavior ($s \in \mathbb{S}^{\mathbb{T}}$) is a map with domain $\mathbb{T}$ and co-domain $\mathbb{S}$. We view a dynamical system as an exclusion law that indicates which trajectories are admissible within the system. Stating otherwise, a trajectory is possible if it is consistent with the laws describing the system. Thus,

$$\mathcal{B} = \{s : \mathbb{T} \to \mathbb{S} | s \text{ is compatible with the laws of } \Sigma\} \tag{2.1}$$

For the quick illustration, consider an example.

Example 2.1. *According to Newton's law of motion, the force (**F**) required to accelerate a physical body is directly proportional to the mass (**m**) and the acceleration (**a**) of the body. Mathematically, it is given as*

$$F = m \cdot a$$

A dynamical system that describes this theory can be written as a triple $(\mathbb{T}, \mathbb{S}, \mathcal{B})$. In this system, the acceleration and the force variables are the function of time. Thus, the behavior of this system is completely described by the trajectories of these variables. Each variable is vector valued, so we take $\mathbb{S}$ as three-dimensional space, i.e. $\mathbb{R}^3 \times \mathbb{R}^3$. The variables evolve continuously, so we take $\mathbb{T}$ as, for example, $\mathbb{R}$. The behavior $\mathcal{B}$ is then given as

$$\mathcal{B} = \{(F, a) \in \mathbb{R} \to \mathbb{R}^3 \times \mathbb{R}^3 | \forall t \in \mathbb{R}, F(t) = m \cdot a(t)\}.$$

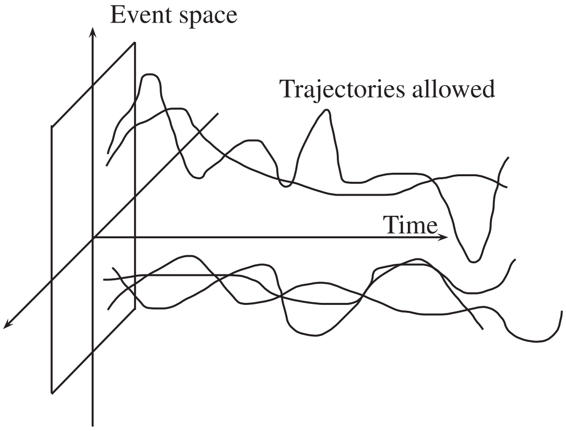


Figure 2.1: Permitted trajectories defining the behavior

If we denote the position of the center of gravity of the body as $\mathbf{x} \in \mathbb{R}^3$*, then the behavior of the dynamical system that describes the relation between the force and the position of the body is*

$$\mathcal{B} = \{(F, x) \in \mathbb{R} \to \mathbb{R}^3 \times \mathbb{R}^3 | \forall t \in \mathbb{R}, F(t) = m \cdot \frac{d^2x(t)}{dt^2}\}. \quad \Box$$

The trajectories describing the behavior of a dynamical system generally follows certain equations, which come from a description of the various laws governing the system. These equations are usually termed as a *representation* of the system. Likewise, the equations in the last example constitute a representation of the Newton's law of motion, and $\mathcal{B}$ is the solution set of this equation. Note the use of representation of the system to describe the behavior. At this point, we are very clear now that it is not the differential equations that describe the behavior instead it is the solution of the differential equations.

These representations can be of either *external* type that includes only the variables of the system or *internal* type that also includes the auxiliary variable together with system variables. The relationships among variables are, however, expressed in terms of the ordinary differential equations. Thus the behavior is the set of all trajectories of the system variables that, according to certain laws, are possible in the event space (See Fig. 2.1). We now proceed with discussing the properties of a dynamical system. Here we shall study dynamical systems that are linear and time-invariant.

Definition 2.2. (Linearity) *A dynamical system* $\Sigma = (\mathbb{T}, \mathbb{S}, \mathcal{B})$ *is called* linear *if*

- $\mathbb{S}$ *is a vector space over a field* $\mathbb{R}$*, and*

- *the behavior $\mathcal{B}$ is a subspace of* $\mathbb{S}^{\mathbb{T}}$

The latter characterizes the superposition principle, *i.e.,*

$$s_1, s_2 \in \mathcal{B} \text{ and } \alpha_1, \alpha_2 \in \mathbb{R} \Rightarrow \alpha_1 s_1 + \alpha_2 s_2 \in \mathcal{B}. \quad \square$$

Consider an example from [CP99].

Example 2.2. *Let $\mathcal{B}$ denote the set of all solutions of the homogeneous differential equation $\frac{d^2}{dt^2}x + 2\frac{d}{dt}x + 3x = 0$ in the signal space $\mathbb{S}^{\mathbb{T}}$. Then $\mathbb{S}$ is a vector space over a field $\mathbb{R}$. If the differential equation is not homogeneous, then it is not a linear space.*

Time-invariance is the property of a dynamical system where the laws governing the system do not explicitly depend on time. No doubt the variables of the system evolve as the functions of time. If the time axis $\mathbb{T}$ is endowed with a commutative and associative binary operation, and the $\sigma^t s$ is defined by $(\sigma^t s)(t^\ddagger) = s(t + t^\ddagger) \forall t, t^\ddagger \in \mathbb{T}$, where $\sigma^t : \mathbb{S}^{\mathbb{T}} \to \mathbb{S}^{\mathbb{T}}$ is the shift operator, then we can also define time invariance of a behavior.

Definition 2.3. (Time-invariance) *A dynamical system $\Sigma = (\mathbb{T}, \mathbb{S}, \mathcal{B})$ is called* time-invariant *if for each trajectory $s \in \mathcal{B}$, the shifted trajectory $\sigma^t s$ is again an element of $\mathcal{B}$, for all $t \in \mathbb{T}$.* □

Considering the time-shifting property, we can now define the concatenation of trajectories.

Definition 2.4. (Concatenation) *Given a dynamical system $\Sigma = (\mathbb{T}, \mathbb{S}, \mathcal{B})$, for any two time instants $t_1, t_2 \in \mathbb{T}$, the concatenation operation $\triangle_{t_2}^{t_1}$ is defined such that for any two trajectories $s_1, s_2 \in \mathcal{B}$,*

$$s_3 = s_1 \triangle_{t_2}^{t_1} s_2 \in \mathbb{S}^{\mathbb{T}},$$

$$s_3(t) = \begin{cases} s_1(t) & t \leq t_1, \\ s_2(t - t_1 + t_2) & t > t_1. \end{cases} \quad \square$$

See Fig. 2.2 for an illustration of the concatenation of trajectories. In the remaining part of this chapter, we shall review the classes of systems that are dealt in this thesis. Regardless of an extensive treatment of these systems, which is already existed in the literature, it is essential to review them in this mathematical framework of behavioral theory.

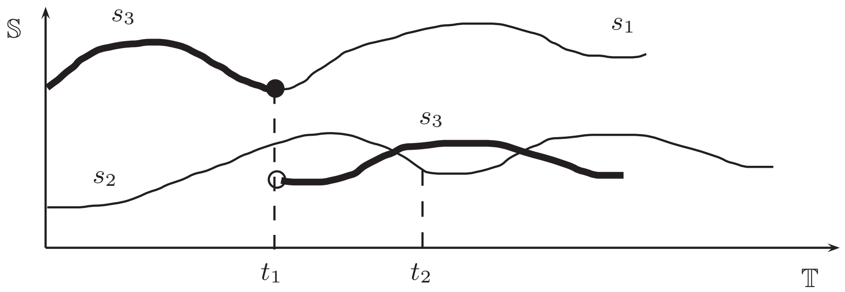


Figure 2.2: An illustration of concatenation of trajectories.

2.2 Linear Differential systems

We continue our discussion by defining a system as an exclusion law that admits only those maps $s \in \mathbb{S}^{\mathbb{T}}$ that satisfy certain laws. Recall the example 2.1, where the solutions of the differential equations describe the behavior of the system. In one of the descriptions of the behavior, there is a linear algebraic relation between $\mathbf{F}$ and $\mathbf{a}$ while the other one is an ordinary linear differential equation between $\mathbf{F}$ and $\mathbf{x}$. Systems governed by laws that are ordinary differential equations in the system variables are known as differential systems. Together with linearity and time-invariance, these systems are called linear differential systems. The set of all linear differential systems with $\mathtt{s}$ variables will be denoted by $\mathfrak{L}^{\mathtt{s}}$. Behaviors of such systems can be expressed as the set of solutions, in a suitable function space, of a system of linear, constant coefficient differential equations. It has been shown before that there exists a clear distinction between the behavior as the space of all solutions to a set of equations, and as the set of equations itself. These set of equations in which the behavior is expressed shall often be termed as the kernel representation of the behavior. The system is defined by a linear differential equation

$$R_0 s + R_1 \frac{d}{dt} s + R_2 \frac{d^2}{dt^2} s + \ldots + R_{\mathtt{n}} \frac{d^{\mathtt{n}}}{dt^{\mathtt{n}}} s = 0, \tag{2.2}$$

where $R_i, i = 0, 1, 2, \ldots, \mathtt{n}$ are real constant matrices belonging to $\mathbb{R}^{\bullet \times \mathtt{s}}$ with finite number of rows and $\mathtt{s}$ columns. Equation (2.2) can compactly be written as

$$R\left(\frac{d}{dt}\right) s = 0, \quad R(\xi) \in \mathbb{R}^{\bullet \times \mathtt{s}}[\xi], \tag{2.3}$$

with $R(\xi) = R_0 + R_1\xi + R_2\xi^2 + \ldots + R_{\mathtt{n}}\xi^{\mathtt{n}}$ where $\mathbb{R}^{\bullet \times \mathtt{s}}[\xi]$ denotes the set of $\bullet \times \mathtt{s}$ polynomial matrices with real coefficients and indeterminate ξ. Then the behavior $\mathcal{B}$ is given by the set

$$\mathcal{B} = \{s \in (\mathbb{R}^{\mathtt{s}})^{\mathbb{R}} | s \text{ satisfies } (2.3)\}. \tag{2.4}$$

The representation used in (2.3) is called the kernel representation of $\mathcal{B}$, and we often write it as $\mathcal{B} = \mathtt{ker}(R(\frac{d}{dt}))$. From the above, clearly a dynamical system is now represented by a set of operating signals given in (2.4). The shift from representing a dynamical system as an input/output processor standpoint to an equivalent set of solutions will be the key idea in the proposed FTC approach.

The only thing requires to show is that when a trajectory $s : \mathbb{R} \to \mathbb{R}^{\mathtt{s}}$ said to be a solution of equation (2.3). Different concepts of solution will result in different sets of trajectories, and thus different behaviors. There are two solution concepts that appear in the literature [PW97, Chapter 2]. They are termed as the *strong solutions* and the *weak solutions.*

Definition 2.5. (Strong solutions) *A function $s : \mathbb{R} \to \mathbb{R}^{\mathtt{s}}$ is called a strong solution of equation (2.3) if the components of $s(t)$ are often differentiable as required by the equation (2.3), and if it is a solution in the ordinary sense.* □

In order to avoid specifying how many times a function is differentiable, it is, instead, called infinitely differentiable function (denoted by $\mathfrak{C}^{\infty}(\mathbb{R}, \mathbb{R}^{\mathtt{s}})$ functions). Before, defining the concept of weak solutions, it is required to introduce the class of locally integrable functions.

Definition 2.6. (Locally integrable function) *A function $s : \mathbb{R} \to \mathbb{R}^{\mathtt{s}}$ is said to be locally integrable if for all $a, b \in \mathbb{R}$,*

$$\int\limits_a^b \|s(t)\| dt < \infty. \quad \square$$

The symbol $\| \bullet \|$ denotes Euclidean norm on $\mathbb{R}^{\mathtt{s}}$. The class of locally integrable functions $s : \mathbb{R} \to \mathbb{R}^{\mathtt{s}}$ is denoted as $\mathfrak{L}_1^{\mathtt{loc}}(\mathbb{R}, \mathbb{R}^{\mathtt{s}})$.

Definition 2.7. (Weak solutions) *The weak solutions to the differential equation 2.3 are locally integrable functions $s(t)$ that satisfy (2.3) in the distributional sense.* □

While discussing about the input/output map, we shall encounter with the class of square integrable functions, denoted by $\mathcal{L}_2(\mathbb{R}, \mathbb{R}^{\mathtt{s}})$. This implies that a function $s : \mathbb{R} \to \mathbb{R}^{\mathtt{s}}$ is said to be square integrable if $\int\limits_{t \in \mathbb{R}} \|s(t)\|^2 dt < \infty$. In a similar way, the weak solutions to (2.3) are square integrable functions $s(t)$ that satisfy (2.3) in the distributional sense. Now suppose we are interested in the $\mathcal{L}_2$-trajectories $s(t)$ that satisfy equation (2.3), then one way of describing this set of solutions, namely, the behavior $\mathcal{B}$ is given by

$$\mathcal{B} = \{s \in \mathcal{L}_2(\mathbb{R}, \mathbb{R}^{\mathtt{s}}) | R\left(\frac{d}{dt}\right) s = 0\}. \tag{2.5}$$

Obviously, the system $\Sigma = (\mathbb{R}, \mathbb{R}^{\mathtt{s}}, \mathcal{B})$ is linear and time-invariant. In fact, the linearity of $R\left(\frac{d}{dt}\right)$ results in the linearity of $\mathcal{B}$. Moreover, since the coefficients of the polynomial matrix $R\left(\frac{d}{dt}\right)$ are constant, this results in time-invariance. The equation $R\left(\frac{d}{dt}\right) s = 0$ is also called a behavioral equation. Indeed, a behavioral equation is the outcome of modeling. However, in the behavioral framework, modeling a system is to describe the behavior of the system and not to obtain just a behavioral equation. In other words, when understanding a system, we usually take care not to get drowned in a behavioral equation representing the behavior.

2.3 Equivalent and minimal representations

The starting point of introducing the behavioral point of view to describe a dynamical system lies in the fact that the behavior of the system is representation independent. As a matter of fact, a behavior $\mathcal{B} \in \mathfrak{L}^{\mathtt{s}}$ can have more than one representation. The question is then arises: Can a kernel representation of $\mathcal{B}$ be unique?

Definition 2.8. (Equivalent representation) *Two representations are said to be equivalent whenever they depict the same behavior.* □

This section contains the results from [PW97] related to equivalent kernel representations.

Theorem 2.1. *Let $\mathcal{B}^1, \mathcal{B}^2 \in \mathfrak{L}^{\mathtt{s}}$ be represented by kernel representations $R_1\left(\frac{d}{dt}\right) s = 0$ and $R_2\left(\frac{d}{dt}\right) s = 0$, respectively. Then $\mathcal{B}^1 \subseteq \mathcal{B}^2$ if and only there exists an $F(\xi) \in \mathbb{R}^{\bullet\times\bullet}[\xi]$ such that $FR_1 = R_2$.* □

Using the above theorem, we easily obtain the desired conditions on R_1 and R_2 under which they induce kernel representations of one and the same behavior. This happens if and only if F is an unimodular square matrix. Consider the following example.

Example 2.3. *Let a dynamical system $\Sigma = (\mathbb{R}, \mathbb{R}^2, \mathcal{B}^1)$ is given by the following differential equations*

$$\begin{aligned} w_1 + w_2 + \frac{d^2}{dt^2} w_2 &= 0 \\ w_2 + \frac{d}{dt} w_2 &= 0. \end{aligned}$$

The behavior $\mathcal{B}^1 \in \mathfrak{L}^2$ of the above system can be given by

$$\mathcal{B}^1 = \left\{ (w_1, w_2) \in \mathbb{R} \to \mathbb{R}^2 \,\middle|\, \begin{bmatrix} 1 & 1 - \xi^2 \\ 0 & 1 + \xi \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = 0 \right\}.$$

We can compute that $\mathcal{B}^1$ consists of trajectories (w_1, w_2), such that for all $t \in \mathbb{R}$,

$$\begin{aligned} w_1(t) &= 0 \\ w_2(t) &= \exp(-t + K), \end{aligned}$$

for some $K \in \mathbb{R}$. Now, take another behavior $\mathcal{B}^2 \in \mathfrak{L}^2$ described by the following kernel representation

$$\mathcal{B}^2 = \left\{ (w_1, w_2) \in \mathbb{R} \to \mathbb{R}^2 \,\middle|\, \begin{bmatrix} 1 & 0 \\ 0 & 1+\xi \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = 0 \right\}.$$

We can easily deduce that the last two kernel representation illustrate the same behavior, i.e. $\mathcal{B}^1 = \mathcal{B}^2$ and the unimodular matrix is

$$F = \begin{bmatrix} 1 & 1-\xi \\ 0 & 1 \end{bmatrix} \text{ which yields } \begin{bmatrix} 1 & 1-\xi^2 \\ 0 & 1+\xi \end{bmatrix} = \begin{bmatrix} 1 & 1-\xi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1+\xi \end{bmatrix}. \quad \square$$

A minimal representation satisfies the property of having a minimum number of rows. This brings us to the following definition.

Definition 2.9. (Minimal kernel representation) *Let $\mathcal{B} \in \mathfrak{L}^{\mathtt{s}}$ and let $R(\xi) \in \mathbb{R}^{\mathtt{p}\times\mathtt{s}}[\xi]$ induce a kernel representation of $\mathcal{B}$. $R\left(\frac{d}{dt}\right) s = 0$ is said to be a* minimal *kernel representation of $\mathcal{B}$ if, whenever $R'(\xi) \in \mathbb{R}^{\mathtt{g}\times\mathtt{s}}[\xi]$ induces a kernel representation of $\mathcal{B}$, then $\mathtt{p} \leq \mathtt{g}$, i.e. $\mathtt{rowdim}(R) \leq \mathtt{rowdim}(R')$.* □

Consider the following example.

Example 2.4. *Take a dynamical system $\Sigma = (\mathbb{R}, \mathbb{R}^2, \mathcal{B})$, with $\mathcal{B} \in \mathfrak{L}^2$, where*

$$\mathcal{B} = \left\{ (w_1, w_2) \in \mathbb{R} \to \mathbb{R}^2 \,\middle|\, \begin{bmatrix} 1 & 1-\xi^2 \\ 0 & 1+\xi \\ 1 & -\xi^2-\xi \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = 0 \right\}.$$

Notice that the third differential equation is nothing but the difference between the first and the second. Hence, this representation is non-minimal, as one of the rows can be removed without affecting the behavior. □

Thus, minimality is equivalent to the number of differential equations being as small as possible. The following theorem [PW97] characterizes minimality of a kernel representation.

Theorem 2.2. *Let $\mathcal{B} \in \mathfrak{L}^{\mathtt{s}}$ and let $R(\xi) \in \mathbb{R}^{\mathtt{p}\times\mathtt{s}}[\xi]$ induce a kernel representation of $\mathcal{B}$. Then the kernel representation $R\left(\frac{d}{dt}\right) s = 0$ is minimal if and only if the polynomial matrix R has full row rank.* □

Hereafter, by kernel representations, we shall always presume these representations to be *minimal representations.*

2.4 Latent variables and elimination

From the modeling point of view, it is argued in [PW97] that for many systems, in general, we need to introduce a set of auxiliary variables. Later, we 'ignore' these variables once we obtain a mathematical model of the variables we are actually interested in. Those auxiliary variables are referred to as the *latent variables* and the variables in which we are interested are referred to as the *manifest variables.* Without dwelling much into the modeling aspects, we partitioned the latent variables (denoted by ℓ), and the manifest variables (denoted by w) into a set of trajectories, i.e. $s = \texttt{column}(w, \ell)$. Thus, the definition of a dynamical system incorporating these latent variables is now given as

Definition 2.10. (Full Behavior) *A dynamical system with latent variables is a triple* $\Sigma_{\texttt{full}} = (\mathbb{T}, \mathbb{W} \times \mathbb{L}, \mathcal{B}_{\texttt{full}})$ *where* $\mathbb{T}$ *is the time axis,* $\mathbb{W}$ *is the manifest signal space,* $\mathbb{L}$ *the latent variable space and* $\mathcal{B}_{\texttt{full}} \subseteq (\mathbb{W} \times \mathbb{L})^{\mathbb{T}}$. $\mathcal{B}_{\texttt{full}}$ *is referred to as the* full behavior *of the system.* □

The dynamical system defined in the above definition is also termed as the latent variable system. Note that it is only the separation of variables into the manifest variables and the latent variables that distinguishes the definition 2.10 from the definition 2.1. We now define the manifest behavior $\mathcal{B}$ that is induced by the latent variable system.

Definition 2.11. (Manifest Behavior) *Let* $\Sigma_{\texttt{full}} = (\mathbb{T}, \mathbb{W} \times \mathbb{L}, \mathcal{B}_{\texttt{full}})$ *be a dynamical system with latent variables. The dynamical system induced by* $\Sigma_{\texttt{full}}$ *is defined as* $\Sigma = (\mathbb{T}, \mathbb{W}, \mathcal{B})$ *with the manifest behavior defined as*

$$\mathcal{B} = \{w \in \mathbb{W}^{\mathbb{T}} \mid \exists \ell \in \mathbb{L}^{\mathbb{T}} \text{ such that } (w, \ell) \in \mathcal{B}_{\texttt{full}}\}. \quad \square \tag{2.6}$$

Unlike to the behavior defined in definition 2.10, we have "ignored" the latent variables $\ell \in \mathbb{L}^{\mathbb{T}}$ in the behavior defined in definition 2.11. In fact, $\mathcal{B}$ can be obtained from $\mathcal{B}_{\texttt{full}}$ by projecting $\mathcal{B}_{\texttt{full}}$ on the manifest variables. Define the projection operator $\Pi_w : (\mathbb{W} \times \mathbb{L})^{\mathbb{T}} \to \mathbb{W}^{\mathbb{T}}$ by $\Pi_w(w, \ell) = w$. In this regard, definition 2.11 of the manifest behavior $\mathcal{B}$ allows us to write $\mathcal{B} = \Pi_w(\mathcal{B}_{\texttt{full}})$.

Assuming $\Sigma_{\texttt{full}}$ to be a linear differential system, then an important question arises about the linearity of the manifest system Σ that whether it is a linear differential system. To resolve this query, Π_w needs to preserve linearity and time-invariance. Further, it has been shown in [PW97] that after eliminating the latent variables, the resulting manifest behavior is a differential system provided $\mathbb{S}^{\mathbb{T}} \equiv \mathfrak{C}^{\infty}(\mathbb{R}, \mathbb{R}^{\mathtt{s}})$.

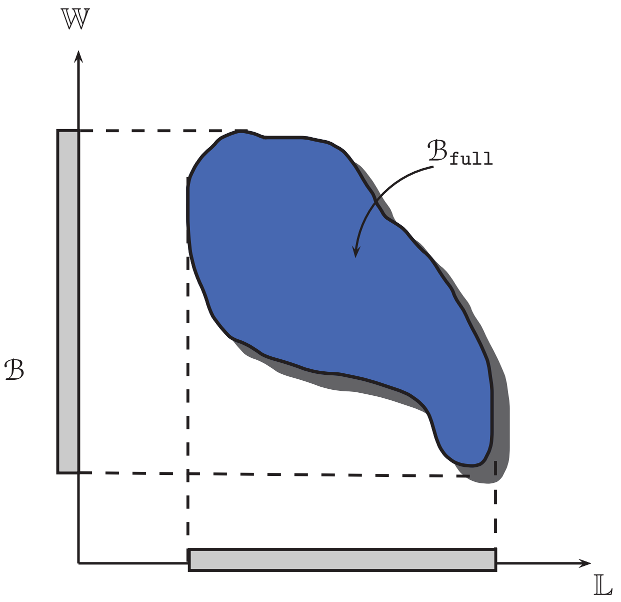


Figure 2.3: Projection of full behavior

Theorem 2.3. *Let $\mathcal{B}_{\mathtt{full}} \in \mathfrak{L}^{\mathtt{w}+1}$. Consider the behavior defined by*

$$\mathcal{B} = \{w \in \mathfrak{C}^{\infty}(\mathbb{R}, \mathbb{R}^{\mathtt{w}}) \mid \exists \ell \in \mathfrak{C}^{\infty}(\mathbb{R}, \mathbb{R}^{1}) \textit{ such that } (w, \ell) \in \mathcal{B}_{\mathtt{full}}\}.$$

Then $\mathcal{B} \in \mathfrak{L}^{\mathtt{w}}$. □

Thus, $\mathcal{B}_{\mathtt{full}} \in \mathfrak{L}^{\mathtt{w}+1} \implies \Pi_w(\mathcal{B}_{\mathtt{full}}) \in \mathfrak{L}^{\mathtt{w}}$. This theorem is called the elimination theorem. Graphically, the projection of a full behavior is shown in figure 2.3, where time axis is the common axis. In the case of $\mathbb{S}^{\mathbb{T}} \equiv \mathcal{L}_2(\mathbb{R}, \mathbb{R}^{\mathtt{s}})$, elimination of latent variables is not always possible. When it is possible, the latent variables are called properly eliminable. It was argued in [PW97] that in the context of linear differential systems, the general form of a system with latent variables is given by the model

$$R\left(\frac{d}{dt}\right) w = M\left(\frac{d}{dt}\right) \ell, \qquad R(\xi) \in \mathbb{R}^{\bullet\times\mathtt{w}}[\xi], M(\xi) \in \mathbb{R}^{\bullet\times\ell}[\xi] \tag{2.7}$$

This representation is called a *latent variable representation* or *hybrid representation*. Now we can write the full behavior as

$$\mathcal{B}_{\mathtt{full}} = \{(w, \ell) \in (\mathbb{R}^{\mathtt{w}+\ell})^{\mathbb{R}} \mid R\left(\frac{d}{dt}\right) w = M\left(\frac{d}{dt}\right) \ell\}. \tag{2.8}$$

Let $\mathcal{B} = \Pi_w(\mathcal{B}_{\mathtt{full}})$, i.e.

$$\mathcal{B} = \{w \in (\mathbb{R}^{\mathtt{w}})^{\mathbb{R}} \mid \exists \ell \in (\mathbb{R}^{\ell})^{\mathbb{R}} \text{ such that } (w, \ell) \in \mathcal{B}_{\mathtt{full}}\}. \tag{2.9}$$

As mentioned, in general, there does not exist an $R'(\xi) \in \mathbb{R}^{\bullet\times\bullet}[\xi]$ that induce a kernel representation of $\mathfrak{B}$. However, the closure of $\mathfrak{B}$ does admit a kernel representation [PW97], [Bel03]. Many intuitive concepts are explained naturally using the concept of latent variables. These variables, in the context of controllability, give the so-called *image representations*. We shall consider an example after introducing the results from [PW97], which perform the elimination operation for linear differential systems.

Theorem 2.4. *Let $\mathfrak{B}_{\mathtt{full}} \in \mathfrak{L}^{\mathtt{w}+\ell}$ be described by the latent variable representation $R\left(\frac{d}{dt}\right) w = M\left(\frac{d}{dt}\right) \ell$ with $R(\xi) \in \mathbb{R}^{\mathtt{g}\times\mathtt{w}}[\xi]$ and $M(\xi) \in \mathbb{R}^{\mathtt{g}\times\ell}[\xi]$. Let $U(\xi) \in \mathbb{R}^{\mathtt{g}\times\mathtt{g}}[\xi]$ be a unimodular matrix such that*

$$UM = \begin{bmatrix} M_1 \\ 0 \end{bmatrix}$$

with $M_1(\xi) \in \mathbb{R}^{\bullet\times\ell}[\xi]$ of full row rank. Partition

$$UR = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$$

accordingly. Then a kernel representation of $\mathfrak{B}$ is given by $R_2\left(\frac{d}{dt}\right) w = 0$. □

As an example, we consider the elimination of one of the variables from the given differential equations of a linear time-invariant system.

Example 2.5. *Consider the continuous-time system given by the following representation*

$$\begin{aligned} \frac{d}{dt}x(t) &= ax(t) + bu(t), \\ y(t) &= cx(t). \end{aligned}$$

We assume, for simplicity, that the variables $x(t), u(t)$, and $y(t)$ are all one dimensional. We also assume that $c \neq 0$. The behavior of this system can be represented by the following kernel representation.

$$\begin{bmatrix} \frac{d}{dt} - a & -b & 0 \\ c & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ u \\ y \end{bmatrix} = 0.$$

Notice that by pre-multiplying the above kernel representation by a unimodular matrix

$$U\left(\frac{d}{dt}\right) = \begin{bmatrix} 0 & 1 \\ c & -\frac{d}{dt} + a \end{bmatrix},$$

we obtain another kernel representation

$$\begin{bmatrix} c & 0 & -1 \\ 0 & -bc & \frac{d}{dt} - a \end{bmatrix} \begin{bmatrix} x \\ u \\ y \end{bmatrix} = 0.$$

Following theorem 2.4, it is seen that the variable x is properly eliminable. Consequently, the kernel representation of the behavior after the elimination is given by the second row of the above equation, i.e.

$$\begin{bmatrix} -bc & \frac{d}{dt} - a \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix} = 0. \quad \square$$

Note that the differential equations in the last example are, in fact, represents the so-called first-order state-space system with $x(t)$ denoting the state, $u(t)$ denoting the input and $y(t)$ denoting the output of the system. However, we have not made any distinction between the inputs and outputs at the moment. We shall explore this issue in the later sections. It has been shown above that for such systems, the state is properly eliminable. Generally speaking, the states are always properly eliminable for higher order systems [PW97].

2.5 Observability and detectability

Observability is one of the important concepts that plays a central role in system theory. The classical definition of observability specializes to the problem when the state of an input/state/output system is *observable* from the input/output trajectories. Indeed, the behavioral approach considers a system to interact with its environment via the *terminal variables* only. In relation to this, the well-known concept of observability (and later, the detectability) is extended viewing this more general viewpoint [PW97].

Suppose we are provided two variables (w_1, w_2) and we are interested in obtaining the values of one variable from another variable, i.e. w_2 from w_1. We can write these variables in the combined form as $s = \begin{bmatrix} w_1^T & w_2^T \end{bmatrix}^T$, see figure 2.4 for an illustration. In this regard, the first component w_1 is viewed as an *observed* variable, and the second component w_2 as a *to-be-deduced* variable. We consider systems of the form $\Sigma = (\mathbb{T}, \mathbb{S}, \mathcal{B})$, where $\mathbb{T} \subseteq \mathbb{R}, \mathbb{S} \subseteq \mathbb{R}^{\mathtt{w}_1 + \mathtt{w}_2}$. Each element of the behavior $\mathcal{B}$ hence consists of a pair of trajectories $(w_1, w_2) : \mathbb{T} \mapsto \mathbb{S}$.

Definition 2.12. (Observability) *Let $\Sigma = (\mathbb{T}, \mathbb{S}, \mathcal{B})$. Assume that trajectories in $\mathcal{B}$ are partitioned as (w_1, w_2) with $w_i : \mathbb{R} \mapsto \mathbb{R}^{\mathtt{w}_i}, i = 1, 2$. The variable w_2 is said to be observable from w_1 whenever*

$$(w_1, w_2'), (w_1, w_2'') \in \mathcal{B} \quad \Longrightarrow \quad w_2' = w_2''. \quad \square$$

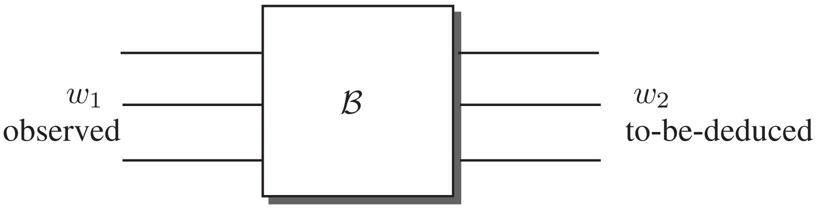


Figure 2.4: Observability

For linear systems, observability of w_2 from w_1 is equivalent to $(0, w_2) \in \mathfrak{B} \implies w_2 = 0$. To get a test for observability, we need to express the behavior in terms of the kernel representation. The following theorem gives the condition for "w_2 observable from w_1" in terms of the representations [PW97].

Theorem 2.5. *Let $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}_1+\mathtt{w}_2}$ be the behavior represented by $R_1\left(\frac{d}{dt}\right) w_1 = R_2\left(\frac{d}{dt}\right) w_2$, where $R_1(\xi) \in \mathbb{R}^{\bullet\times\mathtt{w}_1}[\xi], R_2(\xi) \in \mathbb{R}^{\bullet\times\mathtt{w}_2}[\xi]$. Then, w_2 is observable from w_1 if and only if $\mathtt{rank}(R_2(\lambda)) = \mathtt{w}_2, \forall\lambda \in \mathbb{C}$, equivalently, $R_2(\lambda)$ has full column rank for all $\lambda \in \mathbb{C}$.* □

Observability of a dynamical system, which is represented in the form as given in theorem 2.5, is also equivalent to matrix R_2 having a polynomial left inverse, i.e. there exists a $R_2^{\dagger}(\xi) \in \mathbb{R}^{\bullet\times\bullet}[\xi]$ such that $R_2^{\dagger}R_2 = I$. After computing this $R_2^{\dagger}$, one can obtain w_2, the to-be-deduced variable from w_1, the observed variable. From the above $(w_1, w_2) \in \mathfrak{B}$ implies $w_2 = R_2^{\dagger}\left(\frac{d}{dt}\right) R_1\left(\frac{d}{dt}\right) w_1$. For future reference, whenever the partition of the variable space is understandable, we call the system 'observable'.

Fro a quick demonstration of this result, consider the linear system given in the state-space form

$$\begin{aligned} \frac{d}{dt}x &= Ax + Bu, \\ y &= Cx + Du. \end{aligned}$$

The behavior of this system can be represented by the following kernel representation.

$$\begin{bmatrix} \frac{d}{dt}I - A & B & 0 \\ C & D & I \end{bmatrix} \begin{bmatrix} x \\ u \\ y \end{bmatrix} = 0.$$

Following theorem 2.5, the necessary and sufficient condition for observability of the states from the inputs and outputs is that the matrix $\begin{bmatrix} \lambda I - A \\ C \end{bmatrix}$ should have full rank for all complex numbers, i.e. $\lambda \in \mathbb{C}$. This condition coincides with the renowned Popov-Belevitch-Hautus test for observability.

Relaxing the constraint of full column rank for all $\lambda \in \mathbb{C}$ to full column rank for all $\lambda \in \overline{\mathbb{C}}^{+}$ (the closed right half complex plane) in the last definition and subsequently in the last theorem, we get the concept and a test of detectability, respectively.

Definition 2.13. (Detectability) *Let $\Sigma = (\mathbb{T}, \mathbb{S}, \mathfrak{B})$ be a linear differential system. The trajectories in $\mathfrak{B}$ are partitioned as (w_1, w_2) with $w_i : \mathbb{R} \mapsto \mathbb{R}^{\mathtt{w_i}}, i = 1, 2$. The variable w_2 is said to be detectable from w_1 whenever*

$$(w_1, w_2'), (w_1, w_2'') \in \mathfrak{B} \implies \lim_{t\to\infty} w_2' - w_2'' = 0. \quad \square$$

It is clear from the above that: Observability $\implies$ Detectability but not the otherwise. The notion of detectability is weaker than the notion of observability as the last definition formalizes that we can deduce the to-be-deduced variables from the observed variables *asymptotically.*

Theorem 2.6. *Let $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w_1}+\mathtt{w_2}}$ with system variable (w_1, w_2) be represented by the kernel representation $R_1\left(\frac{d}{dt}\right) w_1 = R_2\left(\frac{d}{dt}\right) w_2$. Then w_2 is detectable from w_1 in $\mathfrak{B}$ if and only if $R_2(\lambda)$ has full column rank for all $\lambda \in \overline{\mathbb{C}}^{+}$.* $\square$

Taking this point of view, the observer design problem within the behavioral framework is investigated in the context of linear shift invariant behaviors in the discrete-time domain in [BVW06], while in the continuous-time domain in [VW99]. Several classical problems addressed for state-space models, like state estimation, the design of unknown input observers or the design of fault detectors and identifiers were casted in this general framework.

2.6 Controllability and stabilizability

The concept of controllability plays a central role in systems and control for the analysis and synthesis of dynamical systems, which deals with modifying the conduct a system exhibit. Let us first recall the implication of the classical definition of controllability, which was introduced and formalized for state-space systems by Kalman in 1960. Consider a state-space equation

$$\frac{d}{dt}x = Ax + Bu$$

where $A \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ and $B \in \mathbb{R}^{\mathtt{n}\times\mathtt{m}}$. The $\mathbb{R}^{\mathtt{n}}$-valued variable x is called the state. This system is said to be *state controllable* if for every $x_0, x_1 \in \mathbb{R}^{\mathtt{n}}$, there exist $T \geq 0$ and $u : \mathbb{R} \mapsto \mathbb{R}^{\mathtt{m}}$ such that the solution $x(t)$ to the above state-space equation with initial state $x(0) = x_0$ satisfies $x(T) = x_1$. In this context, the controllability test is checking that the matrix $\begin{bmatrix} B & AB & A^2B\cdots & A^{n-1}B \end{bmatrix}$

has full row rank. The main drawback of this notion of controllability is being representation dependent. One should realize that a system may be uncontrollable either for the intrinsic reason where the control signal cannot affect the system variables, or because the state was chosen inefficiently. For illustrating the latter case, we consider an example from [Che99] where a dynamical system $\Sigma = (\mathbb{R}, \mathbb{R}^3, \mathcal{B})$ is described by two ordinary differential equations given as

$$\begin{aligned} \frac{d^2y}{dt^2} - 2\frac{dy}{dt} + y &= \frac{du_1}{dt} + u_1, \\ \frac{dy}{dt} - y &= 2u_2. \end{aligned}$$

We are interested in checking the controllability of Σ. Assigning the input / state / output structure to above differential equations, we denote input by, $u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T$, state by x, and output by y. The transformation of these equations results in two representations (RP) in the state-space environment, i.e.,

$$\begin{aligned} (RP:1) \Leftrightarrow \qquad \dot{x} &= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} x + \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} x \\ (RP:2) \Leftrightarrow \qquad \dot{x} &= \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u, \qquad y = \begin{bmatrix} 1 & 2 \end{bmatrix} x \end{aligned}$$

Quickly, we can see that with (RP : 1), the dynamical system is not controllable, however with (RP : 2), it is controllable. Thus, here the controllability depends on a specific representation of the system.

On the other hand, in the behavioral framework, controllability is the property of the system, which is described by the set of trajectories and not of a particular representation of the system [Wil97]. These trajectories are admitted by the system, *the behavior*. We now recall the formal definition of controllability.

Definition 2.14. (Controllability) *Let $\mathcal{B} \in \mathfrak{L}^{\mathrm{s}}$ be the behavior of a dynamical system. The system is said to be* controllable *whenever for all $s_1, s_2 \in \mathcal{B}$, there exists $T \in \mathbb{R}, T \geq 0$ and $s \in \mathcal{B}$ such that*

$$\begin{aligned} s(t) &= s_1(t) \quad \text{for} \quad t < 0, \\ s(t+T) &= s_2(t) \quad \text{for} \quad t \geq 0. \quad \square \end{aligned}$$

The conceptual description of controllability is illustrated in figure 2.5. The above definition refers to the ability to switch from any one trajectory

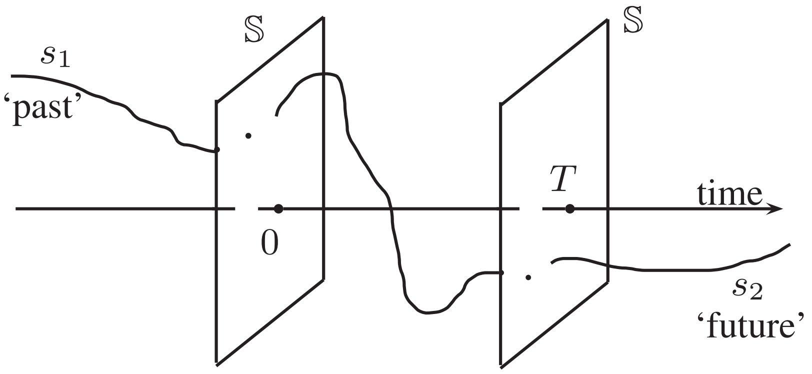


Figure 2.5: Controllability

within the behavior to any other one, allowing some time delay. Of course, if the system happens to be given in terms of a particular representation, we would like to have tests in order to decide whether the system is controllable. For linear differential systems, the following theorem from [PW97] deals in terms of given kernel representations of the behavior.

Theorem 2.7. *Let* $\Sigma = (\mathbb{R}, \mathbb{R}^{\mathtt{s}}, \mathcal{B}) \in \mathfrak{L}^{\mathtt{s}}$. *Let* $R\left(\frac{d}{dt}\right) s = 0$ *be a kernel representation of* $\mathcal{B}$, *with* $R(\xi) \in \mathbb{R}^{\bullet \times \mathtt{s}}[\xi]$. *Then, the following statements are equivalent:*

1. *The system* Σ *is controllable.*
2. *The polynomial matrix* R *has the property that* $\mathtt{rank}(R(\lambda)) = \mathtt{rank}(R)$ *for all* $\lambda \in \mathbb{C}$.
3. *There exists an integer* ℓ *and a polynomial matrix* $M(\xi) \in \mathbb{R}^{\mathtt{s} \times \ell}[\xi]$ *such that* $\mathcal{B}$ *is the image of the differential operator* $M\left(\frac{d}{dt}\right)$. □

A remarkable point of theorem 2.7 is that the controllable system allows a representation in terms of a latent variable ℓ, of the form

$$s = M\left(\frac{d}{dt}\right)\ell \tag{2.10}$$

Equation 2.10 is called an *image representation* of $\mathcal{B}$. Now, reconsider the previous example in the behavioral context. We can describe the behavior of the dynamical system $\Sigma = (\mathbb{R}, \mathbb{R}^3, \mathcal{B})$ by the following kernel representation

$$R\left(\frac{d}{dt}\right) s \quad \Rightarrow \quad \begin{bmatrix} -(\xi+1) & 0 & \xi^2 - 2\xi + 1 \\ 0 & -2 & \xi - 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ y \end{bmatrix} = 0$$

where $R(\xi) \in \mathbb{R}^{2\times 3}[\xi]$, and s is partitioned as, $s = \begin{bmatrix} u_1 & u_2 & y \end{bmatrix}^T$. Following the last theorem, one can easily deduce that the dynamical system is controllable. From this example, we have observed that we do not restrict ourselves to any particular input/state/output structure to determine the controllability of a dynamical system. Rather, we only deal with the variables describing the system. No doubt, (RP : 1) can easily be transformed to (RP : 2) by the so-called *similarity transformation.* However, if the states are chosen randomly, the transformation step unnecessarily increases the computational burden.

Controllability of behaviors will play an important role in this thesis. We shall use this notion while discussing the different modes of the operating system, namely the "desired" and the "undesired" mode. Consider a scenario where our system works in some given mode of operation, considered as the *undesired mode* ('past'). Then, the possibility of transferring our system from that mode to another one, referred to as the *desired mode* ('future'), reflects the ability of controlling the plant. Controllability of a behavior enables us to steer a trajectory to a desired trajectory within some finite time.

We now come to the notion of stabilizability, which is weaker than that of controllability, i.e. "$\mathcal{B}$ is controllable $\implies$ $\mathcal{B}$ is stabilizable" but not the otherwise. This notion is concerned with the situation where we are on a given trajectory of the given behavior $\mathcal{B}$ and we want to switch to a trajectory that *asymptotically* tends to zero, while remaining on a trajectory within the behavior. Following is the definition for linear differential systems.

Definition 2.15. (Stabilizability) $\mathcal{B} \in \mathfrak{L}^{\mathtt{s}}$ *is called* stabilizable *whenever for each $s \in \mathcal{B}$, there exists a $s' \in \mathcal{B}$ such that*

$$\begin{aligned} s'(t) &= s(t) \quad \text{for} \quad t < 0, \\ s'(t) &\to 0 \qquad \text{as} \quad t \to \infty. \quad \square \end{aligned}$$

To test the stabilizability of the given behavior $\mathcal{B}$, we use the following theorem from [PW97] in terms of the given kernel representations.

Theorem 2.8. *Let $\mathcal{B} \in \mathfrak{L}^{\mathtt{s}}$ and let $R(\xi) \in \mathbb{R}^{\bullet\times \mathtt{s}}[\xi]$ induce a kernel representation $R\left(\frac{d}{dt}\right) s = 0$ of $\mathcal{B}$. Then the following statements are equivalent.*

1. *$\mathcal{B}$ is stabilizable.*
2. $\mathtt{rank}(R(\lambda)) = \mathtt{rank}(R)$ *for all* $\lambda \in \overline{\mathbb{C}}^{+}$. □

2.7 Autonomous behavior

Controllability, discussed in the previous section deals with the possibility of moving from a given trajectory to another trajectory within the same behavior. However, for some types of behavior it is not possible. These are termed as the autonomous behaviors. The future of each trajectory within this behavior is completely determined by its past.

Definition 2.16. (Autonomous) *A linear differential system* $\Sigma = (\mathbb{R}, \mathbb{R}^{\mathtt{s}}, \mathcal{B})$ *is said to be* autonomous *whenever for all* $s_1, s_2 \in \mathcal{B}$ *we have*

$$s_1(t) = s_2(t), t \leq 0 \implies s_1 = s_2. \quad \square$$

Now, we would like to have conditions on the underlying polynomial matrices of the behavior to be autonomous. The following theorem from [PW97] effectively relates the property of autonomy to a given kernel representation.

Theorem 2.9. *Let* $\mathcal{B} \in \mathfrak{L}^{\mathtt{s}}$ *and let* $R(\xi) \in \mathbb{R}^{\bullet \times \mathtt{s}}[\xi]$ *induce a kernel representation* $R\left(\frac{d}{dt}\right) s = 0$ *of* $\mathcal{B}$. *Then the following statements are equivalent.*

1. $\mathcal{B}$ *is autonomous.*
2. $\mathtt{rank}(R(\xi)) = \mathtt{s}$, *i.e.* $R(\xi)$ *has full column rank.*
3. $\mathcal{B}$ *is a finite dimensional vector space.* □

Furthermore, if $R\left(\frac{d}{dt}\right) s = 0$ is a minimal kernel representation then any of the above statements is equivalent to R being square and nonsingular. In general, the notion of so-called *characteristic polynomial* is used to describe the set of trajectories in an autonomous behavior. Assume $\mathcal{B} \in \mathfrak{L}^{\mathtt{s}}$ is autonomous and let $R(\xi)$ be a $\mathtt{s} \times \mathtt{s}$ polynomial matrix with $\mathtt{det}(R(\xi)) \neq 0$ such that $\mathcal{B}$ is minimally represented by $R\left(\frac{d}{dt}\right) s = 0$. Now, choose any non-zero $\alpha \in \mathbb{R}$ such that $\mathtt{det}(\alpha R(\xi))$ has the leading coefficient equal to 1. Indeed, for any non-zero $\alpha \in \mathbb{R}$, the polynomial matrix $\alpha R(\xi)$ also yields a kernel representation of $\mathcal{B}$. Denote this monic polynomial by $\chi_{\mathcal{B}}$. This polynomial is termed as the *characteristic polynomial* of $\mathcal{B}$.

One can quickly see that $\chi_{\mathcal{B}}$ depends only on $\mathcal{B}$, and not on the polynomial matrix $R(\xi)$. If $R_1(\xi), R_2(\xi)$ both represent $\mathcal{B}$ minimally then there exists a unimodular $U(\xi)$ such that $R_2(\xi) = U(\xi)R_1(\xi)$. In addition, if $\mathtt{det}(R_1(\xi))$ and $\mathtt{det}(R_2(\xi))$ are monic then $\mathtt{det}(R_1(\xi)) = \mathtt{det}(R_2(\xi))$. In the same context, the roots of the characteristic polynomial are called the *poles* of the autonomous behavior $\mathcal{B}$. Thus, an autonomous behavior $\mathcal{B} \equiv R\left(\frac{d}{dt}\right) s = 0$ is stable if and only if R is Hurwitz, or equivalently, if and only if all its poles are in $\mathbb{C}^-$.

Controllable behaviors and autonomous behaviors, to some extent, are opposite to each other. However, every behavior contains a controllable part and an autonomous part. In fact every behavior can be written as a direct sum of a controllable and an autonomous sub-behavior [PW97].

Theorem 2.10. *Let $R(\xi) \in \mathbb{R}^{\mathtt{g}\times\mathtt{s}}[\xi]$ be a full row rank and let $\mathcal{B}$ be the behavior defined by $R\left(\frac{d}{dt}\right)s = 0$. Then there exist sub-behaviors $\mathcal{B}_{\mathtt{aut}}$ and $\mathcal{B}_{\mathtt{contr}}$ of $\mathcal{B}$ such that*

$$\mathcal{B} = \mathcal{B}_{\mathtt{aut}} \oplus \mathcal{B}_{\mathtt{contr}},$$

where $\mathcal{B}_{\mathtt{contr}}$ is controllable and $\mathcal{B}_{\mathtt{aut}}$ is autonomous, and the characteristic values of $\mathcal{B}_{\mathtt{aut}}$ are exactly those numbers $\lambda \in \mathbb{C}$ for which $\mathtt{rank}(R(\lambda)) < \mathtt{g}$. □

The symbol "$\oplus$″ denotes the direct sum operation. The resulting decomposition of a behavior using the last theorem is, in general, not unique. It is because of the autonomous part which is not unique. However, the controllable part in this decomposition is unique.

2.8 Input/Output Representation

Until now, we have projected the behavioral point of view to deal with dynamical systems as a *representation-free* approach. Thats is, it is not subjected to any special framework to explain the way a system interacts with its environment. Nevertheless, as a particular case, a subset of the system variables may be defined as the inputs while another subset of variables may be fixed as the outputs of the system. In this way, the input/output paradigm becomes (as expected) a special case of this more general setup. Of course, many concepts of the system theory are formalized in the framework of input/output, particularly in feedback control. Thus, it would be interesting to introduce the link between the two points of view.

The behavioral approach takes into account the possibility of having unconstrained system variables, termed as the *free variables*. The underlying idea of these variables is that the system cannot impose any restriction on them and hence is chosen/fixed by the environment. Such variables will be called *inputs* of the system. In general, there are more system variables than equations describing the behavior of the system. As a consequence, a polynomial matrix that represents the system is non-square. This is the main cause for the existence of free variables which are unconstrained and hence, they are labeled as inputs. Once the inputs are fixed, together with the initial conditions they will determine the values of the remaining variables of the system. These remaining variables are then called the outputs. The following definition from [PW97] summarizes these ideas.

Definition 2.17. (Defining inputs and outputs) *Let $\Sigma = (\mathbb{R}, \mathbb{R}^{\mathtt{s}}, \mathcal{B})$ be a linear differential system. Partition the signal space as $\mathbb{R}^{\mathtt{s}} = \mathbb{R}^{\mathtt{w}_1} \times \mathbb{R}^{\mathtt{w}_2}$ and partition s correspondingly as $s = (w_1, w_2)$. This partition is said to be the* input/output (i/o) partition *whenever:*

1. *w_1 is free, i.e., for all $w_1 \in \mathfrak{C}^\infty(\mathbb{R}, \mathbb{R}^{\mathtt{w}_1})$, there exists a $w_2 \in \mathfrak{C}^\infty(\mathbb{R}, \mathbb{R}^{\mathtt{w}_1})$ such that $(w_1, w_2) \in \mathcal{B}$.*

2. *w_2 does not contain any further free components, i.e., given w_1, none of the components of w_2 can be chosen freely.*

If these conditions hold then we also say: w_1 is maximally free. If both conditions above are satisfied, then w_1 is called an input variable and w_2 is called an output variable. □

If w_1 is maximally free then w_2 does not contain any free components. In this case, w_2 is often called *bound.* The following theorem provides conditions in terms of given kernel representation of a dynamical system for the input/output partition in s.

Theorem 2.11. *Let $R(\xi) \in \mathbb{R}^{\mathtt{g}\times\mathtt{s}}[\xi]$ induce a kernel representation of $\mathcal{B}$. Let $s = (w_1, w_2)$ be the partition of s and let $R(\xi) = \begin{bmatrix} R_1(\xi) & R_2(\xi) \end{bmatrix}$ be the corresponding partition of $R(\xi)$. Then,*

1. *w_1 is free if and only if $\mathtt{rank}(\begin{bmatrix} R_1(\xi) & R_2(\xi) \end{bmatrix}) = \mathtt{rank}(R_2(\xi))$,*

2. *once w_1 is fixed, w_2 is a bound if and only if $R_2(\xi)$ has full column rank, i.e. $\mathtt{rank}(R_2(\xi)) = \mathtt{dim}(w_2)$,*

3. *$s = (w_1, w_2)$ is the input/output partition if and only if $\mathtt{rank}(R(\xi)) = \mathtt{rank}(R_2(\xi)) = \mathtt{coldim}(R_2(\xi))$.* □

According to the above theorem, if $R\left(\frac{d}{dt}\right) s = 0$ is a minimal kernel representation, then $s = (w_1, w_2)$ is an i/o partition if and only if $R_2(\xi)$ is square and nonsingular. With this partition, the system can be written as $R_1\left(\frac{d}{dt}\right) w_1 + R_2\left(\frac{d}{dt}\right) w_2 = 0$, and owing to the i/o partition, the matrix $-R_2^{-1}(\xi)R_1(\xi)$ defines the *transfer matrix* of $\mathcal{B}$. For linear differential systems, the transfer matrix is rational, i.e. each entry in this matrix is a ratio of two polynomials. Coming on to the issue of properness, a rational matrix is called *proper* if in each entry the degree of the numerator does not exceed the degree of the denominator. Further, the rational matrix is called *strictly* proper if in each entry the degree of the numerator is strictly less than the degree of the denominator. When we consider only $\mathfrak{C}^\infty$ trajectories, properness is not an issue and in general, the transfer matrix $-R_2^{-1}(\xi)R_1(\xi)$ is not proper. Properness becomes important when we talk about linear differential systems considering $\mathcal{L}_2$ trajectories. The variable u being an input in the $\mathcal{L}_2$ sense is equivalent to the rational matrix $-R_2^{-1}(\xi)R_1(\xi)$ being proper. In this regard, the following method of partitioning R into $R = \begin{bmatrix} R_1 & R_2 \end{bmatrix}$ ensures the properness condition on $-R_2^{-1}(\xi)R_1(\xi)$. In the following example, we will show how an input/output partition occurs naturally in this framework.

Example 2.6. *Consider a simple electrical RCL (Resistor, Capacitor, Inductor) network with the current i that flows into the network, and the voltage v across the network. Therefore, (i, v) serves as the manifest variables of the system. We choose the auxiliary variables as the currents i_R, i_L and i_C passes through the resistor, inductor and capacitor, respectively, the voltages v_R, v_L and v_C across the network. Thus, $(i_R, i_L, i_C, v_R, v_L, v_C)$ serves the latent variables. The equations describing the behavior of the system are given as*

$$\begin{array}{ccc} v_R = Ri_R, & v_L = L\frac{d}{dt}i, & i_C = C\frac{d}{dt}v_C \\ i = i_R, & i_R = i_L, & i_L = i_C \\ & v = v_R + v_L + v_C & \end{array}$$

All these equations can be combined in the form

$$S\left(\frac{d}{dt}\right)s = 0$$

where the polynomial matrix S and the vector-valued signal s are given by

$$S(\xi) = \begin{bmatrix} R & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & \xi L & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -\xi C & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 & 1 \end{bmatrix}, \qquad s = \begin{bmatrix} i_R \\ i_L \\ i_C \\ v_R \\ v_L \\ v_C \\ i \\ v \end{bmatrix}$$

The full behavior of the system is then described as

$$\mathcal{B}_{\texttt{full}} = \left\{ s \in (\mathbb{R}^{\mathtt{s}})^{\mathbb{R}} | S\left(\frac{d}{dt}\right) s = 0 \right\}$$

For the network, the latent variables ℓ and the manifest variables w are chosen as $\ell = \begin{bmatrix} i_R & i_L & i_C & v_R & v_L & v_C \end{bmatrix}^T, w = \begin{bmatrix} i & v \end{bmatrix}^T$. Using the elimination theorem 2.3, we can obtain the manifest behavior of the system, described as

$$\mathcal{B} = \{ w \in (\mathbb{R}^{\mathtt{w}})^{\mathbb{R}} | \exists \ell \text{ such that } (w, \ell) \in \mathcal{B}_{\texttt{full}} \}$$

Eliminating ℓ from the kernel representation of the full system behavior, we obtain the kernel representation of the manifest system behavior in w given as

$$R\left(\frac{d}{dt}\right) w = 0 \quad \equiv \begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} i \\ v \end{bmatrix} = 0.$$

where $R_1(\xi) = LC\xi^2 + RC\xi + 1$ and $R_2(\xi) = C\xi$. Checking the properness condition of the above matrix, it yields that $R_1^{-1}R_2$ is proper. As a consequence, the electrical network can be connected to a voltage source, while the current serves as the output. □

Inputs and outputs determine a set of invariants linked to a given behavior $\mathcal{B}$, see [PW97]. These invariants are defined as follows:

$\mathtt{s}(\mathcal{B})$: the number of components of the system variable s of $\mathcal{B}$.

$\mathtt{m}(\mathcal{B})$: the number of input variables in any input/output partition of the system variables. This number is called the *input cardinality* of $\mathcal{B}$.

$\mathtt{p}(\mathcal{B})$: the number of outputs in any input/output partition of the system variable s. This number is called the *output cardinality* of $\mathcal{B}$. Obviously, $\mathtt{s}(\mathcal{B}) = \mathtt{p}(\mathcal{B}) + \mathtt{m}(\mathcal{B})$.

$\mathtt{n}(\mathcal{B})$: the McMillan degree of $\mathcal{B}$. Suppose $R\left(\frac{d}{dt}\right) s = 0$ is a kernel representation of $\mathcal{B}$ then the McMillan degree of $\mathcal{B}$ is equal to the maximal degree minor of $R(\xi)$.

Having explained about input/output partitions, for $\mathcal{B} \in \mathfrak{L}^{\mathtt{s}}$ given by a kernel representation $R\left(\frac{d}{dt}\right) = 0$, we have the output cardinality $\mathtt{p}(\mathcal{B})$ of $\mathcal{B}$ is equal to $\mathtt{rank}(R)$. However, when $\mathcal{B}$ is autonomous $\mathtt{rank}(R) = \mathtt{p}(\mathcal{B}) = \mathtt{s}$ and hence there is no input.

2.9 Input/State/Output Representation

State variables either show up naturally or can be artificially introduced during the modeling process. The salient feature is that they parametrize the "memory" of a dynamical system. Thats is they "split" the past and future of the behavior and while going from the past into the future, one only needs to see that the states match. In this way, the value of the states at a particular instant can be thought as capturing the entire history of evolution of a system up to that instant. In the behavioral framework, the state x of a system is regarded as a latent variable. In example 2.5, we have shown how the states can be eliminated.

Definition 2.18. *Let $\Sigma_{\mathtt{full}} = (\mathbb{R}, \mathbb{R}^{\mathtt{s}}, \mathbb{R}^{\mathtt{x}}, \mathcal{B}_{\mathtt{full}})$ be a time invariant latent variable system. The latent variable x is said to have the property of state whenever*

$$\begin{aligned}\{(s_1, x_1), (s_2, x_2) \in \mathcal{B}_{\mathtt{full}}\} \ &\textit{and}\ \{x_1(t_0) = x_2(t_0)\}\\ &\textit{and}\ \{x_1, x_2\ \textit{continuous at}\ t = t_0\}\\ &\Longrightarrow\ \{(s_1 \bigtriangleup^{t_0} s_2, x_1 \bigtriangleup^{t_0} x_2) \in \mathcal{B}_{\mathtt{full}}\}.\end{aligned}$$

Recall that the symbol $\triangle^{t_0}$ *denotes the concatenation operation at* $t = t_0$*.* □

According to the above definition, the above latent variable system in which the latent variable has the property of state will be called *state systems.*

Theorem 2.12. *Let* $\Sigma_{\mathtt{full}} = (\mathbb{R}, \mathbb{R}^{\mathtt{s}}, \mathbb{R}^{\mathtt{x}}, \mathcal{B}_{\mathtt{full}})$ *be a linear differential system with latent variable* x *taking values in* $\mathbb{R}^{\mathtt{x}}$ *and* $s = (u, y)$*. Then* $\Sigma_{\mathtt{full}}$ *is a state system if and only if there exist matrices* $A, B, C, D \in \mathbb{R}^{\bullet\times\bullet}$ *such that*

$$\mathcal{B}_{\mathtt{full}} = \left\{ (u, y, x) \,\middle|\, \frac{d}{dt}x = Ax + Bu, y = Cx + Du \right\}. \quad \square$$

Here $u \in (\mathbb{R}^{\mathtt{m}(\mathcal{B}_{\mathtt{full}})})^{\mathbb{R}}$ is the input, $x \in (\mathbb{R}^{\mathtt{n}(\mathcal{B}_{\mathtt{full}})})^{\mathbb{R}}$ is the state, and $y \in (\mathbb{R}^{\mathtt{p}(\mathcal{B}_{\mathtt{full}})})^{\mathbb{R}}$ is the output.

Part II

Novel active fault-tolerant control approaches

Chapter 3

Interconnection of Behaviors

Contents

3.1 Interconnection of dynamical systems 61

3.2 Regular Interconnection 63

3.3 Implementability 66

3.3.1 Full interconnection 67

3.3.2 Partial interconnection case 68

In chapter 2, we have covered the preliminary concepts of the behavioral theory. Here, we will use those concepts to demonstrate the interconnection of dynamical systems. As a matter of fact, in the behavioral framework, the control problem is treated as the interconnection of two dynamical systems, namely the plant, and the controller. Controlling a plant is nothing but restricting its behavior to a desired subset of the behavior. This restriction is brought about by interconnecting the plant with the controller. In the interconnected system, the plant variables have to obey the laws that are imposed by the plant itself and the controller. This interconnected system is called the controlled system, in which the controller is an embedded system. In this chapter, we shall study various concepts of control in the behavioral framework starting from the viewpoint of “control via system interconnection”.

This chapter has been added in this part of the thesis because this chapter would be very helpful when we discuss the interconnection of behaviors within the FTC approaches. In addition, the concepts presented in this chapter are common to the following two chapters.

3.1 Interconnection of dynamical systems

The concept of interconnection plays the central role in modeling and control of systems in the behavioral framework. By an interconnected system, we mean a system that consists of interacting subsystems [Wil07]. Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be the behaviors of linear differential systems. To interconnect these systems, there must exist some common variables. Then the interconnection of $\mathcal{B}_1$ and

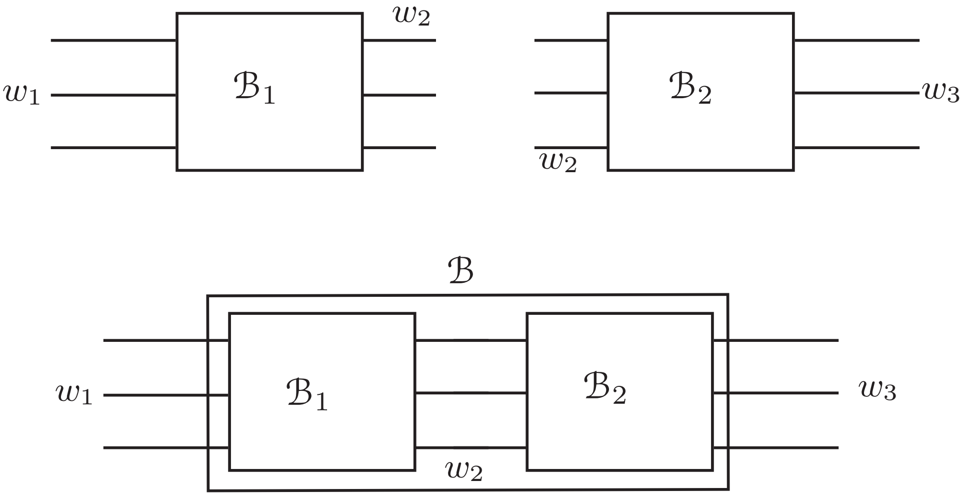


Figure 3.1: Interconnection of systems

$\mathcal{B}_2$ through these common (or shared) variables results in a system in which these variables satisfy the dynamics of both $\mathcal{B}_1$ and $\mathcal{B}_2$. Depending on the way the shared variables interconnect between two systems, there exist two types of interconnection. In the first case, all system variables in $\mathcal{B}_1$ and $\mathcal{B}_2$ are common and the interconnection takes place through all these variables. This is called full interconnection. The second case is when $\mathcal{B}_1$ and $\mathcal{B}_2$ have only a few variables in common and they are interconnected through these variables only. This is called partial interconnection. In terms of behavioral description, these concepts are formalized below.

Definition 3.1. (Full Interconnection) *Given two behaviors $\mathcal{B}_1 \in \mathfrak{L}^{\mathtt{s}}$ and $\mathcal{B}_2 \in \mathfrak{L}^{\mathtt{s}}$ with system variable s, the full interconnection between $\mathcal{B}_1$ and $\mathcal{B}_2$ is denoted by $\mathcal{B}_1 \cap \mathcal{B}_2$, and is defined as*

$$\mathcal{B}_1 \cap \mathcal{B}_2 = \{s | s \in \mathcal{B}_1 \textit{ and } s \in \mathcal{B}_2\}. \quad \square$$

Definition 3.2. (Partial Interconnection) *Given two behaviors $\mathcal{B}_1 \in \mathfrak{L}^{\mathtt{w}_1+\mathtt{w}_2}$ and $\mathcal{B}_2 \in \mathfrak{L}^{\mathtt{w}_2+\mathtt{w}_3}$ with system variable $s = (w_1, w_2, w_3)$, the partial interconnection between $\mathcal{B}_1$ and $\mathcal{B}_2$ is denoted by $(\mathcal{B}_1 \wedge_{w_2} \mathcal{B}_2)_{\mathtt{full}}$, and is defined as*

$$(\mathcal{B}_1 \wedge_{w_2} \mathcal{B}_2)_{\mathtt{full}} = \{(w_1, w_2, w_3) | (w_1, w_2) \in \mathcal{B}_1 \textit{ and } (w_2, w_3) \in \mathcal{B}_2\}. \quad \square$$

The shared variable w_2 in definition 3.2 is referred to as the *interconnection variable*. In the sequel, whenever the interconnection variable is cleared from the context, we omit that variable from the symbol, i.e., we will write $\mathcal{B}_1 \wedge \mathcal{B}_2$ simply. In the last definition, the behavior of an interconnected system is described by all system variables, i.e. (w_1, w_2, w_3) trajectories. That is why, we termed it as the full behavior. Sometimes, after the interconnection, the interconnecting variables can be considered as the latent variables, and the main interest lies in the remaining variables after eliminating the latent variables. Here, the latent variable is w_2, and (w_1, w_3) are the remaining variables (see figure 3.1). Thus, the behavior, after the elimination of latent variable, is given by

$$\mathcal{B}_1 \wedge \mathcal{B}_2 = \{(w_1, w_3) \mid \exists w_2 \text{ such that } (w_1, w_2) \in \mathcal{B}_1 \text{ and } (w_2, w_3) \in \mathcal{B}_2\}.$$

Now we consider the interconnection of systems in terms of kernel representations. Given the kernel representation of $\Sigma_1 = (\mathbb{T}, \mathbb{S}, \mathcal{B}_1)$ by $S_1\left(\frac{d}{dt}\right) s = 0$, and similarly, of $\Sigma_2 = (\mathbb{T}, \mathbb{S}, \mathcal{B}_2)$ by $S_2\left(\frac{d}{dt}\right) s = 0$. Then the full interconnection $\mathcal{B}_1 \cap \mathcal{B}_2$ is represented by

$$\begin{bmatrix} S_1\left(\frac{d}{dt}\right) \\ S_2\left(\frac{d}{dt}\right) \end{bmatrix} s = 0.$$

In case of partial interconnection, suppose the kernel representation of $\Sigma_1 = (\mathbb{T}, \mathbb{S}, \mathcal{B}_1)$ is given by $R_1\left(\frac{d}{dt}\right) w_1 + R_2\left(\frac{d}{dt}\right) w_2 = 0$, and similarly, of $\Sigma_2 = (\mathbb{T}, \mathbb{S}, \mathcal{B}_2)$ by $Q_1\left(\frac{d}{dt}\right) w_2 + Q_2\left(\frac{d}{dt}\right) w_3 = 0$. Then the partial interconnection $(\mathcal{B}_1 \wedge \mathcal{B}_2)_{\texttt{full}}$ is represented by

$$\begin{bmatrix} R_1\left(\frac{d}{dt}\right) & R_2\left(\frac{d}{dt}\right) & 0 \\ 0 & Q_1\left(\frac{d}{dt}\right) & Q_2\left(\frac{d}{dt}\right) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = 0.$$

3.2 Regular Interconnection

Making an interconnection to a dynamical system by another dynamical system is all about imposing restriction in some sense on the system we have. Consider the following example.

Example 3.1. *Let $\mathcal{B}_1$ be the behavior, on which the restrictions has to be imposed and it is given by*

$$\mathcal{B}_1 = \{s | \frac{d^2}{dt^2} s - s = 0\}.$$

Take another behavior $\mathcal{B}_2$ which is described by

$$\mathcal{B}_2 = \{s | \frac{d}{dt} s + s = 0\}.$$

The behavior $\mathcal{B}_2$ has been designed such that all trajectories in $\mathcal{B}_1 \cap \mathcal{B}_2$ are stable (i.e. $\lim\limits_{t \to \infty} s(t) = 0$). Here, we see the trajectory in $\mathcal{B}_1$ is completely *characterized by its past. Moreover, this behavior has unstable exponential trajectories (the trajectories that are not bounded as $t \to \infty$). Thus, $\mathcal{B}_1$ depicts the unstable behavior in the sense of [PW97, Definition 7.2.1]. The interconnection of $\mathcal{B}_1$ and $\mathcal{B}_2$ yields the behavior $\mathcal{B}_1 \cap \mathcal{B}_2 = \mathcal{B}$ which is described by*

$$\mathcal{B} = \{s | \frac{d}{dt} s + s = 0\}.$$

Consequently, all trajectories in $\mathcal{B}$ are stable (i.e., $\lim_{t\to\infty} s(t) = 0$. □

In the above example, notice that the unstable trajectories in $\mathcal{B}_1$ do not belong to the interconnected behavior $\mathcal{B}$. From a viewpoint of the system theory, such dynamical systems are impossible to implement. The prime cause of this is the laws which are already present in $\mathcal{B}_1$ are forced to be repeated in $\mathcal{B}_2$. Therefore, if we do not add any further restrictions on the interconnection, it is perfectly possible that an unstable behavior is stabilized by the so-called irregular interconnection. Such situations can be avoided if the interconnection between $\mathcal{B}_1$ and $\mathcal{B}_2$ is regular. We use module-theoretic properties of behaviors to demonstrate the concept of regular interconnection [RW01]. Recall from the last chapter, when we write "$\mathcal{B} \subseteq \mathbb{S}^{\mathbb{T}}$", we implicitly assumed not only that $\mathbb{S}$ is one of the listed signal space but also that $\mathcal{B}$ can be described by differential equations with constant real coefficients. Given a differential behavior $\mathcal{B} \subseteq \mathbb{S}^{\mathbb{T}}$, it will sometimes be necessary to refer to the sub-modules $\mathfrak{M}$ of $\mathbb{R}^{1\times \mathtt{s}}[\xi]$, defined by

$$\mathfrak{M}(\mathcal{B}) = \left\{ v(\xi) \in \mathbb{R}^{1\times \mathtt{s}}[\xi] \middle| v\left(\frac{d}{dt}\right) s = 0 \; \forall s \in \mathcal{B} \right\} \tag{3.1}$$

where the meaning of vs is simply the $1 \times \mathtt{s}$ polynomial matrix v applied to the trajectory s in the usual way. Thus, $\mathfrak{M}(\mathcal{B})$ is the set of all polynomial equations satisfied by the behavior. There exists one-to-one correspondence between linear differential behavior and submodules. Thus,

$$\mathcal{B}(\mathfrak{M}) = \left\{ s \in (\mathbb{R}^{\mathtt{s}})^{\mathbb{R}} \middle| v\left(\frac{d}{dt}\right) s = 0 \text{ for all } v \in \mathfrak{M} \right\}. \tag{3.2}$$

This implies that if $\mathcal{B} = \mathtt{ker}\left(S\left(\frac{d}{dt}\right)\right)$ then $\mathfrak{M}(\mathcal{B})$ is the submodule of $\mathbb{R}^{1\times \mathtt{s}}[\xi]$ generated by the rows of S. The concept of regular interconnection is formalized below.

Definition 3.3. (Regular Interconnection) *The interconnection between $\mathcal{B}_1$ and $\mathcal{B}_2$ is said to be a regular interconnection if the sets $\mathfrak{M}(\mathcal{B}_1)$ and $\mathfrak{M}(\mathcal{B}_2)$ of system equations intersect trivially.* □

With this definition introduced in [RW01], regular interconnection expresses the idea of "restricting what is not yet restricted". According to this definition, let $\mathcal{B}_1 \subseteq \mathbb{S}^{\mathbb{T}}$, $\mathcal{B}_2 \subseteq \mathbb{S}^{\mathbb{T}}$, and the submodule of these behaviors are given by $\mathfrak{M}(\mathcal{B}_1) \subseteq \mathbb{R}^{1\times \mathtt{s}}[\xi]$ and $\mathfrak{M}(\mathcal{B}_2) \subseteq \mathbb{R}^{1\times \mathtt{s}}[\xi]$, then the interconnection of $\mathcal{B}_1$ and $\mathcal{B}_2$ is called regular if

$$\mathfrak{M}(\mathcal{B}_1) \cap \mathfrak{M}(\mathcal{B}_2) = \{0\}. \tag{3.3}$$

Equation (3.3) indicates that in the regular interconnection, a dynamical system is supposed to impose new restrictions on another dynamical system rather than re-imposing restrictions that were already present. In this sense, a regular system is regarded as a "non-redundant" system. The following theorem shows the relation between the regularity of interconnection and the output cardinalities of the behaviors involved in interconnection [BT02].

Theorem 3.1. *Let $\mathcal{B}_1 \in \mathfrak{L}^{\mathtt{w}_1+\mathtt{w}_2}$ and $\mathcal{B}_2 \in \mathfrak{L}^{\mathtt{w}_2+\mathtt{w}_3}$ with system variable (w_1, w_2) and (w_2, w_3) respectively. Then the following statements are equivalent.*

1. *The interconnection $(\mathcal{B}_1 \wedge_{w_2} \mathcal{B}_2)_{\mathtt{full}}$ is regular.*
2. $\mathtt{p}((\mathcal{B}_1 \wedge_{w_2} \mathcal{B}_2)_{\mathtt{full}}) = \mathtt{p}(\mathcal{B}_1) + \mathtt{p}(\mathcal{B}_2)$. □

In the case of full interconnection, the regularity of interconnection is amounting to $\mathtt{p}(\mathcal{B}_1 \cap \mathcal{B}_2) = \mathtt{p}(\mathcal{B}_1) + \mathtt{p}(\mathcal{B}_2)$. In [Wil97], the problems of pole placement and stabilization are discussed for the case of full interconnection. It is a fact that for a pole placement problem, the controllability of the plant plays an important role. Interestingly in [Wil97], particularly for the full interconnection, pole placement is guaranteed if only if the plant behavior is controllable, *assuming the interconnection of the plant and the controller is regular*. Later it is shown that the stabilizability of the plant is equivalent to the existence of a stabilizing controller, again provided the *interconnection is regular*.

It turns out that the regularity on interconnection between two dynamical systems can be guaranteed whenever these two systems interconnect in the so-called "feedback configuration". In the classical sense, the feedback dynamical system is described as feeding back the sensor (or the output) signals of another dynamical system suitably into the actuator inputs. Stating precisely, we want to attach a controller to a plant such that the controller takes the measured output of the plant as the inputs, vice-versa the controller outputs are fed back into the control inputs of the plant. Thus, the i/o (input/output) partition of a controller is now fixed by the plant. Let $\mathcal{B} \in \mathfrak{L}^{\mathtt{s}}$ and let $R(\frac{d}{dt})s = 0$ be a minimal kernel representation. Then there exists a partition $s = (u, y)$ (after perhaps a permutation of the components within

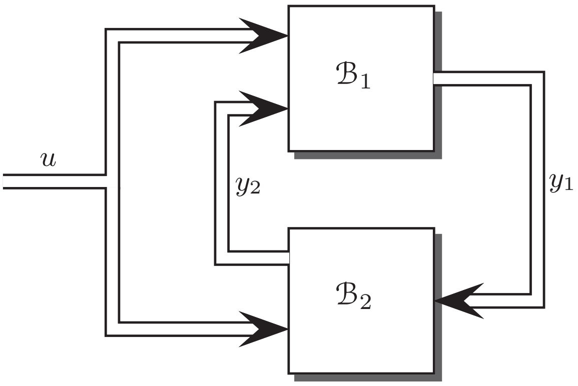


Figure 3.2: Feedback interconnection

s) such that $\mathcal{B}$ is represented minimally by $P(\frac{d}{dt})y = Q(\frac{d}{dt})u$. The variables in $s = (u, y)$ is an i/o partition if and only if $\det(P) \neq 0$. In addition, if the rational matrix $Q^{-1}(\xi)P(\xi)$ is proper with $Q(\xi)$ being square matrix, then this partition is called 'proper input-output partition.' We now recall the definition of feedback interconnection as defined in [Wil97] for the case of full interconnection (see figure 3.2).

Definition 3.4. (Full feedback interconnection) *The interconnection of $\mathcal{B}_1$ and $\mathcal{B}_2 \in \mathfrak{L}^{\mathtt{s}}$ is said to be a* feedback interconnection *if, after permutation of components, there exists a partition of s into $s = (u, y_1, y_2)$ such that*

1. *in $\mathcal{B}_1, (u, y_2)$ is input and y_1 output,*

2. *in $\mathcal{B}_2, (u, y_1)$ is input and y_2 output, and*

3. *in $\mathcal{B}_1 \cap \mathcal{B}_2, u$ is input and (y_1, y_2) output.* □

From the standpoint of practical implementation of the dynamical systems, we shall always consider the feedback interconnections in the sequel. Therefore, whenever we discuss about the interconnection between the plant and the controller, they will always be treated as the regular (or feedback) interconnections.

3.3 Implementability

Implementability deals with an issue that which system behaviors can be achieved (or 'implemented') by interconnecting a given system behavior with another. It may be considered as the scenario where a behavior is prescribed, and the question is whether this "desired" behavior can be achieved by inserting a suitably designed subsystem into the over-all system. We shall discuss,

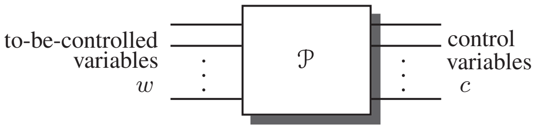


Figure 3.3: Plant

in detail, about the desired behavior in the later chapter. At the moment, we are treating a controller as a dynamical system which is allowed to impose restrictions on another dynamical system called a plant through a subset of the plant variables. So our interest lies in that which behaviors are plausible with this interconnection. The plant we wish to control consists of two types of variables: the *to-be-controlled* variables (denoted by w) and the *control* variables (denoted by c), where w are the variables whose trajectories we aim to influence. The influence is governed by the imposed requirements on these variables. Any type of influence on w is allowed by interconnecting a controller to just the control variables c.

Figure 3.3 illustrates the plant as a dynamical system and Figure 3.4 illustrates the controller. In the classical feedback control problem, the variables that can be measured and/or actuated upon play the role of control variables. Often, of course, there are some common components in w and c. However, sometimes we do not separate these variables, i.e., the case when $w = c$. This corresponds to the full interconnection case (as distinguished from the figure 3.3). It is considered to be a very special case since the controller is attached directly to the (manifest) variables w. Indeed, the case when $w \neq c$ corresponds to the partial interconnection case. In this section, we study which behaviors can be implemented both in the full and the partial interconnection case.

3.3.1 Full interconnection

In the case of full interconnection, take the plant behavior as $\mathcal{P} \in \mathfrak{L}^{\mathsf{s}}$, and a controller that makes an interconnection with $\mathcal{P}$ is given by the behavior $\mathcal{C} \in \mathfrak{L}^{\mathsf{s}}$. The interconnection of $\mathcal{P}$ and $\mathcal{C}$ is the system whose behavior is denoted by $\mathcal{P} \cap \mathcal{C} \equiv \mathcal{K}$. This interconnected behavior is termed as the controlled behavior or the *implemented behavior* $\mathcal{K} = \mathcal{P} \cap \mathcal{C} \in \mathfrak{L}^{\mathsf{s}}$. In terms of given kernel representations, if $\mathcal{P} = \mathtt{ker}\left(R\left(\frac{d}{dt}\right)\right)$ and $\mathcal{C} = \mathtt{ker}\left(C\left(\frac{d}{dt}\right)\right)$, then

$$\mathcal{K} = \mathtt{ker}\left(\begin{bmatrix} R\left(\frac{d}{dt}\right) \\ C\left(\frac{d}{dt}\right) \end{bmatrix}\right). \tag{3.4}$$

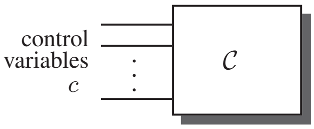


Figure 3.4: Controller

Definition 3.5. (Implementability: full interconnection) *Let $\mathcal{P} \in \mathfrak{L}^{\mathtt{s}}$ be a linear differential system, $\mathcal{C} \in \mathfrak{L}^{\mathtt{s}}$ be a controller, and $\mathcal{K} \in \mathfrak{L}^{\mathtt{s}}$. Whenever $\mathcal{K}$ is obtained by interconnecting $\mathcal{P}$ and $\mathcal{C}$, then we say that "$\mathcal{C}$ implements $\mathcal{K}$". In addition, for a given $\mathcal{K} \in \mathfrak{L}^{\mathtt{s}}$ whenever there exists $\mathcal{C} \in \mathfrak{L}^{\mathtt{s}}$ such that $\mathcal{C}$ implements $\mathcal{K}$, then $\mathcal{K}$ is said to be implementable by full interconnection.* □

Since $\mathcal{K}$ is the restricted behavior, a given $\mathcal{K} \in \mathfrak{L}^{\mathtt{s}}$ is implementable by full interconnection with respect to $\mathcal{P}$ if and only if $\mathcal{K} \subseteq \mathcal{P}$. Using the theorem 2.1, we have the following results in terms of kernel representations.

Theorem 3.2. *Let $\mathcal{P} \in \mathfrak{L}^{\mathtt{s}}$ and $\mathcal{K} \in \mathfrak{L}^{\mathtt{s}}$. Let $\mathcal{P} = \mathtt{ker}\left(R\left(\frac{d}{dt}\right)\right)$ and $\mathcal{K} = \mathtt{ker}\left(K\left(\frac{d}{dt}\right)\right)$ be kernel representations. Then the following statements are equivalent.*

1. *$\mathcal{K}$ is implementable with respect to $\mathcal{P}$ by full interconnection.*
2. *There exists a polynomial matrix $F \in \mathbb{R}^{\bullet\times\bullet}$ with $F(\lambda)$ full row rank for all $\lambda \in \mathbb{C}$ such that $R = FK$.* □

3.3.2 Partial interconnection case

Unlike to the last subsection, in the case of partial interconnection, only a pre-specified subset of the plant variables is available for the interconnection. The implementability in this case is considered to be a more generic case. As mentioned earlier, here the system variables are often partitioned into manifest variables w, and control variables c. Let $\mathcal{P} \in \mathbb{R}^{\mathtt{w}+\mathtt{c}}$ be a linear differential system, with system variable (w, c), where w takes its values in $\mathbb{R}^{\mathtt{w}}$ and c in $\mathbb{R}^{\mathtt{c}}$. Before coming onto the interconnection, we shall introduce two behaviors of the plant that are relevant in the following discussion, namely the full plant behavior $\mathcal{P} \in \mathfrak{L}^{\mathtt{w}+\mathtt{c}}$ of the variables w and c, and the manifest behavior $(\mathcal{P})_w$ of the manifest variables w (with the interconnection variable c eliminated). Consider,

$$\mathcal{P} = \left\{(w, c) \in (\mathbb{R}^{\mathtt{w}+\mathtt{c}})^{\mathbb{R}} \middle| R\left(\frac{d}{dt}\right) w = M\left(\frac{d}{dt}\right) c\right\} \tag{3.5}$$

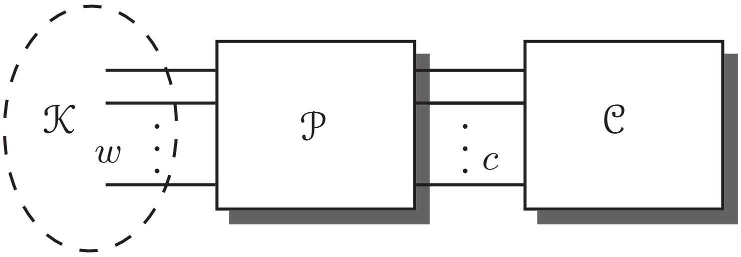


Figure 3.5: The plant and controller after interconnection

with $R(\xi) \in \mathbb{R}^{\bullet\times \mathtt{w}}[\xi], M(\xi) \in \mathbb{R}^{\bullet\times \mathtt{c}}[\xi]$. Then using the elimination theorem, we can get

$$(\mathcal{P})_w = \{w \in (\mathbb{R}^{\mathtt{w}})^{\mathbb{R}} | \exists c \in (\mathbb{R}^{\mathtt{c}})^{\mathbb{R}} \text{ such that } (w, c) \in \mathcal{P}\}. \tag{3.6}$$

Indeed, $(\mathcal{P})_w \in \mathfrak{L}^{\mathtt{w}}$. Let $\mathcal{C} \in \mathfrak{L}^{\mathtt{c}}$ is described by

$$\mathcal{C} = \left\{ c \in (\mathbb{R}^{\mathtt{c}})^{\mathbb{R}} \middle| C\left(\frac{d}{dt}\right) c = 0 \right\} \tag{3.7}$$

with $C(\xi) \in \mathbb{R}^{\bullet\times \mathtt{c}}[\xi]$. A controller $\mathcal{C}$ restricts the trajectories in a plant behavior that c can assume. Once we have defined the plant and the controller as a two separate system, further, when the system interacts, we obtain a new system in which c satisfies both the laws of the plant, and the laws imposed by the controller. Thus, c is governed by $\mathcal{P}$ and $\mathcal{C}$. This brings us to the notion of a full controlled behavior $\mathcal{K}_{\mathtt{full}}$, which is obtained by the interconnection of $\mathcal{P}$ and $\mathcal{C}$ through the variable c, denoted by $\mathcal{K}_{\mathtt{full}} = (\mathcal{P} \wedge_c \mathcal{C})_{\mathtt{full}}$. It is defined as

$$\mathcal{K}_{\mathtt{full}} = \{(w, c) | (w, c) \in \mathcal{P} \text{ and } c \in \mathcal{C}\}.$$

As mentioned before, after an interconnection, the control variable is often considered as the latent variable. We might get rid of this variable by applying the elimination theorem. This, then defines the manifest controlled behavior or simply, the controlled behavior (see figure 3.5), described as follows:

$$\mathcal{K} = \{w \in (\mathbb{R}^{\mathtt{w}})^{\mathbb{R}} \mid \exists c \text{ such that } (w, c) \in \mathcal{K}_{\mathtt{full}}\}$$

or,

$$\mathcal{K} = \{w \in (\mathbb{R}^{\mathtt{w}})^{\mathbb{R}} | \exists c \in \mathcal{C} \text{ such that } (w, c) \in \mathcal{P}\}. \tag{3.8}$$

Recalling again the elimination theorem, we have $\mathcal{K} = \mathcal{P} \wedge_c \mathcal{C} \in \mathfrak{L}^{\mathtt{w}}$. The relationship between the full plant behavior, the manifest behavior, the controller and the controlled behavior are captured in figure 3.6.

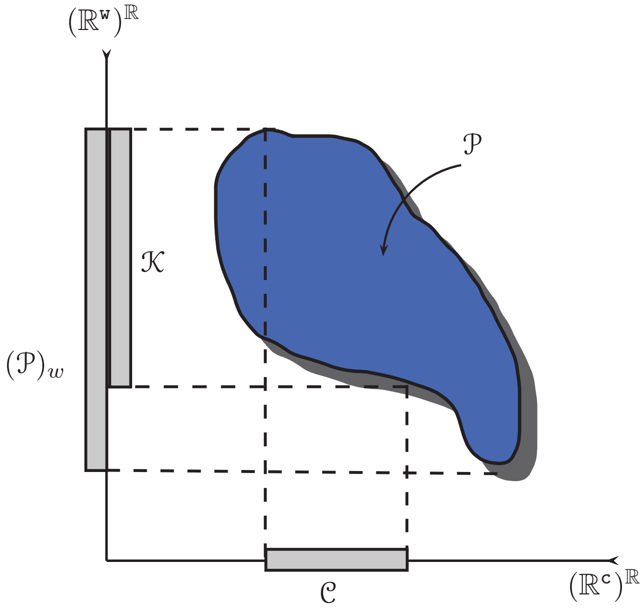

$(\mathbb{R}^{\mathtt{w}})^{\mathbb{R}}$
$\mathcal{P}$
$\mathcal{K}$
$(\mathcal{P})_w$
$\mathcal{C}$
$(\mathbb{R}^{\mathtt{c}})^{\mathbb{R}}$

Figure 3.6: The relation between $\mathcal{P}$, $(\mathcal{P})_w$, $\mathcal{C}$, and $\mathcal{K}$.

Definition 3.6. (Implementability: partial interconnection) *Let $\mathcal{P} \in \mathfrak{L}^{\mathtt{w}+\mathtt{c}}$, $\mathcal{C} \in \mathfrak{L}^{\mathtt{c}}$, and $\mathcal{K} \in \mathfrak{L}^{\mathtt{w}}$. Whenever $\mathcal{K}$ is obtained by interconnecting $\mathcal{P}$ and $\mathcal{C}$ through c, then we say that $\mathcal{C}$ implements $\mathcal{K}$. In addition, for a given $\mathcal{K} \in \mathfrak{L}^{\mathtt{w}}$ whenever there exists $\mathcal{C} \in \mathfrak{L}^{\mathtt{c}}$ such that $\mathcal{C}$ implements $\mathcal{K}$, then $\mathcal{K}$ is said to be implementable through c (with respect to $\mathcal{P}$).* □

The problem of implementability by partial interconnection is to characterize, for given $\mathcal{P} \in \mathfrak{L}^{\mathtt{w}+\mathtt{c}}$, all $\mathcal{K} \in \mathfrak{L}^{\mathtt{w}}$ for which there exists a $\mathcal{C} \in \mathfrak{L}^{\mathtt{c}}$ that implements $\mathcal{K}$ through c. In [WT02], a very simple and elegant solution to this problem is presented. Accordingly, it depends only on the projected full plant behavior $(\mathcal{P})_w$ and on the hidden behavior $\mathcal{N}$. The hidden behavior is the set of trajectories that w can assume after nullifying the control variables.

Definition 3.7. (Hidden behavior) *Let $\mathcal{P} \in \mathfrak{L}^{\mathtt{w}+\mathtt{c}}$. The hidden behavior $\mathcal{N} \in \mathfrak{L}^{\mathtt{w}}$ is the behavior consisting of the to-be-controlled variable that can occur when the control variables are restricted to be equal to zero:*

$$\mathcal{N} = \{w \in (\mathbb{R}^{\mathtt{w}})^{\mathbb{R}} \mid (w, 0) \in \mathcal{P}\}. \quad \square \tag{3.9}$$

From equation 3.7, we see that the controller has access to only the control variables. When the control variables are nullified, the controller receives no information about what is happening in the plant. Hence, the underlying idea of definition 3.7 is that the variables in $\mathcal{N}$ are hidden from the control variables

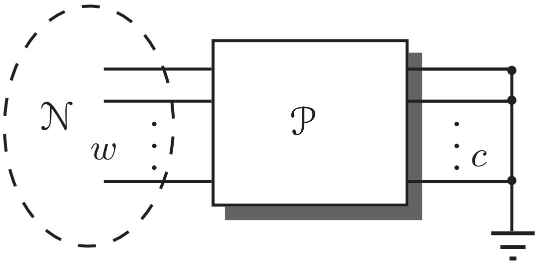


Figure 3.7: The hidden behavior

(see figure 3.7). Based on the hidden behavior and the manifest behavior, the implementability of the controlled behavior is formalized below.

Theorem 3.3. *Let $\mathcal{P} \in \mathfrak{L}^{\mathtt{w}+\mathtt{c}}$ be the full plant behavior. Then $\mathcal{K} \in \mathfrak{L}^{\mathtt{w}}$ is implementable by a controller $\mathcal{C} \in \mathfrak{L}^{\mathtt{c}}$ acting on the interconnection variable c if and only if*

$$\mathcal{N} \subseteq \mathcal{K} \subseteq (\mathcal{P})_w. \quad \square$$

Theorem 3.3 shows that $\mathcal{K}$ can be any behavior that is wedged in between the given behaviors $\mathcal{N}$ and $(\mathcal{P})_w$. The implementability condition played a central role to study the control problems in [TW02], [vdS03], [JWBT05]. Setting $c = 0$ can be considered as the maximum amount of control that a controller can impose. Due to this reason, $\mathcal{N}$ is termed as the maximally controlled behavior in [Pol00]. Moreover, the manifest behavior $(\mathcal{P})_w$ has been called the uncontrolled behavior because no controller can be attached to this manifest behavior. Thus, theorem 3.3 is described as a starting point for assessing the 'limits of performance' of the plant $\mathcal{P}$ when controlled by any controller $\mathcal{C}$ [vdS03].

In addition to implementability issues, the hidden behavior $\mathcal{N}$ also plays a role in describing observability and detectability within the behavior $\mathcal{P}$. The following theorem from [BT02] formalizes the last statement.

Theorem 3.4. *Let $\mathcal{P} \in \mathfrak{L}^{\mathtt{w}+\mathtt{c}}$ and let $\mathcal{N}$ be the hidden behavior as defined in equation 3.9. Then we have*

1. *in $\mathcal{P}$, w is observable from c if and only if $\mathcal{N} = 0$, and*
2. *in $\mathcal{P}$, w is detectable from c if and only if $\mathcal{N}$ is autonomous and stable.*

$\square$

As we have mentioned earlier that from the practical implementation point of view, we always consider the regular interconnection between $\mathcal{P}$ and $\mathcal{C}$. Now, we shall give basic results from [BT02] in the context of equations 3.4, 3.5, and 3.7.

Indeed, the output cardinality of a behavior is equal to the rank of the polynomial matrix in any of its kernel representations. Considering the above fact, the full interconnection of $\mathcal{P}$ and $\mathcal{C}$ is regular if and only if (following the theorem 3.1)

$$\mathtt{rank}(R) + \mathtt{rank}(C) = \mathtt{rank}\left(\begin{bmatrix} R \\ C \end{bmatrix}\right).$$

In the case of partial interconnection, we can write the behavior of the plant $\mathcal{P} \in \mathfrak{L}^{\mathtt{w}+\mathtt{c}}$ by

$$\begin{bmatrix} R\left(\frac{d}{dt}\right) & M\left(\frac{d}{dt}\right) \end{bmatrix} \begin{bmatrix} w \\ c \end{bmatrix} = 0.$$

For the above representation to be minimal, we have $\mathtt{p}(\mathcal{P}) = \mathtt{rowdim}\left(\begin{bmatrix} R & M \end{bmatrix}\right)$. Similarly, for $\mathcal{C} \in \mathfrak{L}^{\mathtt{c}}$, it is given by $C\left(\frac{d}{dt}\right) c = 0$ and $\mathtt{p}(\mathcal{C}) = \mathtt{rowdim}(C)$. Consequently, $\mathcal{K}_{\mathtt{full}}$ is represented by

$$\begin{bmatrix} R\left(\frac{d}{dt}\right) & M\left(\frac{d}{dt}\right) \\ 0 & C\left(\frac{d}{dt}\right) \end{bmatrix} \begin{bmatrix} w \\ c \end{bmatrix} = 0.$$

It is now clear if this representation is minimal then $\begin{bmatrix} R & M \\ 0 & C \end{bmatrix}$ must be of full row rank. This is equivalent to $\mathtt{p}(\mathcal{K}_{\text{full}}) = \mathtt{p}(\mathcal{P}_{\text{full}}) + \mathtt{p}(\mathcal{C})$ and hence, to regularity of the interconnection.

Chapter 4

Projection based approach to FTC in behavioral context

Contents

4.1 Introduction 74

4.2 Fault-Tolerant Control in the behavioral framework 75

4.2.1 Feedback Interconnection 75

4.2.2 Problem Formulation 77

4.3 Design and Implementation of Real-time FTC 80

4.3.1 The Behaviors 80

4.3.2 Set-up of the FTC Architecture 81

In this chapter, we shall present the solution to the problem of Fault-tolerant Control (FTC) by taking the behavioral system theoretic viewpoint. This chapter is composed of the works published in [JYS13e, JYS13a, JYS12c, JYS12f, JYS11b, JYS11c, JYS10b]. As mentioned before, in the mathematical framework of behavioral theory, the concept of interconnection among the system variables is the key point. The problem is that the kernel representation of the behavior we intend to control is not known in real-time. Therefore, we are interested in designing a fault accommodation scheme for an unknown plants' behavior through an appropriate behavioral interconnection. Here we deal simply with the trajectories that are generated by the system in real-time. These trajectories determine the behavior of a system in various (faulty/healthy) modes. Based on the desired interconnected behavior, only those trajectories are selected that obeys certain laws. Thus, whenever the trajectories do not belong to a certain desired behavior it is considered as due to the occurrence of fault in the system. The vantage point is that the fault tolerant control problem now becomes completely a real-time model free scheme. Moreover, no explicit fault diagnosis module is required in the demonstrated approach.

4.1 Introduction

A fault, in general, is defined as un-permitted dynamics that changes the dynamics of a closed-loop system in such a way it no longer satisfies the desired specifications [BKSL03]. Thus the aim of fault tolerant control (FTC) is to counteract those altered dynamics by applying a suitable control law such that the system *encore* achieves the desired specifications. Predominately, the process to re-establish the desired specifications undergoes the following two cascade stages: Fault Detection and Diagnoses (FDD), and Controller Reconfiguration (CR). The purpose of FDD is to use available signals to detect, identify, and isolate possibly the sensor faults, actuator faults, and any other system faults. Conversely, the CR module reckons the to-be-required actions so the system can still continue to operate safely even under the faulty conditions. In terms of condition monitoring or FDD, the existing methods are grouped into the following two categories:

1. Model based FDD [CP99];

2. Data driven FDD including knowledge based FDD [HTYJLM09].

In the early days (1980's onwards), a model-based FDD constituted the mainstream of research, and a number of techniques were developed. Depending on whether the system model can be represented as either a state-space model or an input-output model, FDD can roughly be classified into the following two groups: observer based FDD [BKSL03] and system identification based FDD [Ise84]. On comprising these two respective modules individually with the CR unit, it results in the following FTC strategies, namely *model-based FTC* and *data-driven FTC*.

Model-based FTC approaches have their own limitations to deal with model uncertainties in real-time as shown in Chapter 1. On the other hand, data-driven approaches that comprise the estimation of a plant model involve individual timing issues in fault diagnosis and fault accommodation. See [YK04b], [Sta04] for more details on these issues. It has been shown that the prime cause of these limitations is the use of the FDD unit for *reconfigurable* FTC systems. Therefore, our notion of data-driven approach to FTC does not even involve any use of an explicit FDD module.

The FTC problem is concerned with the control of the faulty system [BKSL03, Definition 7.1], and our main central point takes into account the controller reconfiguration mechanism. We will show that in active FTC systems, the use of the online FDD module can be avoided providing the system can achieve the desired specifications by just changing the control law. Nevertheless, for other types of faults that require “reconfiguring the plant”, i.e.

the "replacement" of actuators or sensors while keeping the same (or even changing the) controller, one need an explicit FDI mechanism to identify the size and the location of a fault.

An FTC approach without utilizing an FDD module is also studied in [YY06]. Unlike the [YY06], first we do not have the online estimates of an occurring fault. Secondly, we do not assume the availability of the system states (in a state-space representation) at anytime. Here, the main objective is to re-configure the controller directly based on the trajectories generated by the system in *real-time*. This renders a fast and a reliable data-driven fault tolerant system. The presented FTC strategy lies under a broad category of projection-based active FTC mechanism. In the demonstrated control architecture for the FTC, the key role will be played by the "control performance evaluator". We directly evaluate the control performance of the closed-loop system unlike evaluating the estimator performance which is mostly seen within the existing literature on projection-based FTC.

4.2 Fault-Tolerant Control in the behavioral framework

The projection-based approach relies on constructing a bank of pre-designed controllers as illustrated in Chapter 1. Within this set of controllers, it is assumed that either only one controller or a subset of controllers has the ability to achieve the performance specifications. This requires switching of the controller corresponding to the operating mode of the plant. In the former case, it is called one-shot switching while in the latter, it is termed as continuous switching or periodic switching. Our interest lies in the former case. Generally, the use of FDD module is seen in projection-based approaches that utilizes one-shot switching to identify the exact operating mode of the plant and to extract the complete information of the working plant. This demand comes from the fact that which controller has to be switched in the closed-loop, which brings a big challenge to deal with this approach from real-time point of view. In the following, we shall illustrate the real-time implementation of this approach without using an explicit FDD module.

4.2.1 Feedback Interconnection

We consider the unity feedback control configuration to deal with an FTC problem. In the classical sense, the architecture of the configuration is illustrated in Fig. 4.1. In the behavioral sense, we can represent this configuration as illustrated in Fig. 4.2. In the latter figure, notice that the directions on

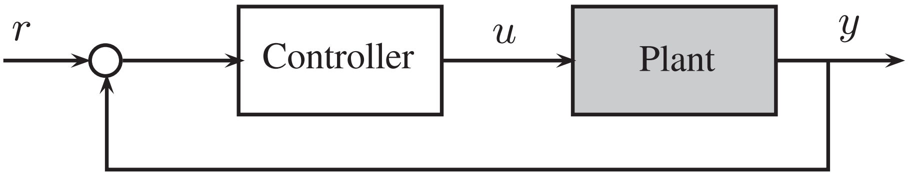


Figure 4.1: Feedback Configuration in the classical sense

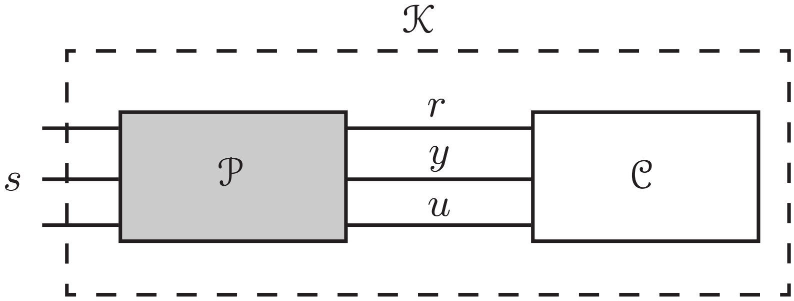


Figure 4.2: Feedback Configuration in the behavioral sense

the variables are not shown as it is mentioned before that the inputs and outputs of a system are, generally, not decided *a priori*. However, they can be naturally distinguished within this framework.

A set of time dependent variables $s = \mathtt{col}(r, y, u)$ is provided whose values lies in the signal space $\mathbb{S}$ having the dimension $\mathtt{s} = \mathtt{r} + \mathtt{y} + \mathtt{u}$. Taking the behavioral point of view, we can now define the trajectory-based dynamical system for the plant and the controller by $\Sigma_{\mathcal{P}} = (\mathbb{T}, \mathbb{S}, \mathcal{P})$, and $\Sigma_{\mathcal{C}} = (\mathbb{T}, \mathbb{S}, \mathcal{C})$ respectively, where $\mathbb{T} \subseteq \mathbb{R}$, $\mathbb{S} \subseteq \mathbb{R}^{\mathtt{r}+\mathtt{y}+\mathtt{u}}$, $\mathcal{P} \subseteq \mathbb{S}^{\mathbb{T}}$, and their behaviors in the following way.

$$\mathcal{P} = \left\{ s = \mathtt{col}(r, y, u) \in \mathbb{S}^{\mathbb{T}} \middle| R\left(\frac{d}{dt}\right) s = 0 \right\}, \tag{4.1}$$

$$\text{where} \quad R(\xi) = \begin{bmatrix} 0_{\mathtt{r}} & D_p(\xi) & -N_p(\xi) \end{bmatrix}. \tag{4.2}$$

with $D_p(\xi) \in \mathbb{R}^{\bullet \times \mathtt{y}}[\xi], N_p(\xi) \in \mathbb{R}^{\bullet \times \mathtt{u}}[\xi]$ being co-prime polynomials, and $0_{\mathtt{r}}$ representing the zero matrix of $\mathtt{r}$ dimension. From the input/output point of view, y is considered as the output of the plant and u as the input. With this partition of inputs and outputs, together with definition 2.17, evidently $D_p(\xi)^{-1}N_p(\xi) = G(\xi)$ defines a proper rational matrix with $D_p(\xi) \neq 0$. In a similar way, the behavior of the controller $\Sigma_{\mathcal{C}}$ is given by

$$\mathcal{C} = \left\{ s = \mathtt{col}(r, y, u) \in \mathbb{S}^{\mathbb{T}} \middle| C\left(\frac{d}{dt}\right) s = 0 \right\}, \tag{4.3}$$

$$\text{where} \quad C(\xi) = \begin{bmatrix} N_c(\xi) & -N_c(\xi) & -D_c(\xi) \end{bmatrix}. \tag{4.4}$$

with $D_c(\xi) \in \mathbb{R}^{\bullet\times\mathtt{u}}[\xi]$, $N_c(\xi) \in \mathbb{R}^{\bullet\times\mathtt{y}}[\xi]$ being co-prime polynomials, and $D_c(\xi)^{-1}N_c(\xi) = H(\xi)$ representing a proper rational matrix with $D_c(\xi) \neq 0$. In this controller configuration, u is the output of the controller, and (r, y) are the inputs. Whenever the above two systems interconnects, the controller imposes some restrictions on the behavior of the plant. These imposed restrictions by $\mathcal{C}$ on $\mathcal{P}$ are termed as the controlled behavior or the *implemented behavior* where the variables satisfy the dynamics of both systems, i.e.

$$\mathcal{K} = \{s = \mathtt{col}(r, y, u)\,|s \in \mathcal{P} \text{ and } s \in \mathcal{C}\} \tag{4.5}$$

The kernel representation of the interconnected system $\mathcal{K}$ is then given by

$$\mathcal{K} \equiv \begin{bmatrix} R\left(\frac{d}{dt}\right) \\ C\left(\frac{d}{dt}\right) \end{bmatrix} s = 0. \tag{4.6}$$

in which (y, u) are the outputs and r is the input. Clearly,

$$\mathtt{p}(\mathcal{K}) = \mathtt{p}(\mathcal{P}) + \mathtt{p}(\mathcal{C}).$$

The last equality demonstrates that the interconnection between $\mathcal{P}$ and $\mathcal{C}$ in the feedback configuration is always a regular interconnection.

4.2.2 Problem Formulation

The real-time problem of controlling a faulty system, as defined in Chapter 1 is posed that the operating plant should achieve the control objectives at anytime, i.e. regardless of any occurrence of a fault. In this respect, we can single out a subset of plants' behavior as desirable. We call it the desired behavior, denoted by $\mathcal{D}$, and it can be considered as equivalent to the control objective $\mathfrak{O}$. The desired behavior is, indeed, defined in terms of the available system variables, and it is given by

$$\mathcal{D} = \left\{s = \mathtt{col}(r, y, u) \in \mathbb{S}^{\mathbb{T}}\,|J(s) \leq \lambda\right\}, \tag{4.7}$$

where $J : (\mathbb{R}^{\mathtt{s}})^{\mathbb{R}} \to \mathbb{R}, s \mapsto J(s)$ defines the control performance functional with $\lambda \in \mathbb{R}$ denoting the threshold limit below which the performance is considered satisfactory. The above performance functional is a function of all the signals obtained from the closed-loop system, which gives the real-time performance measure of the system. If any of the signals is unbounded then it implies that the functional is also unbounded.

Faults affect the dynamics of the system in a way that the control specifications are not satisfied. However, in some cases, the operating controller in the feedback control loop is extremely robust making a fault tolerable within a FTC system. Hence, no change in the control law would be required. Whenever there is an actuator blockage, sensor blockage, or some internal components of the plant change to a large extent, then it is called an occurring fault. We do not have any *a priori* knowledge about the model of the plant in real-time, so, for an instance, the percentage loss of power efficiency in actuators or sensors is not known. In some cases, the operating controller is robust to a certain extent that the loss of power efficiency is easily correctable by the operating controllers in the closed-loop, i.e. the controller reconfiguration mechanism does not need to be initiated. With the above considerations, we define two classes of faults, namely *minor faults*, and *major faults*.

Definition 4.1 (Minor Faults). *A fault is said to be a minor fault whenever there is no need of reconfiguring the controller in the closed-loop.*

Definition 4.2 (Major Faults). *A fault is said to be a major fault whenever* $\mathcal{K} \nsubseteq \mathcal{D}$.

Let us analyze the occurrence of faults in an unknown system where the control objective is that the system should be stable, i.e. the output trajectory for zero reference is zero as time tends to infinity. The plant is a multi-variable system with two actuators and four sensors [Sta04] whose state-space matrices are given as

$$A = \begin{bmatrix} -0.0226 & -36.6 & -18.9 & -32.1 \\ 0 & -1.9 & 0.983 & 0 \\ 0.0123 & -11.7 & -2.63 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ -0.141 & 0 \\ -77.8 & 22.4 \\ 0 & 0 \end{bmatrix}. \tag{4.8}$$

We consider a fault as the loss in power efficiency of actuator-1 appearing on time $t = 7sec$.

Case-1 : An occurring fault causes 50% loss in power efficiency of actuator-1. The sensor's trajectory as illustrated in Fig. 4.3 shows that after the occurrence of fault the stability of the closed-loop is retained. This implies that some type of faults can easily be tolerated without any controller reconfiguration. Such type of faults falls under the category of minor faults.

Case-2 : An occurring fault causes 90% loss in power efficiency of actuator-1. The sensor's trajectory is illustrated in Fig. 4.4. In this case, the controlled system becomes unstable after an occurrence of a fault. Thus, such class of fault requires reconfiguring the controller in real-time to achieve the control objective.

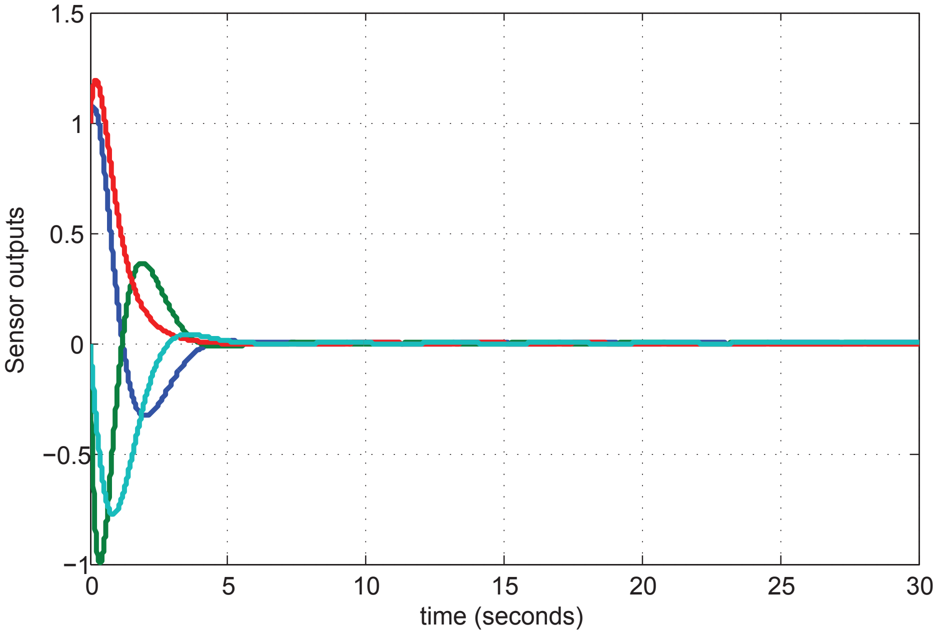


Figure 4.3: Output trajectories in Case-1

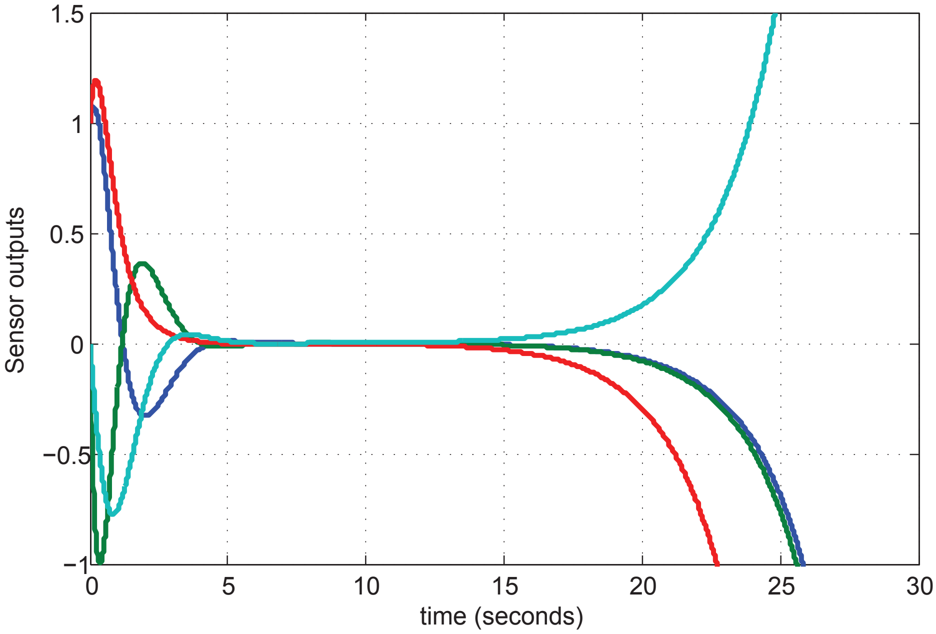


Figure 4.4: Output trajectories in Case-2

Considering the above definitions of occurring faults, the real-time FTC problem we are dealing with is posed in the following way. Given a vector space of time signals $\mathbb{S}^{\mathbb{T}}$, and the desired behavior $\mathcal{D}$, the problem is to find an appropriate controller $\mathcal{C}$, without using any *a priori* knowledge of the model of the plant in real-time, which have the suitable control actions such that the controlled behavior $\mathcal{K}$ satisfy the desired behavior $\mathcal{D}$ at anytime.

4.3 Design and Implementation of Real-time FTC

In chapter 1, various active FTC schemes are discussed which require a precise knowledge of the plants' model during the FDD operation. On the other side, the novelty of the proposed behavioral approach lies in its time-trajectories outlook of approaching an FTC problem, where no such knowledge is required. Nevertheless, the first stage of the development of a fault-tolerant system requires Failure Mode and Effective Analysis (FMEA) [BKSL03]. FMEA's objective is to forecast systematically how fault effects in elements relate to faults at inputs, or outputs within the elements, and what reactions should be imposed on the system when a certain faults appears. Therefore, a mandatory prerequisite for achieving fault-tolerance is to have an effective FMEA of the system. We termed this phase as the Analysis & Development (AD) phase, which aims to provide a complete coverage of possible occurring faults in the closed-loop as well as the corresponding remedial measures. From the AD phase, it is assumed that a finite set of controllers

$$\mathbf{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_N\} \tag{4.9}$$

is constructed, which makes the desired behavior $\mathcal{D}$ implementable. An approach to perform this analysis procedure is discussed in [Wu04b] and the references therein.

4.3.1 The Behaviors

All modeling assumptions about the operating plant are embedded within the $R(\xi)$ matrix given in (5.1). At a run time, this matrix is not determined in the proposed structure. We form a measurement set $\mathcal{M}$, which is non-empty subset of $\mathbb{S}^{\mathbb{T}}$. From the real-time point of view, this is formalized in the following definition.

Definition 4.3 (Experimental plant's behavior)**.** *Given a vector space of time-dependent signals* $\mathbb{S}^{\mathbb{T}}$*, a dynamical system* $\Sigma_{\mathcal{P}} = (\mathbb{T}, \mathbb{S}, \mathcal{P})$*, and a mea-*

surement set $\mathcal{M} \subseteq \mathbb{S}^{\mathbb{T}}$, the behavior of the plant $\mathcal{P}$ is a superset of the (experimental) measurement on time intervals, i.e.

$$\mathcal{M}_\tau \subseteq \mathsf{O}_\tau(\mathcal{P}) \tag{4.10}$$

where O_τ is the time truncation operator given as

$$[\mathsf{O}_\tau(x)](t) = \begin{cases} x(t), & t_n - \tau \leq t < t_n \ ; \\ 0, & \textit{otherwise.} \end{cases}$$

where $t_n = n\tau, \forall n = 1, 2, \ldots$. □

The role of introducing the time-truncation operator is to produce time-dependent subsets $\mathcal{M}_\tau$ of $\mathcal{M}$ for the interval of length $\tau \in \mathbb{R}$. From the above definition, for any controller $\mathcal{C}$ together with the behavior of the plant, we have

$$\mathsf{O}_\tau(\mathsf{O}_\tau^{-1}(\mathcal{M}_\tau) \cap \mathcal{C}) \subseteq \mathsf{O}_\tau(\mathcal{K}), \tag{4.11}$$

where $\mathsf{O}_\tau^{-1}(\mathcal{M}_\tau)$ denotes the pre-image of $\mathcal{M}_\tau$. It is interesting to note from (4.11) that the controlled behavior, by construction, is formulated independent of $\mathcal{P}$, directly. We obtain the following immediate proposition.

Proposition 4.1. *Given a vector space of time-dependent signals $\mathbb{S}^{\mathbb{T}}$, a dynamical system $\Sigma_{\mathcal{C}} = (\mathbb{T}, \mathbb{S}, \mathcal{C})$, the desired behavior $\mathcal{D}$, and a measurement set $\mathcal{M}_\tau \subseteq \mathsf{O}_\tau(\mathcal{P})$, the controller $\mathcal{C}$ achieves the control objective on the interval of length τ if*

$$\mathsf{O}_\tau(\mathsf{O}_\tau^{-1}(\mathcal{M}_\tau) \cap \mathcal{C}) \subseteq \mathsf{O}_\tau(\mathcal{D}). \tag{4.12}$$

□

Stating otherwise, whenever the above inclusion is satisfied, then $\mathcal{C}$ implements $\mathcal{D}$. On the other side, if the above condition is not satisfied for any controller from the controller's set, then the desired behavior "is not implementable" or is not achievable. Proposition 4.1, in point of fact, provides the condition for *invalidation of the controller* which is solely based upon the measurements observed from the closed-loop system during the interval of length τ. However, "the anytime" property to achieve the real-time tolerance against occurring faults is still needed to be demonstrated.

4.3.2 Set-up of the FTC Architecture

Since, we already have a set of control laws that makes the desired behavior implementable, the structure of the real-time fault-tolerant control in the behavioral context is provided in Fig. 4.5. The plant in the figure is shaded as we do not have any *a priori* knowledge of it in real-time. Here in this archi-

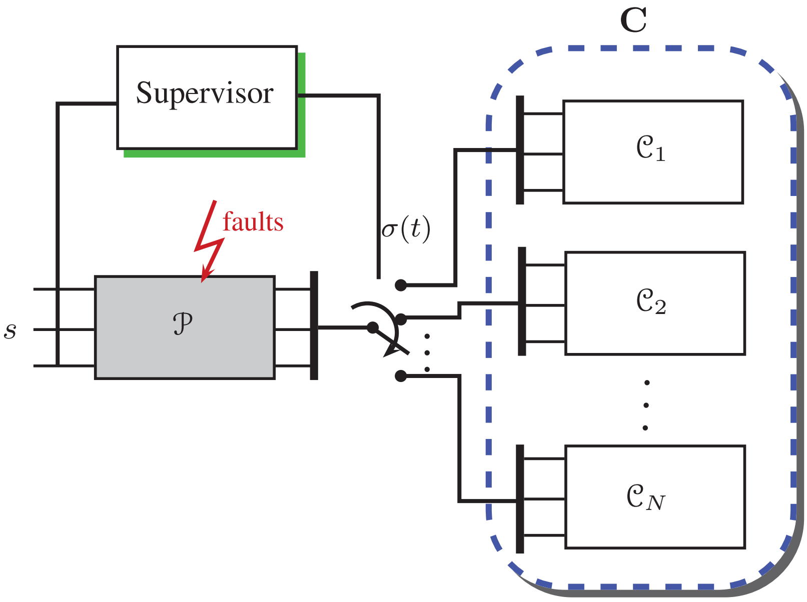


Figure 4.5: Projection-based Active FTC in behavioral context

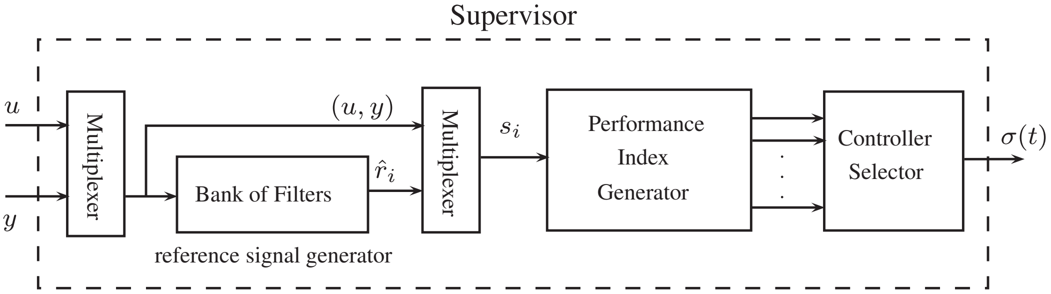


Figure 4.6: Internal structure of the supervisor

tecture, a significant role is played by the supervisor or the Reconfiguration Mechanism (RM). It is the job of the RM block that manages the switching of controllers from the set given in (4.9). Precisely, the RM performs the "when-which" task that implies *when* to change the control law, and *which* controller should be place into the closed-loop. Assuming the existence of at least one corrective controller in the pre-designed set (4.9) for the faults occurring in the plant and taking definition 4.3 into account, a simple conceptual solution to the controller selection would be to evaluate experimentally each candidate controller's performance by applying it to the plant. Unfortunately, not all the potential controllers can be tested simultaneously in the feedback loop.

It is a fact that without further modeling assumptions on the model of the plant in the real-time, it is logically impossible to verify that a controller from the set of controllers will implement the desired behavior. To perform this task without aforesaid assumptions, we construct our supervisor as illustrated in Fig. 4.6. The main job of this supervisor is to switch the controller in the closed-loop having the corrective actions in one-shot, i.e. in a single switch. The explicit structure of the reconfiguration mechanism consists of a bank of filters, a Performance Index Generator (PIG) block and a controller selector block.

Bank of Filters : Since no knowledge of plant's model is available, we only obtain a measurement set $\mathcal{M}_\tau$ during the interval of length τ, which is composed of the trajectories $u(t)$ and $y(t)$ produced by the plant. If a controller $\mathcal{C}$ were in the loop when the plant produced the trajectories $\mathtt{col}(y, u)$, then the restrictions imposed by the controller behavior (4.3) would be

$$D_c(\xi)u(t) = N_c(\xi)r(t) - N_c(\xi)y(t), t_n - \tau \leq t < t_n$$

for some $r \in (\mathbb{R}^{\mathtt{r}})^{\mathbb{R}}$ or equivalently,

$$N_c(\xi)r(t) = D_c(\xi)u(t) + N_c(\xi)y(t), t_n - \tau \leq t < t_n. \tag{4.13}$$

Assume that all controllers in the controller's bank are stable causally left invertible, then based on the observed set $\mathcal{M}_\tau$ the trajectory $r(t)$ can be evaluated as

$$\hat{r}(t) = (N_c(\xi))^{-1}(D_c(\xi)u(t) + N_c(\xi)y(t)), t_n - \tau \leq t < t_n. \tag{4.14}$$

Equation (4.13), in fact, yields the controlled behavior $\mathcal{K}$ as defined in (4.5), since here $(y, u) \in \mathcal{P}$ and $(\hat{r}, y, u) \in \mathcal{C}$. Consequently, for a measurement set $\mathcal{M}_\tau \subseteq \mathcal{P}$, if there exists a trajectory $\hat{r}_i(t)$ corresponding to the i^{th} controller $\mathcal{C}_i, i = 1, 2, \ldots, N$, then it would yield the corresponding controlled behavior

$\mathsf{O}_\tau(\mathcal{K}_i) \supseteq \mathsf{O}_\tau(\mathsf{O}_\tau^{-1}(\mathcal{M}_\tau) \cap \mathcal{C}_i), \forall i \in \{1, 2, \ldots, N\}$. Equation (4.14) defines a filter which reconstructs the virtual reference signal $\hat{r}(t)$ from the measurement set $\mathcal{M}_\tau$ [ST97]. From this, we have now determined the controlled behavior of all controllers with respect to $\mathcal{P}$ at run time, however, no knowledge of the plant's model is used here. Now, we can proceed towards evaluating the performance of these controlled behaviors.

PIG block : The measurements generated by the plant together with the virtual reference, i.e. $\hat{s} = (\hat{r}, y, u) \in \mathbb{S}^{\mathbb{T}}$, in the interval of length τ are then fed to the PIG block. This block yields N performance indices

$$\{J(\hat{s}_i), i = 1, 2, \ldots, N\}, \tag{4.15}$$

for the corresponding N controllers, which are evaluated by considering the signal $\hat{s}$ during the interval of length τ, i.e. $\hat{s}_i \in \mathsf{O}_\tau(\mathcal{P} \cap \mathcal{C}_i)$.

Remark 4.1. *The observed measurement is not related to any particular experimental setting. Hence a deep consequence of proposition 4.1 is that any controller from the bank can be tested, even if it is not actually interconnected to the plant.*

Controller Selection : The controller selector block is the next sub-system that produces a piecewise constant signal (the switching signal) $\sigma(t)$ based on $\{J(\hat{s}_i)\}_{i=1}^N$ whose job is to select the controller having corrective actions from the bank of controllers. The switching signal is a map from the time axis $\mathbb{T}$ to the controllers index set $\{1, 2, ..., N\}$, i.e. $\sigma : \mathbb{T} \to \{1, 2, ..., N\}$. The control performance is evaluated during the interval of length τ, and if it requires switching of the controller, the switch will occur after time τ exclusively. Therefore, it imposes a lower bound on the length of intervals between successive switches. This minimum length of time in which a controller is active in the loop is known as the *dwell time* [see Appendix A]. The control selection logic is then realized through

$$\sigma(t) = \sigma(t_n) \text{ for } t_n \leq t < t_{n+1} \tag{4.16}$$

with the updating rule

$$\sigma(t_{n+1}) = \begin{cases} \sigma(t_n), & \text{if } \mathsf{O}_\tau(\mathcal{K}) \subseteq \mathsf{O}_\tau(\mathcal{D}); \\ \texttt{argmin}\{J(\hat{s}_i)\}_{i \neq \sigma(t_n)}, & \text{if } \mathsf{O}_\tau(\mathcal{K}) \nsubseteq \mathsf{O}_\tau(\mathcal{D}). \end{cases} \tag{4.17}$$

The controller selector block contains the control selection algorithm given in (4.16)-(4.17). The switching logic implements the following: it lets the stable dynamics of the closed-loop switched system have enough time to decay before a next possible switching occurs, and it bounds the detection delay, i.e. the time elapsed from the occurrence of a fault to the invalidation of the active controller. Now we provide the main result of this chapter.

Proposition 4.2. *Given the implementable desired behavior $\mathcal{D}$, and a measurement set $\mathcal{M}_\tau \subseteq \mathsf{O}_\tau(\mathcal{P})$. For any occurrence of a fault, if the switching signal $\sigma(t)$ is selected according to (4.16) together with (4.17) then the system is a real-time fault tolerant control system.* □

Before proving the above proposition, we give the following lemma.

Lemma 4.1. *For any controller $\mathcal{C}_i, \forall i \in \{1, 2, \dots, N\}$, the corresponding virtual reference signal $\hat{r}_i$ converges exponentially to the true reference signal r.*

Proof. For any controller $\mathcal{C}_i$, ((4.13) yields

$$N_{c_i}(\xi)\hat{r}_i(t) = D_{c_i}(\xi)u(t) + N_{c_i}(\xi)y(t). \tag{4.18}$$

Then, the corresponding controller connected in the closed-loop gives the following control signal

$$D_{c_i}(\xi)u(t) = N_{c_i}(\xi)r(t) - N_{c_i}(\xi)y(t). \tag{4.19}$$

Subtracting (4.19) from (4.18), we get

$$N_{c_i}(\xi)(\hat{r}_i(t) - r(t)) = 0. \tag{4.20}$$

Hence, being $N_{c_i}(\xi)$ a stable differential operator, $\hat{r}_i(t) - r(t)$ converges exponentially to zero. □

The sole purpose of above lemma is to show that the trajectory $\hat{s}$ is no different from the trajectory s.

Proof of Proposition 4.2. We give the proof by induction. Without any loss of generality, consider a bank of three controllers $\mathbf{C} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\}$ is constructed in the AD phase. The real-time operation is initiated with an unknown $\mathcal{P}$ interconnected with the valid $\mathcal{C}_1$. Thus, $\mathsf{O}_\tau(\mathsf{O}_\tau^{-1}(\mathcal{M}_\tau) \cap \mathcal{C}_1) = \mathsf{O}_\tau(\mathcal{K}_1) \subseteq \mathsf{O}_\tau(\mathcal{D})$. Suppose, a minor fault occurs into the system. Since the occurring minor fault does not change the behavior of the system, from the evaluation of the control performance after an interval of length τ together with definition 4.1, we would have $\mathsf{O}_\tau(\mathcal{K}_1) \subseteq \mathsf{O}_\tau(\mathcal{D})$. This implies that there is no need to change the controller $\mathcal{C}_1$, and it will remain connected in the closed-loop. Consider now that a major fault occurs into the system. Indeed, this will change the behavior of the plant, and suppose, this new behavior is then given by $\mathcal{P}^f$. From the definition 4.2, it implies that an occurrence of major fault causes $\mathcal{K}_1 \nsubseteq \mathcal{D}$. Therefore, whenever the last inclusion satisfies, the operating controller in the loop is invalidated implying that a fault has occurred. Since there exists a controller in bank that implements $\mathcal{D}$, therefore, the only task remains is to switch that controller in the closed-loop in one-shot.

Again, without any loss of generality, suppose, that controller is $\mathcal{C}_2$ and not $\mathcal{C}_3$. Using the measurement set $\mathcal{M}_\tau$ generated by the plant, two virtual reference signals are evaluated by (4.13), which gives two sets of trajectories: $\hat{s}_2 = \texttt{col}(\hat{r}_2, y, u)$, and $\hat{s}_3 = \texttt{col}(\hat{r}_3, y, u)$. Clearly, for these set of trajectories, we have two corresponding virtual interconnected system, namely $\hat{\mathcal{K}}_2$ and $\hat{\mathcal{K}}_3$, defined as

$$\hat{\mathcal{K}}_2 = \{(\hat{s}_2 = \texttt{col}(\hat{r}_2, y, u)|(y, u) \in \mathcal{P}^f \text{ and } (\hat{r}_2, y, u) \in \mathcal{C}_2\}, \tag{4.21}$$

$$\hat{\mathcal{K}}_3 = \{(\hat{s}_3 = \texttt{col}(\hat{r}_3, y, u)|(y, u) \in \mathcal{P}^f \text{ and } (\hat{r}_3, y, u) \in \mathcal{C}_3\}. \tag{4.22}$$

From the above, $\mathcal{C}_2$ is supposed to be that right controller, not $\mathcal{C}_3$. This will, indeed, satisfies $\mathsf{O}_\tau(\hat{\mathcal{K}}_2) \subseteq \mathsf{O}_\tau(\mathcal{D})$. Consequently, the controller $\mathcal{C}_2$ will be switched in to the closed-loop by the switching logic (4.16-4.17), instead of the controller $\mathcal{C}_3$. The lemma 4.1 gives $\hat{r} = r$, which implies that $\hat{\mathcal{K}}_2 = \mathcal{K}_2$ and hence $\mathsf{O}_\tau(\mathcal{K}_2) \subseteq \mathsf{O}_\tau(\mathcal{D})$. Further, if the controller $\mathcal{C}_2$ is not invalidated with the evolving time, i.e. for $n = 1, 2, \ldots$, then it would stay in the closed-loop, this concludes $\mathcal{K}_2 \subseteq \mathcal{D}$. This proves that the system is a real-time fault-tolerant control system.

Assuming that the above proof holds true for i number of controller in the bank, the inductive step is to show that it holds for $i+1$ number of controllers as well. From the FMEA analysis, there exists at least one controller in the bank that implements the desired behavior. Therefore, there could be more than one controller that implements $\mathcal{D}$ so the $(i+1)^{th}$ controller can be either a valid controller or an invalid controller. Suppose that the $(i+1)^{th}$ controller is an additional valid controller. This yields the controlled behaviors, given by

$$\hat{\mathcal{K}}_2 = \{(\hat{s}_2 = \texttt{col}(\hat{r}_2, y, u)|(y, u) \in \mathcal{P}^f \text{ and } (\hat{r}_2, y, u) \in \mathcal{C}_2\}, \tag{4.23}$$

$$\hat{\mathcal{K}}_{i+1} = \{(\hat{s}_{i+1} = \texttt{col}(\hat{r}_{i+1}, y, u)|(y, u) \in \mathcal{P}^f \text{ and } (\hat{r}_{i+1}, y, u) \in \mathcal{C}_{i+1}\}. \tag{4.24}$$

These behaviors, indeed, satisfy $\mathsf{O}_\tau(\hat{\mathcal{K}}_2) \subseteq \mathsf{O}_\tau(\mathcal{D})$, $\mathsf{O}_\tau(\hat{\mathcal{K}}_{i+1}) \subseteq \mathsf{O}_\tau(\mathcal{D})$. However, according to the switching logic (4.16-4.17), only that controller will be switched in the loop, which has the minimum value of the performance functional. Now, suppose that the $(i+1)^{th}$ controller is an invalid controller. Indeed, the controlled behaviors satisfy $\mathsf{O}_\tau(\hat{\mathcal{K}}_j) \nsubseteq \mathsf{O}_\tau(\mathcal{D}), \forall j \in \{1, 3, \ldots, i, i+1\}$, and $\mathsf{O}_\tau(\hat{\mathcal{K}}_2) \subseteq \mathsf{O}_\tau(\mathcal{D})$. This concludes that the closed-loop system with $(i+1)$ numbers of controllers in the bank is a real-time fault-tolerant control system. □

Chapter 5

Online redesign approach to FTC in behavioral context

Contents

5.1 Fault-Tolerant Control: partial interconnection case 88

5.1.1 Feedback Interconnection within the partial interconnection 88

5.1.2 Problem Formulation 90

5.1.3 Design and Implementation of Real-time FTC via partial interconnection 91

5.2 Fault-tolerant Control: full interconnection case . . . 95

5.2.1 Feedback Interconnection within the full interconnection 95

5.2.2 Design and Implementation of Real-time FTC via full interconnection 97

In the previous chapter, we have presented a novel projection-based approach to solve the active fault-tolerant control problem posed in the chapter 1. In this chapter, we shall present a novel second approach lying under the taxonomy of AFTC systems, namely the online redesign approach. This chapter is composed of the works published in [JYS13f, JYS13d, JYS13b, JYS12a, JYS12d, JYS12g, JYS12b, JYS11a, JYS11c, JYS10b, JYS10c]. Unlike the previous approach, here we do not assume that a set of controllers is provided. Instead we design an online controller by taking the real-time measurements generated by the plant. To illustrate this approach, we use the mathematical framework of behavioral theory. The prevailing issue is that the behavior we intend to provide tolerance against occurring fault is not known in real-time. The key advantage in this approach, as well, is that we do not use an explicit fault diagnosis module to achieve the above task, which make this approach quite attractive from real-time point of view. In the following sections, we shall directly present our online controller design approach for fault-tolerance without dwelling much into the introduction of fault-tolerant systems.

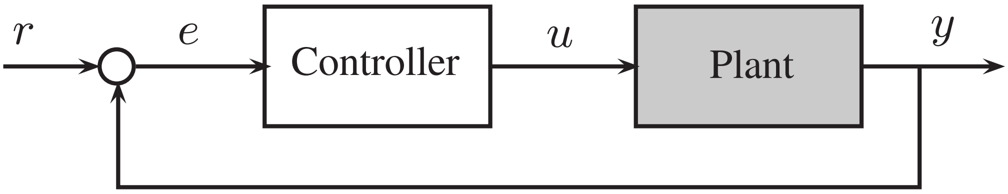


Figure 5.1: Feedback Configuration in the classical sense

5.1 Fault-Tolerant Control: partial interconnection case

The online redesign based approach relies on synthesizing a new control law for a plant subject to occurring fault. The classical way of achieving the fault-tolerance of the system, as mentioned before, involves two cascaded operating modules. In this section, we present an approach that deals with fault-tolerant systems by taking the benefits of the trajectory based viewpoint of behavioral theory. In the course of achieving the above tasks, we do not use any *a priori* information about the plant' model in real-time. Consequently, a new online controller that can satisfy the control objective is computed solely based on the real-time trajectories generated by the system.

5.1.1 Feedback Interconnection within the partial interconnection

We consider the unity feedback control configuration to deal with an active FTC problem. In Chapter 4, we have shown the feedback control configuration in the classical sense together with its interpretation in the behavioral sense. The interconnection is considered as the full interconnection between the plant and the controller. Here, we shall demonstrate the results for both types of interconnections, namely the full interconnection and the partial interconnection. In the classical sense, the architecture of the feedback configuration is illustrated in Fig. 5.1. Interpreting this configuration within the case of partial interconnection in a behavioral sense, it is shown in the Fig. 5.2.

A set of time dependent variables $s = \mathtt{col}(r, y, e, u)$ are provided whose values lies in the signal space $\mathbb{S}$ having the dimension $\mathtt{s} = \mathtt{r} + \mathtt{y} + \mathtt{e} + \mathtt{u}$. In the case of partial interconnection, all components of s do not take part while making an interconnection, but only few of them. The variables through which the plant and the controller interconnect are termed as the control variables, which are denoted by $c = \mathtt{col}(e, u)$. The rest of the variables are termed as the manifest variables, denoted by $w = \mathtt{col}(r, y)$. Clearly, $s = \mathtt{col}(w, c)$.

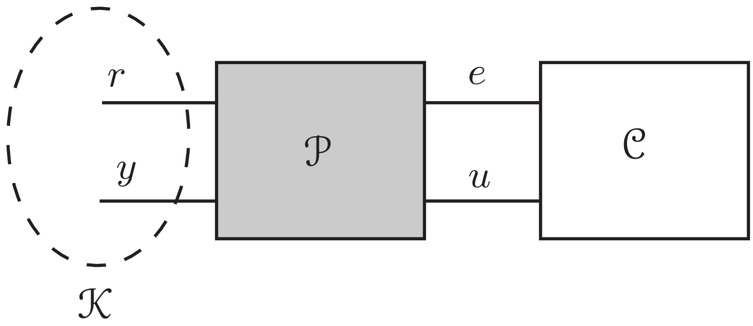


Figure 5.2: Feedback Configuration in the behavioral sense

Taking this point of view, we can now define the trajectory-based dynamical system for the plant and the controller by $\Sigma_{\mathcal{P}} = (\mathbb{T}, \mathbb{S}, \mathcal{P})$, and $\Sigma_{\mathcal{C}} = (\mathbb{T}, \mathbb{S}, \mathcal{C})$ respectively, where $\mathbb{T} \subseteq \mathbb{R}$, $\mathbb{S} \subseteq \mathbb{R}^{\mathtt{r}+\mathtt{y}+\mathtt{e}+\mathtt{u}}$, $\mathcal{P} \subseteq \mathbb{S}^{\mathbb{T}}$, and their behaviors in the following way.

$$\mathcal{P} = \left\{ s = \mathtt{col}(w, c) \in \mathbb{S}^{\mathbb{T}} \middle| \begin{bmatrix} R\left(\frac{d}{dt}\right) & -M\left(\frac{d}{dt}\right) \end{bmatrix} \begin{bmatrix} w \\ c \end{bmatrix} = 0 \right\}, \tag{5.1}$$

$$\text{where} \quad R(\xi) = \begin{bmatrix} I_{\mathtt{r}} & -I_{\mathtt{y}} \\ 0_{\mathtt{r}} & D_p(\xi) \end{bmatrix}, \quad M(\xi) = \begin{bmatrix} I_{\mathtt{e}} & 0_{\mathtt{u}} \\ 0_{\mathtt{e}} & N_p(\xi) \end{bmatrix} \tag{5.2}$$

with $D_p(\xi) \in \mathbb{R}^{\bullet \times \mathtt{y}}[\xi], N_p(\xi) \in \mathbb{R}^{\bullet \times \mathtt{u}}[\xi]$ being co-prime polynomials, and $0_\bullet$, and $I_\bullet$ representing the zero matrix, and the identity matrix of suitable dimension. From the input/output point of view, y is considered as the output of the plant and u as the input. With this partition of inputs and outputs, together with definition 2.17, evidently $D_p(\xi)^{-1} N_p(\xi) = G(\xi)$ defines a proper rational matrix with $D_p(\xi) \neq 0$. In a similar way, the behavior of the controller $\Sigma_{\mathcal{C}}$ is given by

$$\mathcal{C} = \left\{ c \in \mathbb{S}^{\mathbb{T}} \middle| C\left(\frac{d}{dt}\right) c = 0 \right\}, \tag{5.3}$$

$$\text{where} \quad C(\xi) = \begin{bmatrix} N_c(\xi) & -D_c(\xi) \end{bmatrix} \tag{5.4}$$

with $D_c(\xi) \in \mathbb{R}^{\bullet \times \mathtt{u}}[\xi]$, $N_c(\xi) \in \mathbb{R}^{\bullet \times \mathtt{y}}[\xi]$ being co-prime polynomials, and $D_c(\xi)^{-1} N_c(\xi) = H(\xi)$ representing a proper rational matrix with $D_c(\xi) \neq 0$. In this controller configuration, u is the output of the controller, and e is the input. Whenever the above two systems interconnects, the controller impose some restrictions on the behavior of the plant. The imposed restrictions on $\mathcal{P}$ by $\mathcal{C}$ yields the full controlled behavior, which is given by

$$\mathcal{K}_{\mathtt{full}} = \{ s = \mathtt{col}(w, c) \,| (w, c) \in \mathcal{P} \text{ and } c \in \mathcal{C} \} \tag{5.5}$$

In terms of kernel representations, the interconnected or controlled behavior $\mathcal{K}_{\texttt{full}}$ is represented by

$$\mathcal{K}_{\texttt{full}} \equiv \begin{bmatrix} R\left(\frac{d}{dt}\right) & -M\left(\frac{d}{dt}\right) \\ 0_{\texttt{w}} & C\left(\frac{d}{dt}\right) \end{bmatrix} \begin{bmatrix} w \\ c \end{bmatrix} = 0. \tag{5.6}$$

Clearly,

$$\mathtt{p}(\mathcal{K}_{\texttt{full}}) = \mathtt{p}(\mathcal{P}) + \mathtt{p}(\mathcal{C}).$$

The last equality demonstrates that the interconnection between $\mathcal{P}$ and $\mathcal{C}$ in the feedback configuration is always a regular interconnection.

5.1.2 Problem Formulation

For the case of partial interconnection, generally, the interest lies in controlling the behavior of the manifest variables in the controlled system. This is achieved by imposing some restrictions through the control variables. The controlled behavior in terms of the manifest variables in the full interconnected system, defined in (5.5), can be obtained by using the elimination theorem, which is given as

$$\mathcal{K} = \{w \in \mathbb{S}^{\mathbb{T}} | \exists c \in \mathcal{C} \text{ such that } (w, c) \in \mathcal{P}\}. \tag{5.7}$$

The real-time problem of controlling a faulty system is that the operating plant should achieve the control objectives at anytime, i.e. regardless of any occurrence of a fault. In this respect, we can single out a subset of plants' behavior as desirable. We call it the desired behavior, denoted by $\mathcal{D}$, which is provided by an effective FMEA analysis that aims at providing a complete coverage of possible occurring faults into the system as well as the achievable desired behavior. An approach to perform this analysis procedure is presented in [MJL08]. The behavior $\mathcal{D}$ can be considered as equivalent to the control objective $\mathfrak{O}$ since the solution set satisfying the control objectives also belongs to the desired behavior. The desired behavior will, indeed, be defined in terms of the manifest variables, which is given by

$$\mathcal{D} = \left\{ w \in \mathbb{S}^{\mathbb{T}} | D\left(\frac{d}{dt}\right) w = 0 \right\}, \tag{5.8}$$

$$\text{where } D(\xi) = \begin{bmatrix} D_r(\xi) & -D_y(\xi) \end{bmatrix}. \tag{5.9}$$

with $D_r(\xi) \in \mathbb{R}^{\bullet \times \mathtt{r}}[\xi], D_y(\xi) \in \mathbb{R}^{\bullet \times \mathtt{y}}[\xi]$ as the co-prime polynomials, and $D_y(\xi)^{-1} D_r(\xi)$ representing a set of proper rational matrices with $D_y(\xi) \neq 0$.

With the above facts, the real-time FTC problem we are dealing with can now be posed in the following way. Given a vector space of time signals $\mathbb{S}^{\mathbb{T}}$, and the desired behavior $\mathcal{D}$, the problem is to "synthesize" an appropriate controller $\mathcal{C}$, without using any *a priori* knowledge of the model of the plant in real-time, which have the suitable control actions such that the controlled behavior $\mathcal{K}$ satisfy the desired behavior $\mathcal{D}$ at anytime.

5.1.3 Design and Implementation of Real-time FTC via partial interconnection

The implementability of the desired behavior plays a key role in an online design of the controller. Otherwise, if the desired behavior is not achievable or not implementable, then no controller exists that can guarantee the fault tolerance. Roughly speaking, the faults for which the desired behavior is not implementable can be termed as "intolerable faults". To support the implementability of $\mathcal{D}$, we state the "Willems' Theorem" [WT02].

Theorem 5.1 (Willems' Theorem)**.** *Let $\mathcal{P}$ be a behavior of the plant, and let $\mathcal{D}$ be a desired behavior. Then the following statements are equivalent:*

(i). $\mathcal{D}$ is achievable or implementable with respect to the plant.

(ii). These exists a controller $\mathcal{C}$ that implements $\mathcal{D}$.

(iii). $\mathcal{N} \subseteq \mathcal{D} \subseteq \mathcal{P}_w$.

□

Based on the above implementability theorem, van der Schaft [vdS03] gives a "general behavioral description" of the existing controller, irrespective of any particular control configuration, that can implements the desired behavior.

Theorem 5.2. *Let $\mathcal{P}$ be a behavior of the plant, and let $\mathcal{D}$ be the implementable desired behavior. Then the controller, defined as*

$$\mathcal{C} = \{c \in (\mathbb{R}^{\mathtt{c}})^{\mathbb{R}} | \exists \tilde{w} \text{ such that } (\tilde{w}, c) \in \mathcal{P} \text{ and } \tilde{w} \in \mathcal{D}\}, \tag{5.10}$$

implements the desired behavior $\mathcal{D}$. □

The controller defined in theorem 5.2 is termed as the canonical controller. Basically, this controller is constructed by the interconnection of the plant (with reversed terminal) and the desired behavior. Pictorially, the idea of constructing this controller is shown in Fig. 5.3.

For determining the kernel representation of the above controller $\mathcal{C}$, we will now use the implementability theorem. From the first inclusion of theorem 5.1, i.e. $\mathcal{N} \subseteq \mathcal{D}$, there exists a polynomial matrix, say $L(\xi)$ such that

$$D(\xi) = L(\xi)R(\xi) \tag{5.11}$$

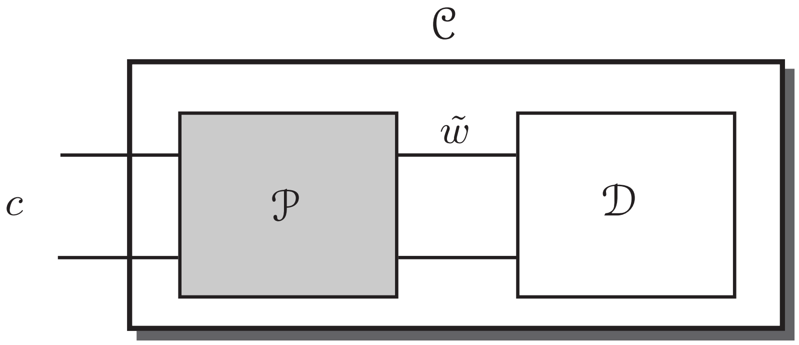


Figure 5.3: The canonical controller

where $\mathcal{D} = \mathtt{ker}(D(\xi))$, and $\mathcal{N} = \mathtt{ker}(R(\xi))$. The full behavior of the plant is given by the following kernel representation $R(\xi)w = M(\xi)c$. Pre-multiplying the last differential equation by $L(\xi)$, we get $L(\xi)R(\xi)w = L(\xi)M(\xi)c$. From the above, it follows that $D(\xi)w = L(\xi)R(\xi)w = 0$. This yields the kernel representation of the canonical controller, which is given by

$$\mathcal{C} \equiv L(\xi)M(\xi)c = 0. \tag{5.12}$$

From the above, clearly the controller is constructed for general systems without imposing any *realizability* requirements. This issue is of utmost practical importance for a possible implementation of the controller in the closed-loop. Theoretically, in [JWBT05, Theorem 16], the so-called regularity of interconnection is imposed for the design of the canonical controller. By construction, the control configuration considered in this thesis guarantees that whatever be the controller, it will always make a regular interconnection with the plant. However, giving a closer look to the kernel representation of the controller given in (5.12), it includes the knowledge of the plant embedded within the $M(\xi)$ matrix, which has to be available in real-time while synthesizing an online controller. As we mentioned before, we do not have any *a priori* information about the plant's model in real-time, i.e. $R(\xi)$ and $M(\xi)$ matrices are not available during the controller reconfiguration process, therefore, we cannot use the above equation to compute the controller's polynomials.

The main result of this section is given in the following proposition where we directly compute the controller polynomials using the real-time measurements observed from the plant. First, we define the "filtered" plant signals, denoted by $(\bar{u}, \bar{y})$, which are given as

$$\bar{u} = D_r(\xi)u, \quad \bar{y} = (D_r(\xi) - D_y(\xi))y, \tag{5.13}$$

together with polynomials $D_r(\xi), D_y(\xi)$ considering to take the form as $D_r(\xi) = d_r(\xi)I_{\mathtt{r}}, D_y(\xi) = d_y(\xi)I_{\mathtt{y}}$, where $d_r(\xi) \in \mathbb{R}^{1\times 1}[\xi], d_y(\xi) \in \mathbb{R}^{1\times 1}[\xi]$.

Proposition 5.1. *Given a vector space of time-dependent signals* $(\mathbb{T} \times \mathbb{S})$*, the implementable desired behavior* $\mathcal{D}$*, for any closed-loop controller* $\mathcal{C}$ *if an unknown fault occurs into the system then the following statements are equivalent:*

(i). The system is a real-time fault-tolerant control system.

(ii). The trajectories $(\bar{u}, \bar{y})$ *belongs to the controller* $\mathcal{C}$*, which is equivalent to saying that the following differential equation holds.*

$$N_c(\xi)\bar{y} + D_c(\xi)\bar{u} = 0. \tag{5.14}$$

Proof. $(i) \implies (ii)$: Since the desired behavior $\mathcal{D}$ is implementable, from theorem 5.1 it follows that there exists a controller $\mathcal{C}$ that implements $\mathcal{D}$. Therefore, we now only required to synthesize the kernel representation of that controller $\mathcal{C}$ without using any *a priori* information of the plant's model to guarantee the fault tolerance. For the considered feedback configuration, substitute the explicit kernel representation of $\mathcal{D}$ and $\mathcal{N}$ into (5.11), which gives

$$\begin{bmatrix} D_r(\xi) & -D_y(\xi) \end{bmatrix} = L(\xi) \begin{bmatrix} I_{\mathtt{r}} & -I_{\mathtt{y}} \\ 0_{\mathtt{r}} & D_p(\xi) \end{bmatrix} \tag{5.15}$$

In the sequel, the dimension of the identity and zero matrix will be avoided whenever it is clear from context. The matrix $R(\xi)$ can be factorized as

$$\begin{bmatrix} I & -I \\ 0 & D_p(\xi) \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & D_p(\xi) \end{bmatrix} \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}. \tag{5.16}$$

Putting (5.16) in (5.15), we get

$$\begin{bmatrix} D_r(\xi) & -D_y(\xi) \end{bmatrix} = L(\xi) \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & D_p(\xi) \end{bmatrix} \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}.$$

Since the matrix $D_p(\xi)$ is invertible, so we can write the last equation in the following form

$$\begin{bmatrix} D_r(\xi) & -D_y(\xi) \end{bmatrix} = L(\xi) \begin{bmatrix} D_p(\xi) & 0 \\ 0 & D_p(\xi) \end{bmatrix} \begin{bmatrix} D_p^{-1}(\xi) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}$$

$$\begin{bmatrix} D_r(\xi) & -D_y(\xi) \end{bmatrix} \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} D_p^{-1}(\xi) & 0 \\ 0 & I \end{bmatrix}^{-1} = L(\xi) \begin{bmatrix} D_p(\xi) & 0 \\ 0 & D_p(\xi) \end{bmatrix}$$

$$\begin{bmatrix} D_r(\xi)D_p(\xi) & D_r(\xi) - D_y(\xi) \end{bmatrix} = L(\xi) \begin{bmatrix} D_p(\xi) & 0 \\ 0 & D_p(\xi) \end{bmatrix}.$$

Here, knowing that the matrix $\begin{bmatrix} D_p(\xi) & 0 \\ 0 & D_p(\xi) \end{bmatrix}$ is a diagonal matrix, we can write the right hand side of the above equation as $\begin{bmatrix} D_p(\xi) & 0 \\ 0 & D_p(\xi) \end{bmatrix} L(\xi)$ and assign it to $L'(\xi)$. From (5.12), it follows that

$$L(\xi)M(\xi)c = 0 \implies L'(\xi)M(\xi)c = 0.$$

Accordingly, the kernel representation of the controller (still in terms of plant's parameters) can be written as

$$\mathcal{C} \equiv L'(\xi)M(\xi)c = 0. \tag{5.17}$$

Writing it explicitly, we have

$$\begin{bmatrix} D_r(\xi)D_p(\xi) & D_r(\xi) - D_y(\xi) \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & N_p(\xi) \end{bmatrix} \begin{bmatrix} e \\ u \end{bmatrix} = 0 \tag{5.18}$$

$$D_r(\xi)D_p(\xi)e + (D_r(\xi) - D_y(\xi))N_p(\xi)u = 0. \tag{5.19}$$

Pre-multiply the last equation by $D_p^{-1}(\xi)$, and re-arranging it yields

$$D_r(\xi)e + (D_r(\xi) - D_y(\xi))D_p^{-1}(\xi)N_p(\xi)u = 0. \tag{5.20}$$

From the structure of the feedback configuration, we have the relation $y = D_p^{-1}(\xi)N_p(\xi)u$. Also, $e = N_c^{-1}(\xi)D_c(\xi)u$. From the above, we obtain

$$N_c^{-1}(\xi)D_c(\xi)D_r(\xi)u + (D_r(\xi) - D_y(\xi))y = 0. \tag{5.21}$$

Pre-multiplying above by $N_c(\xi)$, it gives

$$D_c(\xi)D_r(\xi)u + N_c(\xi)(D_r(\xi) - D_y(\xi))y = 0 \Leftrightarrow N_c(\xi)\bar{y} + D_c(\xi)\bar{u} = 0. \tag{5.22}$$

$(ii) \implies (i)$: The proof of this implication is trivial, which can be obtained by substituting the filtered plant signals (5.13) in the kernel representation of the controller $\mathcal{C}$ (5.3). □

One of the deep consequences of the above proposition is that the observed signals $\bar{w} = \texttt{col}(\bar{u}, \bar{y})$ is independent of any particular setting in the feedback configuration, i.e. one can collect these signals with any arbitrary controller working in the closed-loop. Therefore, the controller synthesized in the above manner is a pure “data-driven online controller”, i.e. a controller which is directly synthesized without a mathematical model of the plant but solely on the basis of the desired behavior and the any experimental input/output data produced by the plant. In this way, we can use the signal $\bar{w}$, which amounts to the measurements of the physical plant signals $\mathtt{col}(u, y)$, to design an online controller "which-when" makes an interconnection with the plant subject to faults yields the desired behavior. Interestingly, solving the equation (5.14) is a continuous-time system identification problem, which can be solved using various methods listed in the literature [GW08] [see Appendix B]. Note that on fixing (or if we know) the degree of controller's polynomials at the outset, the controller synthesized using the tools borrowed from the system identification community becomes an approximated controller that implements the desired behavior.

5.2 Fault-tolerant Control: full interconnection case

In this section, we shall present the online controller design strategy to guarantee the fault tolerance using the case of full interconnection. Similar to the above, no *a priori* knowledge will be used to design an FTC controller in real-time.

5.2.1 Feedback Interconnection within the full interconnection

Here we deal with a more general type of feedback interconnection, in which the controller has two degrees of freedom (abbreviated as 2DOF). The degree of freedom of a control system is defined as the number of closed-loop input-output maps that can be adjusted independently. Various works on 2DOF controllers have been reported in the literature, which demonstrated naturally the advantages of 2DOF control system over a one degree of freedom control system [AT03]. A general form of the 2DOF control system is illustrated in Fig. 5.4, where the controller scheme consists of two compensators C_r and C_y.

Taking the behavioral point of view, the above structure can be illustrated by Fig. 5.5, where we have access to only s-trajectory, i.e. $s = \mathtt{col}(r, y, u)$. In this figure, $\mathcal{P} \in \mathfrak{L}^{\mathtt{s}}$ denotes the behavior of the plant and $\mathcal{C} \in \mathfrak{L}^{\mathtt{s}}$ denotes the behavior of the controller in terms of the system variable s, whose values lies in the signal space $\mathbb{S}$ having the dimension $\mathtt{s} = \mathtt{r} + \mathtt{y} + \mathtt{u}$. We can now define the trajectory-based dynamical system for the plant and the controller as $\Sigma_{\mathcal{P}} = (\mathbb{T}, \mathbb{S}, \mathcal{P})$, and $\Sigma_{\mathcal{C}} = (\mathbb{T}, \mathbb{S}, \mathcal{C})$ respectively, where $\mathbb{T} \subseteq \mathbb{R}$, $\mathbb{S} \subseteq \mathbb{R}^{\mathtt{r}+\mathtt{y}+\mathtt{u}}$, $\mathcal{P} \subseteq \mathbb{S}^{\mathbb{T}}$, and their behaviors in the following way.

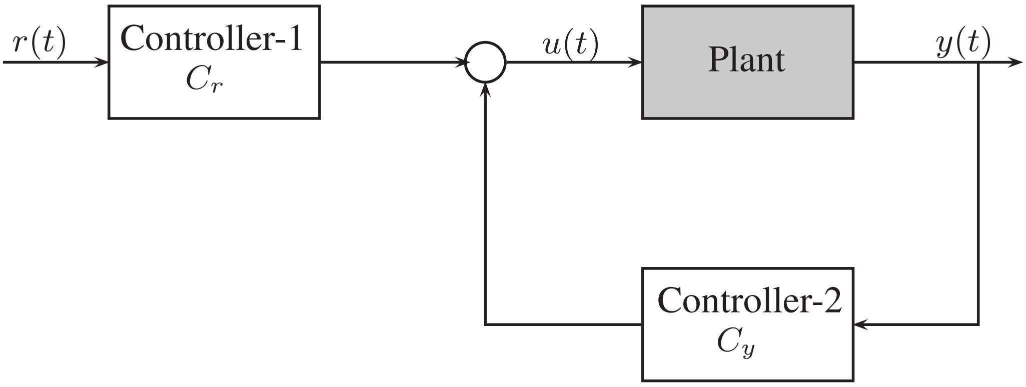


Figure 5.4: 2DOF feedback control in the classical sense

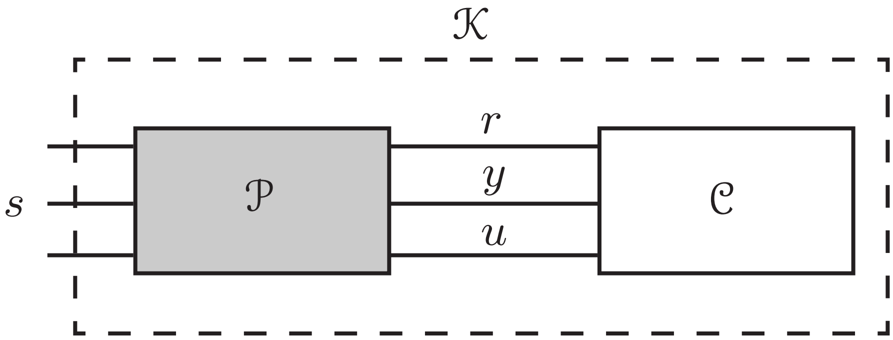


Figure 5.5: 2DOF feedback control in the behavioral sense

$$\mathcal{P} = \left\{ s = \mathtt{col}(r, y, u) \in \mathbb{S}^{\mathbb{T}} \middle| R\left(\frac{d}{dt}\right) s = 0 \right\}, \tag{5.23}$$

where $R(\xi) = \begin{bmatrix} 0_{\mathtt{r}} & D_p(\xi) & -N_p(\xi) \end{bmatrix}$ with $D_p(\xi) \in \mathbb{R}^{\bullet \times \mathtt{y}}[\xi], N_p(\xi) \in \mathbb{R}^{\bullet \times \mathtt{u}}[\xi]$ being the co-prime polynomials, and $0_{\mathtt{r}}$ representing the zero matrix of dimension $\mathtt{r}$. The trajectory y is considered as the output of the plant and u as the input. With this input/output partition, evidently $D_p(\xi)^{-1}N_p(\xi) = G(\xi)$ defines a proper rational matrix with $D_p(\xi) \neq 0$ [Wil91, Section VIII]. In a similar way, the behavior of the controller $\Sigma_{\mathcal{C}}$ is given by

$$\mathcal{C} = \left\{ s = \mathtt{col}(r, y, u) \in \mathbb{S}^{\mathbb{T}} \middle| C\left(\frac{d}{dt}\right) s = 0 \right\}, \tag{5.24}$$

where $C(\xi) = \begin{bmatrix} N_{c_r}(\xi) & -N_{c_y}(\xi) & -D_c(\xi) \end{bmatrix}$ with $D_c(\xi) \in \mathbb{R}^{\bullet \times \mathtt{u}}[\xi], N_{c_r}(\xi) \in \mathbb{R}^{\bullet \times \mathtt{r}}[\xi], N_{c_y}(\xi) \in \mathbb{R}^{\bullet \times \mathtt{y}}[\xi]$ being co-prime polynomials, and $D_c(\xi)^{-1}N_{c_r}(\xi) = C_r(\xi), D_c(\xi)^{-1}N_{c_y}(\xi) = C_y(\xi)$ representing the proper rational matrices with common denominator $D_c(\xi) \neq 0$. Whenever the above two systems interconnects, the controller impose some restrictions on the behavior of the plant. The interconnection of $\mathcal{P}$ and $\mathcal{C}$ through the shared variable s results in a system in which these variables satisfy the dynamics of both $\mathcal{P}$ and $\mathcal{C}$. The behavior of this interconnected system is termed as the controlled behavior or the *implemented behavior* $\mathcal{K}$, defined as $\mathcal{K} = \mathcal{P} \cap \mathcal{C} \in \mathfrak{L}^{\mathtt{s}}$, which is equivalent to

$$\mathcal{K} = \{ s = \mathtt{col}(r, y, u) \,| s \in \mathcal{P} \text{ and } s \in \mathcal{C} \},$$

where the symbol '$\cap$' denotes the interconnection operation.

Definition 5.1. (Implementability) *Let $\mathcal{P} \in \mathfrak{L}^{\mathtt{s}}$ be a linear differential system, $\mathcal{C} \in \mathfrak{L}^{\mathtt{s}}$ be a controller, and $\mathcal{K} \in \mathfrak{L}^{\mathtt{s}}$. Whenever $\mathcal{K}$ is obtained by interconnecting $\mathcal{P}$ and $\mathcal{C}$, then we say "$\mathcal{C}$ implements $\mathcal{K}$". In addition, for a given $\mathcal{K} \in \mathfrak{L}^{\mathtt{s}}$ whenever there exists $\mathcal{C} \in \mathfrak{L}^{\mathtt{s}}$ such that $\mathcal{C}$ implements $\mathcal{K}$ then $\mathcal{K}$ is said to be implementable by the interconnection.*

The above definition implies that $\mathcal{K}$ is the restricted behavior satisfying the dynamics of both $\mathcal{P}$ and $\mathcal{C}$. Accordingly, a given $\mathcal{K} \in \mathfrak{L}^{\mathtt{s}}$ is implementable by an interconnection with respect to $\mathcal{P}$ if and only if $\mathcal{K} \subseteq \mathcal{P}$ [WT02].

5.2.2 Design and Implementation of Real-time FTC via full interconnection

The real-time notion of controlling a faulty system is that the operating plant must achieve the control objectives at anytime, i.e. regardless of any occurrence of a fault. In this respect, we can single out a subset of plants' behavior as desirable. This desirable behavior is provided by FMEA analysis, whose objective is to forecast systematically how fault effects in elements relate to faults at inputs, or outputs within the elements, and what reactions should be imposed on the system whenever a certain faults appears. FMEA provides a complete coverage of possible occurring faults into the system as well as of the desired behavior $\mathcal{D} \in \mathfrak{L}^{\mathtt{s}}$ capturing the control objectives. In the case of full interconnection, we define the behavior $\mathcal{D}$ as

$$\mathcal{D} = \left\{ s = \mathtt{col}(r, y, u) \in \mathbb{S}^{\mathbb{T}} \middle| D\left(\frac{d}{dt}\right) s = 0 \right\}, \tag{5.25}$$

$$\text{where} \quad D(\xi) = \begin{bmatrix} N_{T_y}(\xi) & -D_T(\xi) & 0_{\mathtt{u}} \\ N_{T_u}(\xi) & 0_{\mathtt{y}} & -D_T(\xi) \end{bmatrix}.$$

in which $D_T(\xi), N_{T_y}(\xi), N_{T_u}(\xi)$ are the co-prime polynomials with $D_T(\xi) \neq 0$. The system is considered to have the same dimension, i.e. $\mathtt{r} = \mathtt{y} = \mathtt{u}$. Proceeding with designing an online controller, first we define two filtered plant signals as

$$\bar{w} = N_{T_y}(\xi)(-N_{T_u}(\xi)y + N_{T_y}(\xi)u), \quad \bar{z} = N_{T_u}(\xi)(N_{T_u}(\xi)y - N_{T_y}(\xi)u), \tag{5.26}$$

together with polynomials $N_{T_y}(\xi), N_{T_u}(\xi), D_T(\xi)$ considering to be the diagonal matrices.

Proposition 5.2. *Given a vector space of time-dependent signals* $(\mathbb{T} \times \mathbb{S})$*, the implementable desired behavior* $\mathcal{D}$*, for any unknown occurred fault into the system, if the controller* $\mathcal{C}$ *is designed using* $\mathcal{C} \equiv V(\xi)D(\xi)s = 0$ *with* $V = \begin{bmatrix} v_1 & v_2 \end{bmatrix}$*, where the polynomials* $v_1(\xi)$ *and* $v_2(\xi)$ *are computed by*

$$v_1\bar{w} + v_2\bar{z} = 0 \tag{5.27}$$

then the closed-loop system is a real-time fault-tolerant system.

Proof. First, we will show that if $\mathcal{D}$ is implementable then there exists a controller $\mathcal{C}$ that implements $\mathcal{D}$. Later, we will derive the explicit relation to compute the controller polynomials. From the inclusion $\mathcal{D} \subseteq \mathcal{P}$, there exists a polynomial matrix $F(\xi)$ such that $R(\xi) = F(\xi)D(\xi)$ with $F = \begin{bmatrix} f_1 & f_2 \end{bmatrix}$. Let $V(\xi)$ be a polynomial matrix such that the matrix $\mathtt{col}(F, V)$, with $V = \begin{bmatrix} v_1 & v_2 \end{bmatrix}$, is unimodular. Define the controller as

$$\mathcal{C} \equiv VDs = 0. \tag{5.28}$$

The controller $\mathcal{C}$ defined in (5.28) has to make an interconnection with $\mathcal{P}$. The controlled behavior $\mathcal{K}$ is then given by

$$\mathcal{K} = \mathcal{P} \cap \mathcal{C} = \mathtt{ker}\left(\begin{bmatrix} R \\ VD \end{bmatrix}\right). \tag{5.29}$$

Since $\mathcal{D}$ is implementable, (5.29) yields

$$\mathcal{K} = \texttt{ker}\left(\begin{bmatrix} F \\ V \end{bmatrix} D\right).$$

By assumption, $\texttt{column}(F, V)$ is unimodular. From the section 2.3, we conclude, $\mathcal{K} = \mathcal{D}$. This proves that the controller $\mathcal{C}$ defined in (5.28) implements the desired behavior $\mathcal{D}$. Writing (5.24) explicitly, we have

$$\begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} N_{T_y} & -D_T & 0 \\ N_{T_u} & 0 & -D_T \end{bmatrix} \begin{bmatrix} r \\ y \\ u \end{bmatrix} = 0. \tag{5.30}$$

From the above equation, it is only required to compute the polynomials $v_1(\xi)$ and $v_2(\xi)$ using the filtered plant signals $(\bar{w}, \bar{z})$. Since the reference trajectory is an external signal and does not involve in capturing the plant's behavior, in the following we shall eliminate the variable r from (5.30). Simplifying (5.30), we get

$$\begin{bmatrix} v_1 N_{T_y} + v_2 N_{T_u} & -v_1 D_T & -v_2 D_T \end{bmatrix} \begin{bmatrix} r \\ y \\ u \end{bmatrix} = 0.$$

From the above, we know that $\mathcal{D} \equiv D(\xi)s = 0 \implies V(\xi)D(\xi)s = 0 \equiv \mathcal{C}$. Therefore, the trajectories belonging to the desired behavior also satisfy the above equation. Further, we can write the last equation as

$$\begin{bmatrix} v_1 N_{T_y} + v_2 N_{T_u} & -v_1 D_T & -v_2 D_T \\ N_{T_y} & -D_T & 0 \\ N_{T_u} & 0 & -D_T \end{bmatrix} \begin{bmatrix} r \\ y \\ u \end{bmatrix} = 0. \tag{5.31}$$

Pre-multiply the above equation by the matrix $\begin{bmatrix} N_{T_y} & -v_2 N_{T_u} & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}$ it yields

$$\begin{bmatrix} N_{T_y} v_1 N_{T_y} & (-N_{T_y} v_1 + v_2 N_{T_u}) D_T & -N_{T_y} v_2 D_T \\ N_{T_y} & -D_T & 0 \\ N_{T_u} & 0 & -D_T \end{bmatrix} \begin{bmatrix} r \\ y \\ u \end{bmatrix} = 0. \tag{5.32}$$

Clearly, the trajectory s belonging to the behavior described by (5.31) implies that s belongs to the behavior described by (5.32) (see section 2.3). Now, pre-multiply the above equation by the matrix $\begin{bmatrix} N_{T_u} & 0 & -N_{T_y} v_1 N_{T_y} \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}$ it yields

$$\begin{bmatrix} 0 & -N_{T_u}(N_{T_y} v_1 - v_2 N_{T_u}) D_T & -N_{T_u} N_{T_y} v_2 D_T + N_{T_y} v_1 N_{T_y} D_T \\ N_{T_y} & -D_T & 0 \\ N_{T_u} & 0 & -D_T \end{bmatrix} \begin{bmatrix} r \\ y \\ u \end{bmatrix} = 0. \tag{5.33}$$

The polynomials in the controllers' kernel representation can now be evaluated by the first row of the above equation, given as

$$\begin{bmatrix} -N_{T_u}(N_{T_y}v_1 - v_2N_{T_u}) & -N_{T_u}N_{T_y}v_2 + N_{T_y}v_1N_{T_y} \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix} = 0. \tag{5.34}$$

with $\begin{bmatrix} D_T & 0 \\ 0 & D_T \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix} = 0$ being the stable autonomous behavior. Equivalently (5.34) can be written as

$$-N_{T_u}N_{T_y}v_1y + v_2N_{T_u}y - N_{T_u}N_{T_y}v_2u + N_{T_y}v_1N_{T_y}u = 0 \tag{5.35}$$

Re-arranging the last equation, it gives

$$v_1[N_{T_y}(-N_{T_u}y + N_{T_y}u)] + v_2[N_{T_u}(N_{T_u}y - N_{T_y}u)] = 0 \tag{5.36}$$

Thus, we conclude that $v_1\bar{w} + v_2\bar{z} = 0$. □

Similar to the previous section, solving the last equation now becomes solving a continuous time system identification problem, which can be solved using various methods listed in the literature [GW08]. Consequently, we obtain the polynomials $v_1(\xi)$ and $v_2(\xi)$ directly from the filtered plant signals, which yields an online controller $\mathcal{C}$ implementing the desired behavior $\mathcal{D}$ at anytime.

Part III

Practical Aspects

Chapter 6

Real-time Smooth Interconnection in the behavioral context

Contents

6.1 Limitations in AFTC systems 103

6.2 Basic Cause of Switching Transients 104

6.3 Guaranteeing the Smooth Interconnection 108

The prime aim of a fault-tolerant control system is to maintain the system performance, defined in term of the desired behavior at anytime, i.e. even after an occurrence of a fault. In Part II of this thesis, we have presented novel real-time controller reconfiguration mechanisms to deal with the aforementioned issue. These systems are termed as Active Fault-Tolerant Systems (AFTCS). In this context, specifications for the system performance fall into three durations of overall system operations [ZJ08]:

1. fault-free period;
2. transient period during the reconfiguration; and
3. steady-state period after the reconfiguration.

As demonstrated earlier, we have guaranteed that the system satisfy the desired behavior during the first, and the third course of operations. In this chapter, we shall present a novel real-time algorithm to guarantee the systems' performance during the transient period of controller reconfiguration mechanism as well without using any *a priori* knowledge of the plant's model. This chapter is composed of the works published in [JYS13c, JYS12e].

6.1 Limitations in AFTC systems

An active FTCS reacts to system component malfunctions (including actuators, the system itself, and sensors) by reconfiguring the controller based on the real-time information. The term 'active' represents corrective actions taken actively by the reconfiguration mechanism to adapt the control system in response to the occurring faults. Based on AFTCS architecture, the design objectives of an active FTCS are [JY12] :

1. to reconfigure the existing control scheme effectively to achieve stability and acceptable closed-loop system performance; and
2. to commission the reconfigured controller "smoothly" into the system by minimizing potential switching transients.

In an active FTCS, a newly reconfigured controller has to be switched into replacing the pre-fault controller. The switching may cause undue transients. From the practical implementation point of view, the transients may be harmful to the safe operations of the system as they could cause saturations in actuators, and at worst, it may damage the components within the system. A comparative study has been done in [JY12] in which it is shown that the undesired transients shocks the system in some sense. Therefore, the appearance of such transients should be minimized as much as possible.

On the other hand, considering the above issue particularly, Passive Fault-Tolerant Control Systems (PFTCS) appear more attractive. In a passive FTCS the controller, once designed, does not need to be changed or reconfigured during the course of operation. In practice, a passive FTCS has a simple structure and has no controller switching associated transients. Therefore, the additional real-time computational demand is low for a passive FTCS. Since no switching is involved in a passive FTCS, the behavior of the system is much smoother than that of an active FTCS. Furthermore, since the passive FTCS does not require any FDD unit, there is no delay between the fault occurrence and the corresponding control actions. However, a passive FTCS is designed with the consideration of both normal system operation and design basis faults. Compared with an active approach, the performance achieved by a passive FTCS can never be optimal for all design scenarios. If one attempts to design a passive FTCS to accommodate an excessive number of faults, the overall conservatism increases. No controller may be found to satisfy all the design requirements. A passive FTCS is less flexible and has limited fault-tolerant capabilities, especially in the case of beyond design basic failures.

6.2 Basic Cause of Switching Transients

Consider the scenario of controller reconfiguration as illustrated in Fig. 6.1, which demonstrates that the online controller during the time-interval $[0, t_{inter})$

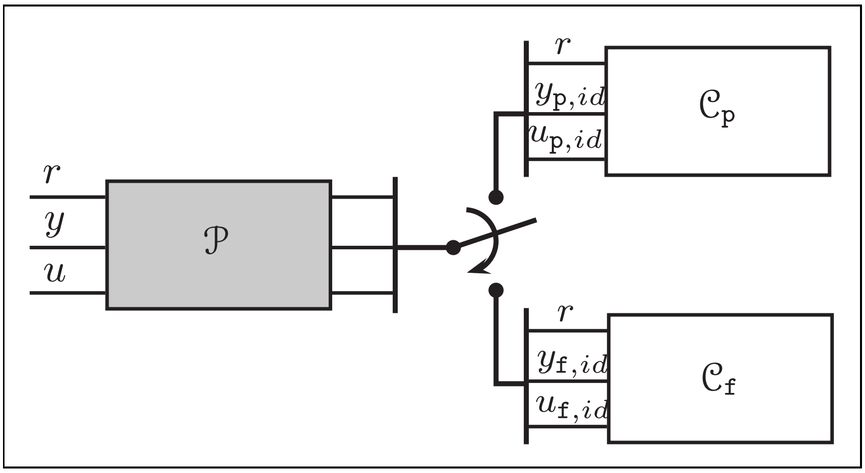


Figure 6.1: Switching from controller 1 to controller 2: controller reconfiguration

is controller $\mathcal{C}_{\mathtt{p}}$ and at time t_{inter}, the second controller $\mathcal{C}_{\mathtt{f}}$ makes an interconnection with the unknown plant, where the subscript p and f denotes the past and the future controller respectively. Generally, the above operation is done so that the closed-loop with controller $\mathcal{C}_{\mathtt{f}}$ can satisfy the desired behavior $\mathcal{D}$. We can view this as two individuals interconnected controlled behaviors, defined as

$$\mathcal{K}_{\mathtt{pic}} = \{(r, y_{\mathtt{p},id}, u_{\mathtt{p},id}) \in \mathbb{S}^{\mathbb{T}} | (r, y_{\mathtt{p},id}, u_{\mathtt{p},id}) \in \mathcal{P} \text{ and } (r, y_{\mathtt{p},id}, u_{\mathtt{p},id}) \in \mathcal{C}_{\mathtt{p}}\} \quad \forall t < t_{inter}, \tag{6.1}$$

$$\mathcal{K}_{\mathtt{fic}} = \{(r, y_{\mathtt{f},id}, u_{\mathtt{f},id}) \in \mathbb{S}^{\mathbb{T}} | (r, y_{\mathtt{f},id}, u_{\mathtt{f},id}) \in \mathcal{P} \text{ and } (r, y_{\mathtt{f},id}, u_{\mathtt{f},id}) \in \mathcal{C}_{\mathtt{f}}\} \quad \forall t \geq t_{inter}, \tag{6.2}$$

where the subscript pic and fic on the controlled behavior denotes the past interconnected system and future interconnected system respectively.

Definition 6.1 (Real-time Smooth Interconnection)**.** *The interconnection is said to be a real-time smooth interconnection whenever $\mathcal{K}_{\mathtt{fic}} = \mathcal{D}$ at the time of interconnection, i.e. at time $t = t_{inter}$.* □

In fact, the switching phenomenon shown in Fig. 6.1 illustrates the concatenation of the past behavior with the future behavior. Assume that no transients appear in the closed-loop when the controller $\mathcal{C}_{\mathtt{f}}$ is switched in the loop. Consequently, the output signal of this so-called switched mode system is obtained by concatenating ideal (id) output signal $y_{\mathtt{p},id}$ on $[0, t_{inter})$ with signal $y_{\mathtt{f},id}$ on $[t_{inter}, \infty)$, that is,

$$y(t) = (y_{\mathtt{p},id} \bigtriangleup^{t_{inter}} y_{\mathtt{f},id})(t) = \begin{cases} y_{\mathtt{p},id}(t) & \text{for } t \in [0, t_{inter}) \\ y_{\mathtt{f},id}(t) & \text{for } t \geq t_{inter}, \end{cases} \tag{6.3}$$

Similarly, the ideal input signal to the plant in this switched-mode system is given by

$$u(t) = u_{id}(t) = (u_{\mathtt{p},id} \bigtriangleup^{t_{inter}} u_{\mathtt{f},id})(t) \quad \text{for } t \in [0, \infty), \tag{6.4}$$

where $D_p(\xi)y_{\mathtt{p}}(t) = N_p(\xi)u_{\mathtt{p},id}(t)$, and $D_p(\xi)y_{\mathtt{f}}(t) = N_p(\xi)u_{\mathtt{f},id}(t)$ with $N_p(\xi)$, $D_p(\xi)$ representing the polynomials of the plant. A key feature of such ideal control u_{id} is that its segment on $[t_{inter}, \infty)$, i.e., trajectory $u_{\mathtt{f},id}$ from t_{inter} to ∞, is issued from the controller $\mathcal{C}_{\mathtt{f}}$ which has been constantly connected to the unknown plant in closed-loop. Looking at $t = t_{inter}$, the trajectories $(y_{\mathtt{p},id}, u_{\mathtt{p},id}) \in \mathcal{P}$ but there is no $r-$trajectory in the signal space $(\mathbb{R}^{\mathtt{r}})^{\mathbb{R}}$ for all $t < t_{inter}$ that satisfies $(r, y_{\mathtt{p},id}, u_{p,id}) \in \mathcal{C}_{\mathtt{f}}$. It is some $r-$trajectory together with $(y_{\mathtt{f},id}, u_{\mathtt{f},id}) \in \mathcal{P}$ that satisfy $(r, y_{\mathtt{f},id}, u_{\mathtt{f},id}) \in \mathcal{C}_{\mathtt{f}}$. Therefore, at $t = t_{inter}$, $(r, y_{\mathtt{p},id}, u_{\mathtt{p},id}) \notin \mathcal{D}$. This results in a non-smooth interconnection due to which undesirable transients appear in the closed-loop at the time of interconnection. The above development illustrates the fact that the controller $\mathcal{C}_{\mathtt{f}}$ is connected to the plant at t_{inter} only, and consequently, its *history* on $[0, t_{inter})$ has no relation with the history of the plant in that interval. Note that, as mentioned in Chapter 2, it is the “state-trajectory” that keeps the history of the system.

Now, we shall take a more precise look in terms of the state-trajectory that maintains the history of the closed-loop dynamics. Recall the equivalent kernel representation introduced in Chapter 2. Let $(A_{\mathtt{cp}}, B_{\mathtt{cp}}, C_{\mathtt{cp}}, D_{\mathtt{cp}})$, $(A_{\mathtt{cf}}, B_{\mathtt{cf}}, C_{\mathtt{cf}}, D_{\mathtt{cf}})$, (A_p, B_p, C_p) denote the state-space realizations for the controllers $\mathcal{C}_{\mathtt{p}}, \mathcal{C}_{\mathtt{f}}$, and the plant $\mathcal{P}$, respectively. Above we have illustrated the phenomenon where the input signal to the plant cannot be equal to (6.4). It is then given by

$$\begin{aligned} u(t) = u_{id}(t) + u_{tr}(t) \quad \forall t \in [0, \infty), \\ \text{where } u_{tr}(t) = (0 \bigtriangleup^{t_{inter}} (u - u_{id}))(t) \end{aligned} \tag{6.5}$$

is the transient signal induced by the switching at a time t_{inter} [YK07]. In terms of input/state/output representation, the closed-loop behavior with controller $\mathcal{C}_{\mathtt{f}}$ being constantly in the loop is given by

$$\begin{bmatrix} \dot{x}_{\mathtt{cf}} \\ \dot{x}_p \end{bmatrix} = \begin{bmatrix} A_{\mathtt{cf}} & -B_{\mathtt{cf}}C_p \\ B_pC_{\mathtt{cf}} & A_p - B_pD_{\mathtt{cf}}C_p \end{bmatrix} \begin{bmatrix} x_{\mathtt{cf}} \\ x_p \end{bmatrix} + \begin{bmatrix} B_{\mathtt{cf}} \\ B_pD_{\mathtt{cf}} \end{bmatrix} r \tag{6.6}$$

where $x_{\mathtt{cf}}$ and x_p are the state-trajectories of $\mathcal{C}_{\mathtt{f}}$ and $\mathcal{P}$ respectively. The control signal issued by $\mathcal{C}_{cf}$ is given by

$$u_{\mathtt{f},id} = \begin{bmatrix} C_{\mathtt{cf}} & -D_{\mathtt{cf}}C_p \end{bmatrix} \begin{bmatrix} x_{\mathtt{cf}} \\ x_p \end{bmatrix} + D_{\mathtt{cf}}r. \tag{6.7}$$

When controller $\mathcal{C}_{\mathtt{p}}$ is connected in the loop and controller $\mathcal{C}_{\mathtt{f}}$ is not connected, the output signal of $\mathcal{C}_{\mathtt{f}}$, i.e. $u_{\mathtt{f},if}$, is still given by (6.7). However, the evolution of the state-trajectories $x_{\mathtt{cf}}$ and x_p is obtained by the following augmented behavior

$$\begin{bmatrix} \dot{x}_{\mathtt{cf}} \\ \dot{x}_p \\ \dot{x}_{\mathtt{cp}} \end{bmatrix} = \begin{bmatrix} A_{\mathtt{cf}} & -B_{\mathtt{cf}}C_p & 0 \\ 0 & A_p - B_pD_{\mathtt{cp}}C_p & B_pC_{\mathtt{cp}} \\ 0 & -B_{\mathtt{cp}}C_p & A_{\mathtt{cp}} \end{bmatrix} \begin{bmatrix} x_{\mathtt{cf}} \\ x_p \\ x_{\mathtt{cp}} \end{bmatrix} + \begin{bmatrix} B_{\mathtt{cf}} \\ B_pD_{\mathtt{cp}} \\ B_{\mathtt{cp}} \end{bmatrix} r. \tag{6.8}$$

For the convenience of notations, assign $\mathtt{col}(x_{\mathtt{cf}}, x_p) = \chi$, which denotes the joint state of $\mathcal{C}_{\mathtt{f}}$ and $\mathcal{P}$. With this, we can represent the state-trajectory of the controlled behavior with $\mathcal{C}_{\mathtt{f}}$ from the time of origin, i.e. $t_0 = 0$ up to time t together with initial condition $\chi(t_0) = \chi_0$ as

$$\chi_{id}(t) = \chi_{id}(t; t_0, \chi_0, r) = \Phi(t, t_0)\chi_0 + \Theta(t, t_0)r, \tag{6.9}$$

where $\Phi(t, t_0)$ is the state transition matrix of the system (6.6) (see [PW97, Definition 4.5.15]) and $\Theta(t, t_0)$ is an integral operator. When $\mathcal{C}_{\mathtt{f}}$ is not connected in the loop, the $\chi-$trajectory is obtained from the augmented behavior (6.8). This trajectory is the first block component of vector $\eta = \mathtt{col}(\chi, x_{\mathtt{cp}})$ which is the solution of the differential equation (6.8) with initial condition $\eta_0 = \mathtt{col}(\chi_0, x_{\mathtt{cp},0})$. When the switched-mode system, as illustrated in Fig. 6.1, transfers the control authority to the controller $\mathcal{C}_{\mathtt{f}}$ at time t_{inter}, the closed-loop dynamics starts evolving on $[t_{inter}, \infty)$ according to (6.8) with "initial condition $\chi(t_{inter}^-)$ obtained from the left limit of $\eta(t)$ at t_{inter}", that is,

$$\chi(t) = \Phi(t, t_{inter})\chi(t_{inter}^-) + \Theta(t, t_{inter})r; \tag{6.10}$$

while for the controlled behavior with controller $\mathcal{C}_{\mathtt{f}}$ constantly connected in the loop, the state-trajectory on $[t_{inter}, \infty)$ is

$$\chi_{id}(t) = \Phi(t, t_{inter})\chi_{id}(t_{inter}) + \Theta(t, t_{inter})r. \tag{6.11}$$

The switching transient is the *free motion* obtained by taking the difference of trajectories in (6.10) and (6.11), given by

$$\chi_{tr}(t) = \Phi(t, t_{inter}) \cdot (\Delta\chi)_{t_{inter}} \quad \forall t \geq t_{inter}, \tag{6.12}$$

$$\text{where } (\Delta\chi)_{t_{inter}} = \chi_{id}(t_{inter}) - \chi(t_{inter}^-)$$

is the mismatch between the ideal vector and the actual state vector χ at the switching instant. We termed it as the *dynamical inconsistency* due to which transients appear. The transient in the input signal to the plant is then given by

$$u_{tr}(t) = \begin{bmatrix} C_{\mathtt{cf}} & -D_{\mathtt{cf}} C_p \end{bmatrix} \chi_{tr}(t). \tag{6.13}$$

It is clear now that if at the switching instant t_{inter}, the joint state $\chi(t^-_{inter})$ of the plant and offline controller $\mathcal{C}_{\mathtt{f}}$ is equal to the ideal state $\chi_{id}(t_{inter})$, then the input signal (6.5) to the plant will be transient-less after switching even it is experiencing a jump $(\Delta u)_{t_{inter}} = u_{\mathtt{f}}(t^+_{inter}) - u_{\mathtt{p}}(t^-_{inter}) \neq 0$ at the switching instant (the arguments t^+_{inter} and t^-_{inter} stand, respectively, for the right and left limits at t_{inter} of the corresponding signal). The above development illustrates the significance of the joint state-trajectory $\chi(t^-_{inter})$, which contains all the information about the past required to be able to understand what the future may look like. This roughly implies that the joint state has to be initialized with utmost attention, which leads to a so-called state-resetting phenomena [KDY09].

6.3 Guaranteeing the Smooth Interconnection

In both the real-time solutions as presented in Part II of the thesis, the controller reconfiguration process performs, basically, the "when-which" task. It is the latter task, where a controller is "switched" in the closed-loop subject to the occurrence of a fault. In one of the methods, a bank of controllers is designed in which the switching signal allows one of the controllers to be operating in closed-loop. While in another method, a controller is designed, which also has to be switched in closed-loop once synthesized. In this section, we shall deal with issues, from the practical implementation point of view, concerning the *when* task such that the above demonstrated transient phenomenon does not appear during the controller reconfiguration process. This is termed as guaranteeing the smooth interconnection.

To illustrate the case of non-smooth interconnection, let us consider an academic Single Input Single Output (SISO) numerical example. The desired behavior (or the control objective) in this example is to reconfigure the controller such that the $y-$trajectory follows the $r-$trajectory. The behavior of the controller is defined as

$$\mathcal{C}_{\mathtt{j}} = \left\{ s \in \mathbb{S}^{\mathbb{T}} \,\middle|\, \begin{bmatrix} n_{c\mathtt{j}} & -n_{c\mathtt{j}} & -d_{c\mathtt{j}} \end{bmatrix} \begin{bmatrix} r \\ y \\ u \end{bmatrix} \right\} = 0, \text{ where } \mathtt{j} \in \{\mathtt{p}, \mathtt{f}\} \tag{6.14}$$

We use two controllers where the past controller $\mathcal{C}_{\mathtt{p}}$ does not satisfy the desired behavior, and due to a high level supervisory mechanism, it is required to

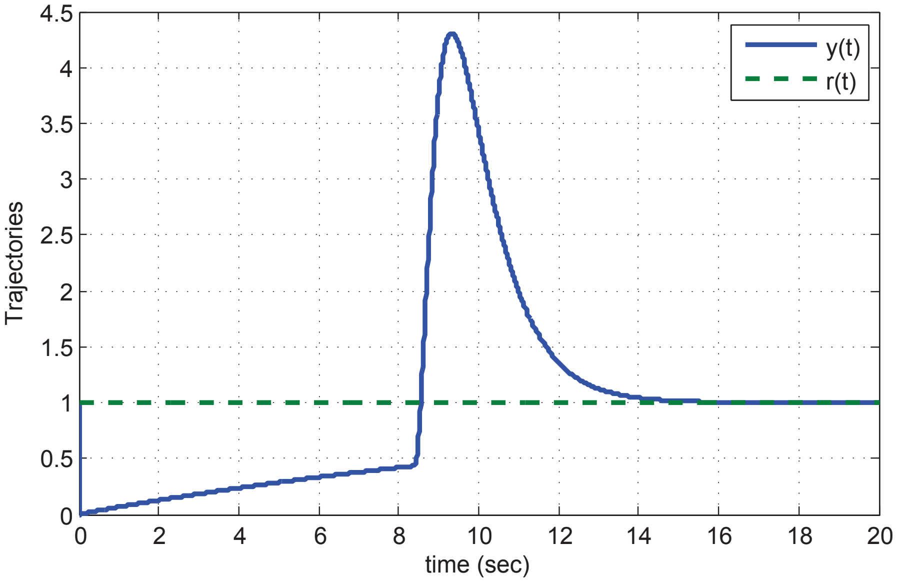


Figure 6.2: Example : Non-smooth Interconnection

switch the future controller $\mathcal{C}_{\mathtt{f}}$ in closed-loop with polynomials : $n_{\mathtt{cp}} = (\xi + 1)(53.33\xi + 160)$, $d_{\mathtt{cp}} = \xi(\xi + 44)$, $n_{\mathtt{cf}} = (\xi + 1)(13.33\xi + 40)$, $d_{\mathtt{cf}} = \xi(\xi + 14)$. Suppose the controller $\mathcal{C}_{\mathtt{f}}$ makes an interconnection with the unknown plant at time $t_{inter} = 8.4sec$. In this case, the concerned trajectories are illustrated in Fig. 6.2, which clearly shows the effects of transients appearing in closed-loop. Thus, it is a non-smooth interconnection.

In order to guarantee the smooth interconnection, we first give the following proposition, which, in fact, is the key result of this chapter.

Proposition 6.1. *Given a vector space of time signals* $(\mathbb{T} \times \mathbb{S})$*, the implementable desired behavior* $\mathcal{D}$*, if there exists a trajectory* $r_{\mathtt{f}} \in (\mathbb{R}^{\mathtt{r}})^{\mathbb{R}}$ *such that* $(r_{\mathtt{f}}, y, u) \in \mathcal{C}_{\mathtt{f}}$*, for any* $(y, u) \in \mathcal{P}$ *up to time* t_{inter}*, then the interconnection between* $\mathcal{C}_{\mathtt{f}}$ *and* $\mathcal{P}$ *is a real-time smooth interconnection.*

Proof. Before time t_{inter}, when $\mathcal{C}_{\mathtt{p}}$ is working in closed-loop, then for any $(r, y, u) \in (\mathbb{R}^{\mathtt{r}+\mathtt{y}+\mathtt{u}})^{\mathbb{R}}$, it satisfies $(r, y, u) \in \mathcal{P} \cap \mathcal{C}_{\mathtt{p}}$. This implies that (y, u) also belongs to $\mathcal{P}$. Since we know the polynomials of $\mathcal{C}_{\mathtt{f}}$, we evaluate the trajectory $r_{\mathtt{f}}$ using the kernel representation of $\mathcal{C}_{\mathtt{p}}$ by

$$n_{\mathtt{cf}} r_{\mathtt{f}} = d_{\mathtt{cf}} u + n_{\mathtt{cf}} y, \qquad \text{where } (y, u) \in \mathcal{P} \tag{6.15}$$

All trajectories are observed for a finite interval of length ι before the controller $\mathcal{C}_{\mathtt{f}}$ is switched in the closed-loop. Now, looking at time t_{inter}, we have $(y, u) \in \mathsf{O}_{\tau}(\mathcal{P})$ and $(r_{\mathtt{f}}, y, u) \in \mathsf{O}_{\tau}(\mathcal{C}_{\mathtt{f}})$. This implies that $(r_{\mathtt{f}}, y, u) \in \mathsf{O}_{\tau}(\mathcal{P} \cap \mathcal{C}_{\mathtt{f}})$ before the controller $\mathcal{C}_{\mathtt{f}}$ is actually interconnected in closed-loop. Indeed,

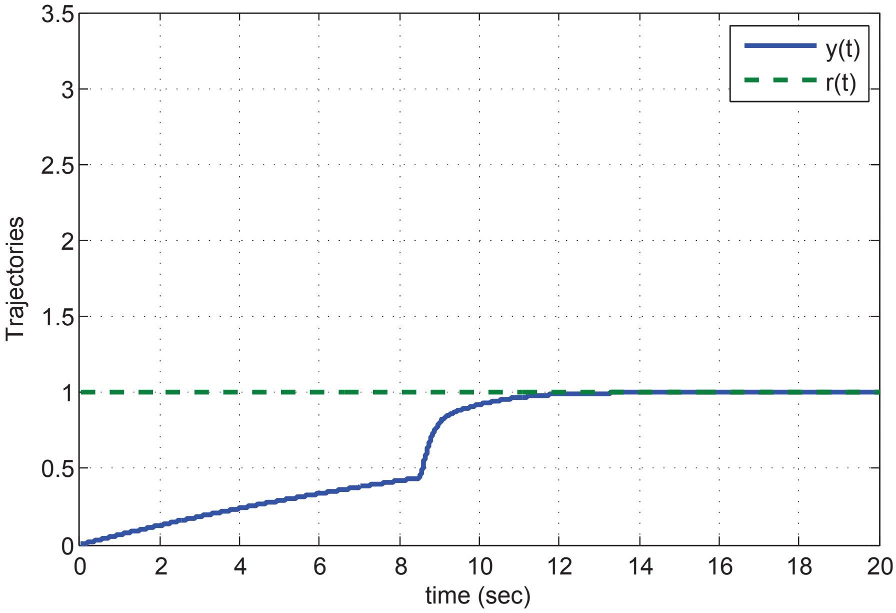


Figure 6.3: Example : Smooth Interconnection

$\mathsf{O}_\tau(\mathcal{P} \cap \mathcal{C}_{\mathtt{f}}) = \mathsf{O}_\tau(\mathcal{K}_{\mathtt{fic}})$. Thus, $(r_{\mathtt{f}}, y, u) \in \mathsf{O}_\tau(\mathcal{D})$ at time t_{inter}. This ensures, followed from the definition 6.1, the interconnection is a real-time smooth interconnection. □

The above phenomenon can be viewed as if the controller $\mathcal{C}_{\mathtt{f}}$ had been previously in the loop. Computation of the $r_{\mathtt{f}}-$trajectory requires "inverting" the controller $\mathcal{C}_{\mathtt{f}}$. This requirement can easily be relieved by designing a biproper (approximate) controller or using another feedback configuration. The sole aim to illustrate the existence of $r_{\mathtt{f}}-$trajectory is that we can compute the ideal steady state-trajectory using the input / state / output realization of the controller as

$$\dot{x}_{\mathtt{f}}^{c'} = (A_{c\mathtt{f}} - B_{c\mathtt{f}} D_{c\mathtt{f}}^{-1} C_{c\mathtt{f}}) x_{\mathtt{f}}^{c'} + B_{c\mathtt{f}} D_{c\mathtt{f}}^{-1} u \tag{6.16}$$

$$r_{\mathtt{f}} = y - D_{c\mathtt{f}}^{-1} C_{c\mathtt{f}} x_{\mathtt{f}}^{c'} + D_{c\mathtt{f}}^{-1} u. \tag{6.17}$$

Thus, if (6.16) is run in parallel with $(u(\iota), y(\iota)) \in \mathcal{K}_{\mathtt{pic}}$ for some $\iota \leq t_{\text{inter}}$, a state trajectory $x_{\mathtt{f}}^{c'}(\iota), \iota \leq t_{\text{inter}}$ can be determined in such a way that

$$\{(r_{\mathtt{f}}(\iota), y(\iota), u(\iota))\}, \iota \leq t_{\text{inter}}\} \in \mathcal{D} \tag{6.18}$$

From the practical viewpoint, it can be considered that it suffices to run (6.16) over a finite time window corresponding to the settling time of the system (6.16), say t_{setl}, i.e. $\iota = t_{\text{setl}}$, to obtain a reasonable estimate for $x_{\mathtt{f}}^{c'}(t_{\text{inter}})$. The insight view of section 6.2 shows that it is the mismatch between the ideal vector and the actual state vector χ at the switching instant. Indeed the vector χ is comprised of the state of the controller $\mathcal{C}_{\mathtt{f}}$ and $\mathcal{P}$. Here, it is only required to reset the state of future controller to the value obtained above. In the non-smooth interconnection, the value of controller's state is obtained by the evolving dynamics of $\mathcal{C}_{\mathtt{p}}$ and $\mathcal{P}$. While, using the above demonstrated procedure, this value is obtained by the evolving dynamics of $\mathcal{C}_{\mathtt{f}}$ and $\mathcal{P}$. This leads to the following real-time algorithm that guarantees the smooth interconnection.

Real-time Algorithm

1. For $t < \tau$, simulate $\forall(y,u) \in \mathcal{P}$

$$\mathcal{C}_{\mathtt{p}} : \begin{cases} \dot{x}^c_{\mathtt{p}} & = A_{c\mathtt{p}} x^c_{\mathtt{p}} + B_{c\mathtt{p}}(r-y) \\ u & = C_{c\mathtt{p}} x^c_{\mathtt{p}} + D_{c\mathtt{p}}(r-y) \end{cases}$$

2. For $\tau - \iota < t \leq \tau$, simulate, in parallel with above, $\forall(y,u) \in \mathcal{P}$

$$\mathcal{C}_{\mathtt{f}} : \begin{cases} \dot{x}^{c'}_{\mathtt{f}} & = (A_{c\mathtt{f}} - B_{c\mathtt{f}} D_{c\mathtt{f}}^{-1} C_{c\mathtt{f}}) x^{c'}_{\mathtt{f}} + B_{c\mathtt{f}} D_{c\mathtt{f}}^{-1} u \\ r_{\mathtt{f}} & = y - D_{c\mathtt{f}}^{-1} C_{c\mathtt{f}} x^{c'}_{\mathtt{f}} + D_{c\mathtt{f}}^{-1} u. \end{cases}$$

3. At $t = t_{\text{inter}} = \tau$, make the interconnection of $\mathcal{P}$ and $\mathcal{C}_{\mathtt{f}}$ with $x^c_{\mathtt{f}}(t_{\text{inter}}) = x^{c'}_{\mathtt{f}}(t_{\text{inter}})$.

Consider again the example introduced before. The settling time of the inverse of the future controller is computed as $\iota = 3.48sec$. From the theory developed above, controller $\mathcal{C}_{\mathtt{f}}$ is run in parallel during the period $[5, 8.48]sec$. This controller, with reinitialized state unlike to the non-smooth interconnection, makes an interconnection at time $t_{inter} = 8.48sec$. The closed-loop signals are illustrated in Fig. 6.3 that clearly shows the smooth interconnection between $\mathcal{C}_{\mathtt{f}}$ and $\mathcal{P}$.

Chapter 7

Simulation Results and Discussion

Contents

7.1 Aircraft Auto landing mechanism 113

7.1.1 Model Description 114

7.1.2 Fault scenarios 116

7.1.3 Constructing a controller bank 116

7.1.4 Projection-based FTC mechanism 117

7.2 Hydraulic Plant 119

7.2.1 Model of the plant 119

7.2.2 Fault scenario 121

7.2.3 Online redesign based FTC mechanism 121

7.3 Wind Turbine 124

7.3.1 Description of Wind Turbine 125

7.3.2 Fault Scenario 127

7.3.3 Fault-Tolerant Control Objectives 128

7.3.4 Projection-based FTC 130

7.3.5 Online-redesign based FTC 134

In this chapter, we shall demonstrate the applications of approaches on various case-studies, namely Aircraft during the landing phase, the two-tanks system, and the wind turbine system, that effectively validate the theory developed in previous chapters.

7.1 Aircraft Auto landing mechanism

Modern aircraft have extensive automation which helps the pilot by performing computations, obtaining data, and completing procedural tasks [OMB+02].

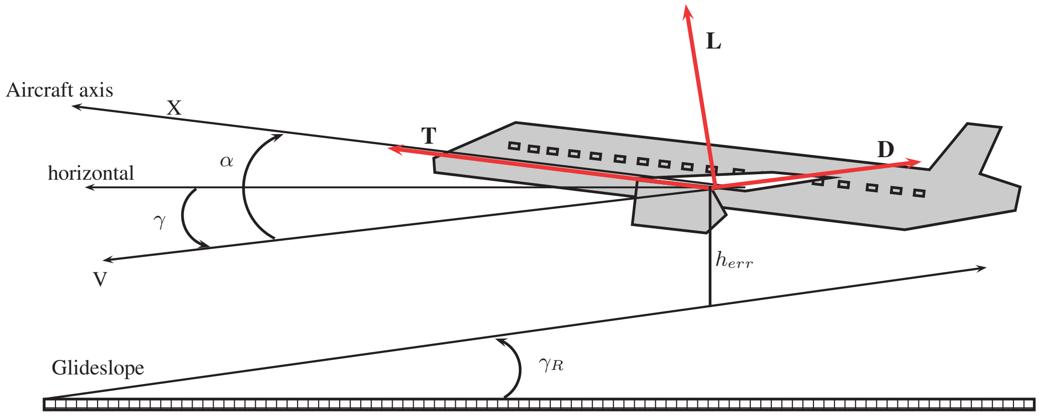


Figure 7.1: Aircraft during the landing phase

The auto-land system of modern aircraft is one of the most safety-critical components, and is subject to stringent certification criteria. In this section, we demonstrate an application of the projection-based approach to construct the fault-tolerant autopilot mechanism for an aircraft during the landing phase. This system is constructed with an objective that the aircraft follows a certain trajectory called the glide-slope. The landing of a civil transport aircraft is divided into three parts, namely approaching a trajectory, flare, and a touch-down and ground run. Fig. 7.1 shows the aircraft in a certain trajectory-approaching phase, which constitutes the final phase of the descent (i.e. the glide-slope). The instrument landing system (ILS) on ground determines the difference between the actual trajectory of the aircraft and the reference trajectory imposed for the descent. Here the purpose is to design a fault tolerant autopilot that fully supports the conduct of the flight in the vertical plane along the glide-slope. Throughout achieving this above task, we have ignored the lateral movement and rolling movements of the aircraft assuming that these aspects are handled by another automated system.

7.1.1 Model Description

For the problem considered (longitudinal flight), the aircraft is seen as a system with three outputs that are measured in real-time: speed V, angle γ of the flight path and the distance from the center of mass of the aircraft relative to the glide-slope h_{err}. The control inputs of the system are the aircraft thrust T and the elevator command δ, where the elevator is a movable aerodynamic surface located in the empennage that controls the pitch of the aircraft θ. The relationship between flight path's angle, angle of attack, and pitch of the aircraft is given by $\theta = \alpha + \gamma$. We assume that there are no dynamics between

Parameter	Description	Value	Unit
m	Mass of the Aircraft	190000	kg
g	Gravitational Constant	9.81	m/s^2
γ_R	flightpath angle	0.052359	rad
C_{L_0}	Coefficient of lift for 0 angle of attack	0.8212	–
C_{D_0}	Coefficient of drag for 0 angle of attack	0.025455	–
C_{L_α}	Coefficient of lift due to angle of attack	5.105	$1/rad$
K	Coefficient of drag due to the lift	0.04831	–
ρ	Air density	1.225	kg/m^3
S	Surface area	427.8	m^3

Table 7.1: Aerodynamics parameters

the elevator command and the angle of attack α of the wing. Thus, α can be seen as equivalent to δ and consequently, for the sake of simplicity, we treat α as a control input. The thrust controls the speed V of the aircraft. The objective is that the aircraft follows along the glide-slope, making a desired flight path angle at 3 degrees clockwise (i.e. $\gamma_r = -3\,deg.$). Thus, it makes h_{err} null. The nonlinear model of the longitudinal dynamics of a large jet aircraft is given as:

$$m\frac{dV}{dt} = -D(\alpha, V) + T\text{cos}\alpha - mg\text{sin}\gamma \tag{7.1}$$

$$mV\frac{d\gamma}{dt} = L(\alpha, V) + T\text{sin}\alpha - mg\text{cos}\gamma \tag{7.2}$$

$$\frac{dh_{err}}{dt} = V(\text{sin}\gamma + \text{cos}\gamma\text{tan}\gamma_R) \tag{7.3}$$

It is the aircraft's lift $L(\alpha, V)$ and the drag $D(\alpha, V)$ that induce non-linearity in the above equations, given as

$$L(\alpha, V) = C_L(\alpha) \cdot \frac{1}{2}\rho V^2 S, \quad \text{with } C_L(\alpha) = C_{L_0} + C_{L_\alpha} \cdot \alpha \tag{7.4}$$

$$D(\alpha, V) = C_D(\alpha) \cdot \frac{1}{2}\rho V^2 S, \quad \text{with } C_D(\alpha) = C_{D_0} + KC_L^2(\alpha). \tag{7.5}$$

The other parameters appearing in above equations are described in Table 7.1. As explained above, the control inputs of the aircraft are considered as $u = \texttt{col}(\alpha, T)$. The interest lies in the state of equilibrium when the aircraft is on the flightpath angle, i.e. $\gamma = -0.05236rad$, $h_{err} = 0$, and with velocity setpoint $V = V_c = 81.8m/s$ [Yam05].

7.1.2 Fault scenarios

The above description of the model is given just to show how the dynamics of the aircraft evolves [JYS12c]. However, we do not use any *a priori* information of the model in real-time to demonstrate the fault-tolerant control mechanism. For illustrating the FTC mechanism, we consider the complete loss of one of the control surfaces, i.e. a fault in the elevator. Two modes of the aircraft system are considered: the nominal mode (no fault) and a complete stuck in the angle of attack (faulty mode). We use the linearized model around the trim points, $\alpha = 2.686\, deg.$ and $T = 4.23 \times 10^4 N$, given as

$$\dot{x} = Ax + Bu, \quad z = Cx, \tag{7.6}$$

where $x = \begin{bmatrix} V & \gamma & h_{err} \end{bmatrix}^T, u = \begin{bmatrix} \alpha & T \end{bmatrix}^T,$

$$A = \begin{bmatrix} -0.0180 & -9.7966 & 0 \\ 0.0029 & -0.0063 & 0 \\ 0 & 81.9123 & 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In the fault-free mode, $B-$matrix is given by

$$B = B_h = \begin{bmatrix} -4.8374 & 5.2574 \times 10^{-6} \\ 0.5786 & 3.0149 \times 10^{-9} \\ 0 & 0 \end{bmatrix},$$

while in the faulty mode, it is given by

$$B = B_f = \begin{bmatrix} 0 & 5.2574 \times 10^{-6} \\ 0 & 3.0149 \times 10^{-9} \\ 0 & 0 \end{bmatrix}.$$

7.1.3 Constructing a controller bank

As mentioned before, the control objective is to maintain the h_{err} equal to zero. That is, the references to be tracked are V and h_{err}. Thus, the output is now given as $y = C_o x$, where $C_o = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. We design two corresponding controllers for the two different modes based on (7.6). The control law is given as

$$u(t) = -K.z(t) + K_p.e(t) + K_i. \int_0^t e(\vartheta) d\vartheta \tag{7.7}$$

with $e(t) = w(t) - y(t)$, where w is the reference trajectory. The matrix gain corresponding to the measurement (or state) feedback is designed using the pole-placement technique. The poles for both modes are placed at

$(-2.8782, -2.3026 \pm 1.7269i)$ that makes the system internally stable. The matrix gains corresponding to "Proportional+Integral" (PI) structure on error assist to follow the desired trajectory. The gains for healthy and faulty mode are then chosen as

$$K_h = \begin{bmatrix} 2.328 \times 10^{-3} & 7.919 & 0.174 \\ 5.461 \times 10^{5} & 5.409 \times 10^{-3} & 1.598 \times 10^{5} \end{bmatrix}$$

$$K_{ph} = \begin{bmatrix} 2.2 \times 10^{-3} & 5.21 \times 10^{-2} \\ 9.2755 \times 10^{5} & 5.9528 \times 10^{6} \end{bmatrix},$$

$$K_{ih} = \begin{bmatrix} 2.2 \times 10^{-2} & 1.563 \times 10^{-1} \\ 9.2755 \times 10^{6} & 1.7858 \times 10^{7} \end{bmatrix},$$

$$K_f = \begin{bmatrix} 0 & 0 & 0 \\ 7.926 \times 10^{5} & 1.091 \times 10^{9} & 1.886 \times 10^{7} \end{bmatrix}$$

$$K_{pf} = \begin{bmatrix} 1 & 131.63 \\ -1.9018 \times 10^{4} & 9.5928 \times 10^{4} \end{bmatrix},$$

$$K_{if} = \begin{bmatrix} 0.6339 & 2.7853 \times 10^{3} \\ -1.2057 \times 10^{4} & 2.0298 \times 10^{6} \end{bmatrix}$$

The subscript h and f represent the gains for healthy and faulty mode respectively.

Note that these controllers are pre-designed, i.e. before the system is put in real-time operation. In addition, designing of the controllers is described for the ease of the reader. Since the main aim of the provided controller's bank is that at least one of the controllers implements the desired behavior, however these controllers can also be designed in other ways. The underlying aim is not to stress on the designing of controllers, instead how the supervisory mechanism switches the fault correcting controller in one-shot.

7.1.4 Projection-based FTC mechanism

The time interval during which the measurements are taken is τ and the *control performance* is evaluated at $t_n = n\tau$, the instants of possibly switching. The performance threshold λ distinguishes various modes of the system. These parameters are set to $\lambda = 5, \tau = 5s$, where the performance functional is chosen as

$$J = \int_{n\tau}^{(n+1)\tau} \|(\hat{r} - y)(\varsigma)\|_2^2 d\varsigma, \tag{7.8}$$

The system-level Simulink architecture of the FTC scheme is illustrated in Fig. 7.2.

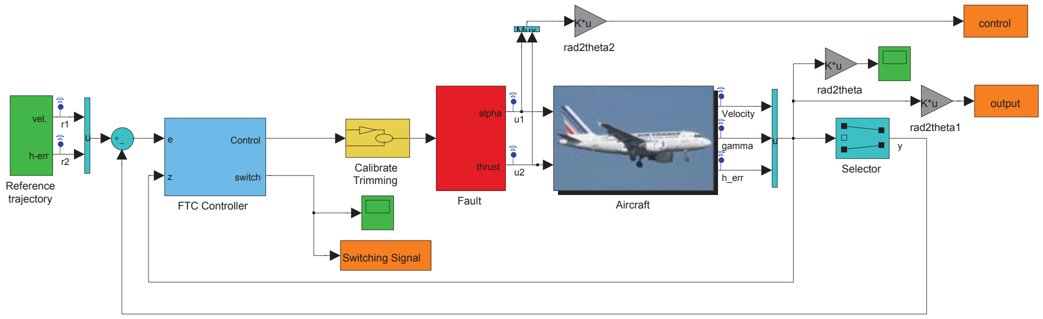


Figure 7.2: System-level Simulink Architecture of Projection-based Fault-Tolerant Aircraft Control

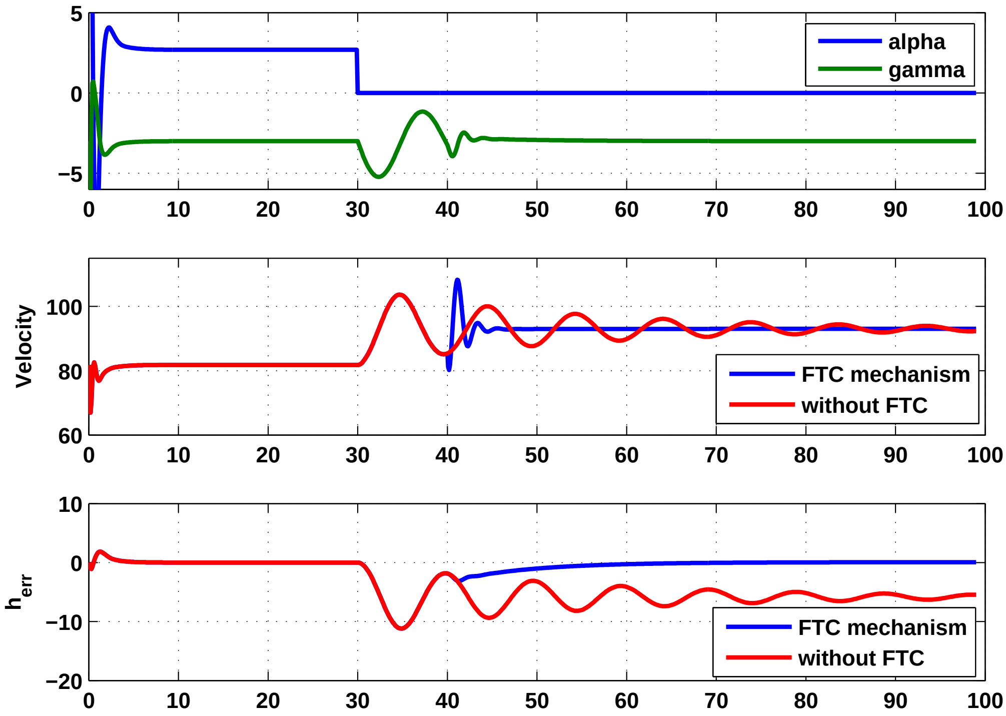


Figure 7.3: Closed-loop signals aircraft autopilot landing system

An experiment is run with a completely stuck in the angle of attack appearing at time $30s$. The closed-loop signals of Fig. 7.3 show that the real-time FTC system has successfully reacted at time $40s$ by switching to controller 2 (faulty mode controller). After an acceptable transient, the control objective is recovered as seen from the distance from the center of mass of the aircraft relative to the glide-scope approaching to zero. Note that since the FTC scheme is based on the control performance, when the active controller is invalidated by the operating plant data, the supervisor puts into feedback the best controller from the potential controllers set, that is the controller yielding optimal closed-loop performance in *real-time*.

7.2 Hydraulic Plant

The plant is composed of two interconnected tanks, two pumps that provide the flow rates Q_1 and Q_2, two level sensors L_1, L_2, five flow-rate sensors for the measurements of Q_1, Q_2, Q_{F_1}, Q_{F_2} and Q_{12} and three valves. This two-tank system is illustrated in Fig. 7.4 [YS08]. The control inputs to the plant are the voltages V_{pump_1}, V_{pump_2} applied to the pumps and the voltage V_{12} for the throttling of the interconnection valve. The flows Q_{F_1} and Q_{F_2} are mixed through the valves located at the output of the tanks. The main objective of the system is to keep the sum y_1 and the ratio y_2 of the output flow rates to desired set-points r_1 and r_2, where

$$y_1 = Q_{F_1} + Q_{F_2}, \tag{7.9}$$

$$y_2 = \frac{Q_{F_1}}{Q_{F_2}}. \tag{7.10}$$

The technique developed in Chapter 5 has been applied to the plant.

7.2.1 Model of the plant

The system has two state variables which are the liquid levels L_1 and L_2 of the tanks. The equations describing the evolution of the states are

$$\begin{aligned} S_1\dot{Q}_1 &= Q_1 - Q_{12} - Q_{F_1} \\ S_2\dot{Q}_2 &= Q_2 - Q_{12} - Q_{F_2} \end{aligned} \tag{7.11}$$

The variables in the right-hand side of these state equations are given by the known nonlinear maps

$$Q_1 = \pi_1(V_{pump_1}), \qquad Q_2 = \pi_2(V_{pump_2}) \tag{7.12}$$

$$Q_{F_1} = R_1\sqrt{L_1}, \qquad Q_{F_2} = R_2\sqrt{L_2} \tag{7.13}$$

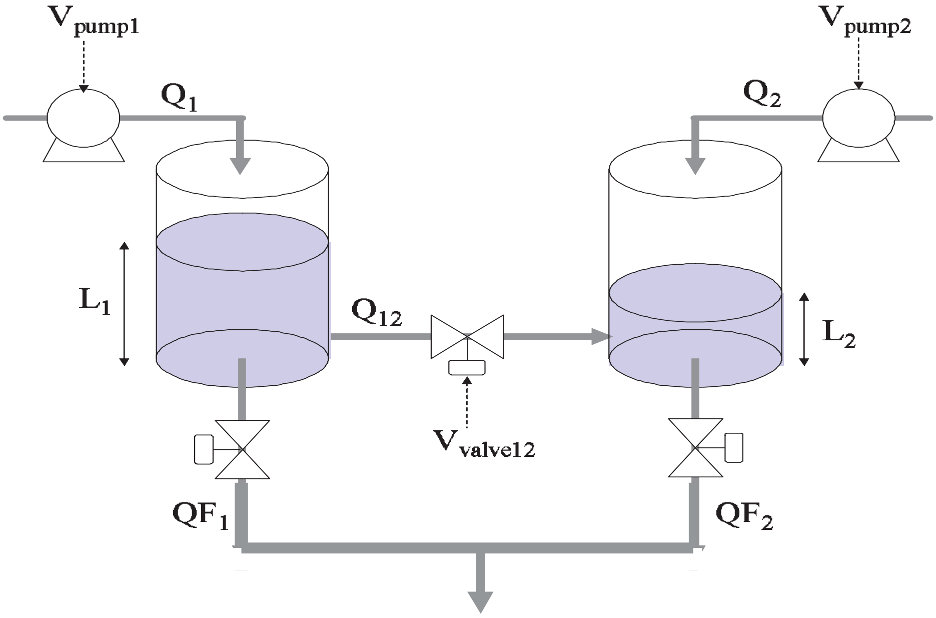


Figure 7.4: The two-tanks plant

and

$$Q_{12} = R_{12}(V_{12}) \cdot \sqrt{|L_1 - L_2|} \cdot \texttt{sign}(|L_1 - L_2|) \tag{7.14}$$

where π_1, π_2 and R_{12} are nonlinear transformations which describe the characteristics of the pumps and the interconnection valve as a function of the corresponding input voltages. The parameters R_1, R_2 are the throttling of valves 1 and 2, and S_1, S_2 are the section of tank 1 and tank 2 respectively. With the explicit expression of Q_{F_1} and Q_{F_2}, the controlled outputs of the system are given by

$$y_1 = R_1\sqrt{L_1} + R_2\sqrt{L_2}, \tag{7.15}$$

$$y_2 = \frac{R_1\sqrt{L_1}}{R_1\sqrt{L_2}}. \tag{7.16}$$

Since these controlled outputs are required to follow the desired set-points r_1 and r_2, these set-points can be rewritten as desired set-points L_1^0, L_2^0 for the measured levels L_1, L_2 with

$$L_1^0 = \left(\frac{r_1 r_2}{R_1(1+r_2)}\right)^2, \quad L_2^0 = \left(\frac{r_1}{R_2(1+r_2)}\right)^2 \tag{7.17}$$

7.2.2 Fault scenario

The main hardware devices used for controlling and sensing the pilot plant, i.e. the two pumps, the interconnection valve and the two level sensors, can be affected by a fault. The nominal fault-free (or healthy) system operating point is fixed at $(L_1^0, L_2^0) = (0.4, 0.5)$ meters, $V_{12} = 2$ Volts. The linearization of the nonlinear equations (7.11) at the nominal operating point yields

$$\dot{x} = Ax + Bv, \quad y = Cx \tag{7.18}$$

with $y = x = \begin{bmatrix} l_1 & l_2 \end{bmatrix}^T$ and $v = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T$

$$A = \begin{bmatrix} -0.0037 & -0.0017 \\ -0.0018 & -0.0035 \end{bmatrix},$$
$$B = B_h = \begin{bmatrix} 64.9351 & 0 & -0.0001 \\ 0 & 65.7895 & 0.0002 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7.19}$$

where $l_i = L_i - L_i^0, u_i = V_{pump_i} - V_{pump_i}^0$ for $i = 1, 2$ and $u_3 = V_{12} - V_{12}^0$; the variables with superscript 0 denotes values at the nominal point, where L_1, L_2 are the two level sensors and the control inputs to the plant are the voltages V_{pump1}, V_{pump2} applied to the pumps and the voltage V_{12} for the throttling of the interconnection valve. The interconnection valve will be maintained open at the constant nominal value $V_{12}^0 = 2$ Volts at anytime.

Different types of faults, such as bias, drift, power loss and stuck can be realized on these devices. For the purpose of illustrating the FTC technique of the previous section, we consider pump 2 subject to a *power loss* fault with an effectiveness factor of 0.5. Whenever this fault appears in the system, it changes the above $B-$matrix to

$$B = B_f = \begin{bmatrix} 64.9351 & 0 & -0.0001 \\ 0 & 32.8947 & 0.0002 \end{bmatrix}. \tag{7.20}$$

7.2.3 Online redesign based FTC mechanism

The control objective is to maintain the levels of the two tanks at their set-point values at anytime [JYS12g]. With the above consideration, the plant can be viewed as a multi-variable system with two controlled inputs, and two sensed outputs. Therefore, to restrict the behavior of the plant to a desired set, it is required to make an interconnection between the plant and the controller. At the outset, we assign the structure of the controller as a multi-variable "Proportional+Integral" (PI) control that processes sensor outputs into actuator inputs. The controller $\mathcal{C}$ is then given as

$$C_e(\xi)u = C_u(\xi)e \tag{7.21}$$

with $e = r - y$,

$$C_e(\xi) = \xi.I_2 \text{ and } C_u(\xi) = \begin{bmatrix} k_{p_{1,1}}\xi + k_{i_{1,1}} & k_{p_{1,2}}\xi + k_{i_{1,2}} \\ k_{p_{2,1}}\xi + k_{i_{2,1}} & k_{p_{2,2}}\xi + k_{i_{2,2}} \end{bmatrix}.$$

Further, a way the PI controller is formulated in equation (7.21), it can be viewed as equivalent to the way it is often written in the classical sense

$$u(t) = K_p(r(t) - y(t)) - K_i \int_0^t (r - y)(\delta)d\delta \tag{7.22}$$

where $K_p = \begin{bmatrix} k_{p_{1,1}} & k_{p_{1,2}} \\ k_{p_{2,1}} & k_{p_{2,2}} \end{bmatrix}$ and $K_i = \begin{bmatrix} k_{i_{1,1}} & k_{i_{1,2}} \\ k_{i_{2,1}} & k_{i_{2,2}} \end{bmatrix}$ are the controller gain matrices whose coefficients are to be computed. The implementable desired behavior $\mathcal{D}$ that achieves the control objective is given as

$$\begin{bmatrix} D_r(\xi) & -D_y(\xi) \end{bmatrix} w = 0 \tag{7.23}$$

where $D_y(\xi) = d_y(\xi)I_2, D_r(\xi) = d_r(\xi)I_2$ with

$$\begin{aligned} d_y(\xi) &= \xi^5 + 1.994\xi^4 + 1.609\xi^3 + 0.6126\xi^2 + 0.09437\xi, \\ d_r(\xi) &= 0.0153\xi^4 + 0.01995\xi^3 + 0.009416\xi^2 + 0.001453\xi \end{aligned}$$

and $w = (r^T, y^T)^T$. The reference levels are given by $r = \begin{bmatrix} 0.4 & 0.5 \end{bmatrix}$, and the interval of length τ is chosen as $\tau = 15s$.

A measurement set $\mathcal{M}_\tau$ is formed during every interval of length τ. Using the tools borrowed from the system identification community as described in [GW08], the main controller synthesis equation is computed using the online measurements which yields the new parameters of the controller such that when it makes an interconnection with the plant, the closed-loop achieves the desired specifications. The initial nominal controller working in the closed-loop that achieves the desired behavior is computed as

$$K_p^h = \begin{bmatrix} 0.0435 & -0.0478 \\ 0.0159 & -0.0142 \end{bmatrix}, K_i^h = \begin{bmatrix} -0.0069 & 0.0105 \\ 0.0023 & -0.0062 \end{bmatrix} \tag{7.24}$$

The superscripts h and f on the controller gain matrices refer to the nominal mode and the faulty mode respectively. A "power loss" fault occurs in the system at time $t = 60s$. According to the proposition 5.1, an online controller is designed based on the measurement set $\mathcal{M}_\tau$ observed during the time $t \in [60, 75]sec$. The faulty mode controller is then computed online and the evaluated gain matrices are

$$K_p^f = \begin{bmatrix} 0.3514 & -0.1551 \\ 0.0361 & -0.0164 \end{bmatrix}, K_i^f = \begin{bmatrix} -0.3068 & 0.3340 \\ 0.0032 & -0.0037 \end{bmatrix} \tag{7.25}$$

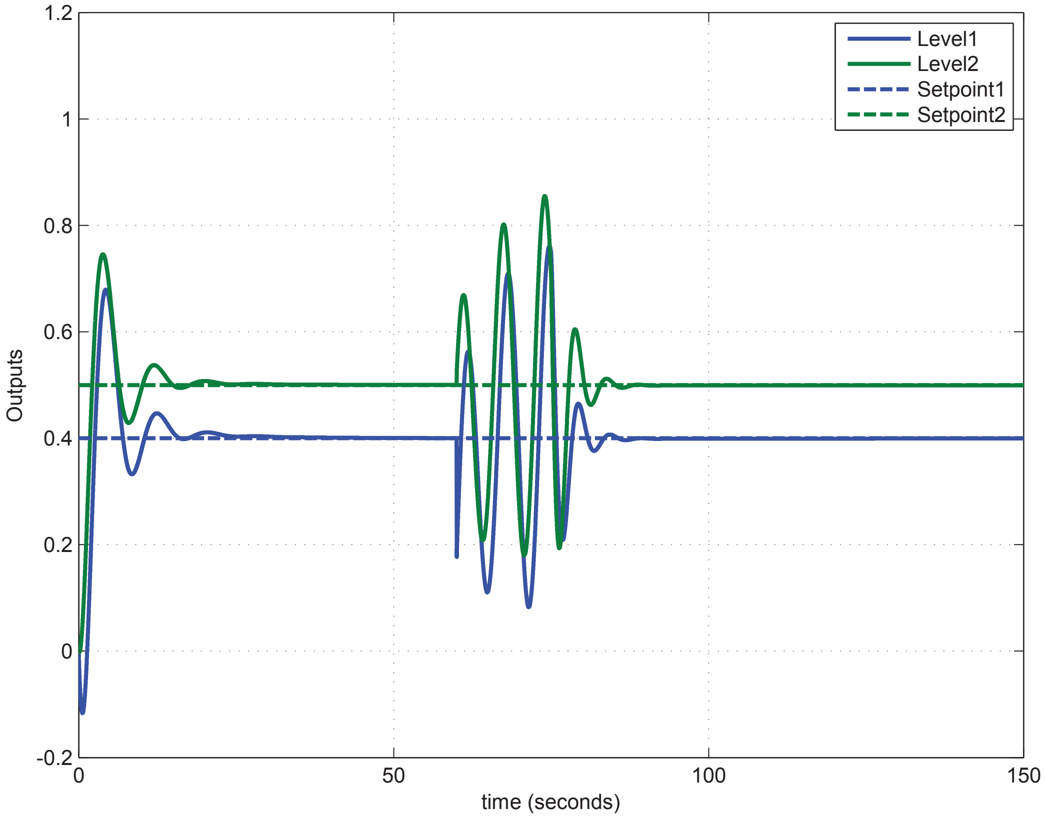


Figure 7.5: Closed loop signals of two-tank FTC system

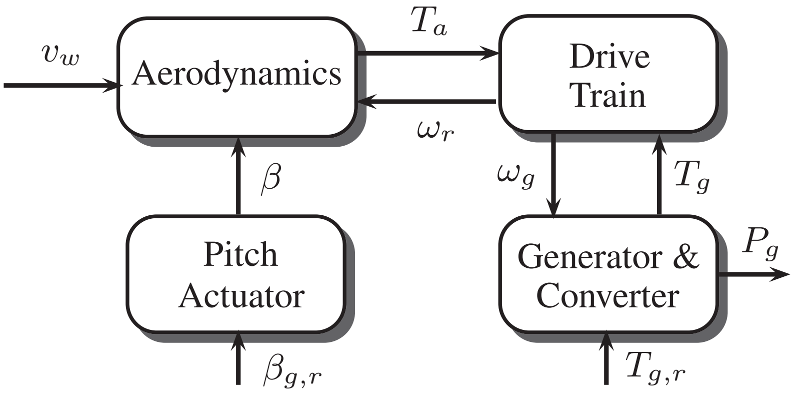


Figure 7.6: System Overview of the Wind-Turbine model

This online synthesized controller then makes an interconnection with the plant at time $t = 75sec$. Fig. 7.5 illustrates the outputs of the closed-loop system. Before the occurrence of a fault, the closed-loop system achieves the control objective. However, due to an occurring fault after, the closed-loop behavior does not satisfy the desired behavior. Since, the desired behavior is implementable, the faulty closed-loop system with the synthesized controller again achieves the control objective. Thus, we conclude that the demonstrated control system is a real-time fault-tolerant control system.

7.3 Wind Turbine

Renewable sources of energy are now considered to be on the priority level of energy policies in many countries. This is, as a matter of fact, easy to say after seeing an enormous increase in world population, and the growing demands of polluting sources of energy. Particularly, for generating electricity, there has been seen a large call for the use of fossil fuels, which together contributing to the pollution has also made electricity more expensive. Among the non-polluting sources of energy, wind power is recognized as one of the valuable resources, even two centuries before the internal-combustion engines and modern power plants were developed. In view of this, today many Wind Turbines (WT) are installed offshore, which contribute to a larger part of the world's power production. At present, due to its bigger size, the installation of such devices is very expensive. Consequently, to maintain the reliability of these turbines becomes an important issue. The block structure of the wind turbine is illustrated in Fig. 7.6, where the driving force is the wind speed. It is a variable-speed, 3-blade Horizontal Axis Wind Turbine (HAWT) with a full converter. The basic functionality includes a two-step energy conversion.

The first step is to convert the wind energy into the mechanical energy, where the wind turns the turbine blades around. In the second step, the mechanical energy is converted to electrical energy by a generator fully coupled to a converter.

7.3.1 Description of Wind Turbine

The system is composed of various sub-systems, namely Blade & Pitch System (BPS), Drive Train (DT), and Generator & Converter (GC). The BPS model is the combination of an aerodynamic model, and a pitch model. The latter model is treated as the actuator within the system and will be discussed later together with other actuators. The aerodynamic properties of the wind turbine are affected by the pitch angles of the blades, the speed of the rotor, and the wind speed. This aerodynamic torque is applied to the rotor T_a and is expressed as

$$T_a(t) = \frac{\rho \pi R^3 C_q(\lambda(t), \beta(t)) v_w(t)^2}{2}, \tag{7.26}$$

where $\lambda(t) = \frac{\omega_r(t)R}{v_w(t)}$. See Table 7.2 for the definitions of the parameters. On the basis of (7.26), an aerodynamic torque is transferred from the rotor to the generator through the drive train. The DT model includes a low-speed shaft (LSS) and a high-speed shaft (HSS) that are linked together by a gearbox modeled as a gear ratio. This is responsible for gearing up the rotor rotational speed to a higher speed required by the generator. The model of the wind turbine drive train can be expressed by the following differential equations:

$$\begin{aligned}
J_r \frac{d}{dt}\omega_r(t) &= T_a(t) - K_{dt}\theta_\triangle(t) - (B_{dt} + B_r)\omega_r(t) + \frac{B_{dt}}{N_g}\omega_g(t), \\
J_g \frac{d}{dt}\omega_g(t) &= \frac{K_{dt}}{N_g}\theta_\triangle(t) + \frac{B_{dt}}{N_g}\omega_g(t) - T_g(t), \\
\frac{d}{dt}\theta_\triangle(t) &= \omega_r(t) - \frac{1}{N_g}\omega_g(t).
\end{aligned}$$

The above briefly described model together with another model, termed as the Tower model are integrated within the FAST (Fatigue, Aerodynamics, Structures, and Turbulence) aeroelastic wind turbine simulator designed by the NREL [JB09]. The model of the tower is not described above. However, it is also integrated within the FAST code. Basically, the movement of the tower is modeled using the spring-damper phenomenon. This movement acts as a disturbance to the wind speed. The sole aim of describing the aerodynamic model is to show the basic source of non-linearity in the WT. Otherwise, all other models can be expressed as a linear time-invariant systems similar to the DT model.

Table 7.2: Aerodynamics parameters

Parameter	**Description**	**Unit**
ρ	Air density	kg/m^3
R	Radius of the rotor	m
C_q	Torque co-efficient	–
λ	tip-speed ratio	–
v_w	wind speed	m/s
β	blade pitch angle	deg
β_r	blade pitch controller output	
ω_r	rotor speed	$radian/minute$
ω_g	generator speed	
P_r	Rated Power	$MegaWatt$
P_g	Generated Power	
J_r	Rotor inertia	kgm^2
J_g	Generator inertia	
B_r	rotor external damping	$Nm/(rad/s)$
B_g	generator external damping	
B_{dt}	torsion damping co-efficient	
T_g	Generator Torque	Nm
$T_{g,r}$	Generator Torque controller output	
T_a	Aerodynamic Torque	
N_g	gearbox ratio	–
K_{dt}	torsion stiffness	Nm/rad
$\theta_{\triangle}$	torsion angle	rad
η_g	efficiency of generator	–
α_{gc}	GC model parameter	–
ω_n	natural frequency	$radian/second$
ζ	damping factor	–

7.3.1.1 Actuator Model

In this benchmark WT model, three actuators for the pitch, generator, and yaw systems are modeled and implemented externally, i.e. apart from the embedded FAST code. Here, we mainly concentrate on the other two actuators but the yaw system. Actually, the yaw actuator model and the associated yaw controller conceived as an overall yaw mechanism is used to orient the wind-turbine upright to the wind direction. The FAST implementing nonlinear WT model requires a yaw angular velocity and yaw angular position as one of the inputs. We assume throughout that a yawing system exists, which keep the wind direction perpendicular to the rotor plane.

The hydraulic pitch system consists of three identical pitch actuators, which is modeled as a linear differential equation with time-dependent variables, pitch angle $\beta(t)$ and its reference $\beta_r(t)$. In principle, it is a piston servo-system which can be expressed as a second-order differential system [OJ12]:

$$\frac{d^2}{dt^2}\beta(t) + 2\zeta\omega_n \frac{d}{dt}\beta(t) + \omega_n^2 \beta(t) - \omega_n^2 \beta_r(t) = 0. \tag{7.28}$$

The dynamics in (7.28) are associated with each of the three pitch actuators. Some constraints are imposed on the pitch actuators. In particular, the pitch angle is restricted to vary within the interval $[-2\deg, 90\deg]$, and pitch rate $[-8\deg/s, 8\deg/s]$.

In the GC system, the converter loads the generator producing the electric power with a certain torque. The dynamics of the converter can be approximated by a first-order differential system [OJ12], which is given by

$$\frac{d}{dt}T_g(t) + \alpha_{gc}T_g(t) - \alpha_{gc}T_{g,r}(t) = 0, \tag{7.29}$$

with $\alpha_{gc} = 50$. The output from the converter has a saturation limit, and a slew rate limit which is embedded within the benchmark model. The power produced by the generator is given by

$$P_g(t) = \eta_g \omega_g(t) T_g(t). \tag{7.30}$$

7.3.2 Fault Scenario

Various types of faults are addressed in the literature [OJ12]. However, we consider a fault that causes an abrupt power drop in the hydraulic pressure. This power drop fault affects the dynamics of the pitch system by changing the parameters, ζ and ω_n from their nominal or healthy-mode values ζ_n and $\omega_{n,n}$ to their values in faulty-mode ζ_f and $\omega_{n,f}$. The faulty dynamics of the pitch system can be described by the following second-order differential system

$$\frac{d^2}{dt^2}\beta(t) + 2\zeta(\Theta_f(t))\omega_n(\Theta_f(t))\frac{d}{dt}\beta(t) + \omega_n^2(\Theta_f(t))\beta(t) \\ - \omega_n^2(\Theta_f(t))\beta_r(t) = 0, \quad (7.31)$$

where

$$\begin{aligned} \omega_n^2(\Theta_f(t)) &= (1 - \Theta_f(t))\omega_{n,n}^2 + \Theta_f(t)\omega_{n,f}^2 \\ 2\zeta(\Theta_f(t))\omega_n(\Theta_f(t)) &= 2(1 - \Theta_f(t))\zeta_n\omega_{n,n} + 2\Theta_f(t)\zeta_f\omega_{n,f} \end{aligned}$$

with $\Theta_f(t) \in [0, 1]$ representing the various operating modes of the WT.

7.3.3 Fault-Tolerant Control Objectives

The wind turbine principally operates inside four regions or control-zones depending on the speed of wind $v_w(t)$. The boundary limits of these regions are marked by cut-in wind speed $v_{w,cut-in}$, rated wind speed $v_{w,rated}$, and cut-out wind speed $v_{w,cut-out}$ as shown in Fig. 7.7. The simulator is the NREL's 5 MW "baseline" turbine and for this turbine, the values of $v_{w,cut-in}$, $v_{w,rated}$, and $v_{w,cut-out}$ are given as $3, 11.4$, and 25 units respectively. In Fig. 7.7, Zone-I is a startup of the turbine; Zone-II is called the partial load region, where the control objective is to maximize the power generated by the turbine; Zone-III is called the full load region, where the control objective is to keep the generator power around the rated generator power; Zone-IV is called the high wind speed region, where the wind turbine is allowed to shut down to avoid stresses and fatigue damages.

Throughout demonstrating the approaches, we consider the wind speed at a mean value of above 11.4 units. Therefore, our main interest lies in Zone-3. The control objective is to design controllers such that the generated power $P_g(t)$ can track the rated power P_{rated} around its mean value of 5 units. Nevertheless, under a fault occurrence, satisfying the above requirements can no longer be guaranteed. Consequently, the fault-tolerant control objective is to design a real-time controller reconfiguration mechanism such that the aforementioned requirements can be fulfilled at anytime. Moreover, suppressing large transients during accommodating an occurring fault is another requirement from the practical implementation point of view. The basic architecture of the proposed Fault-Tolerant Wind Turbine Control is illustrated in Fig. 7.8 [JYS13e].

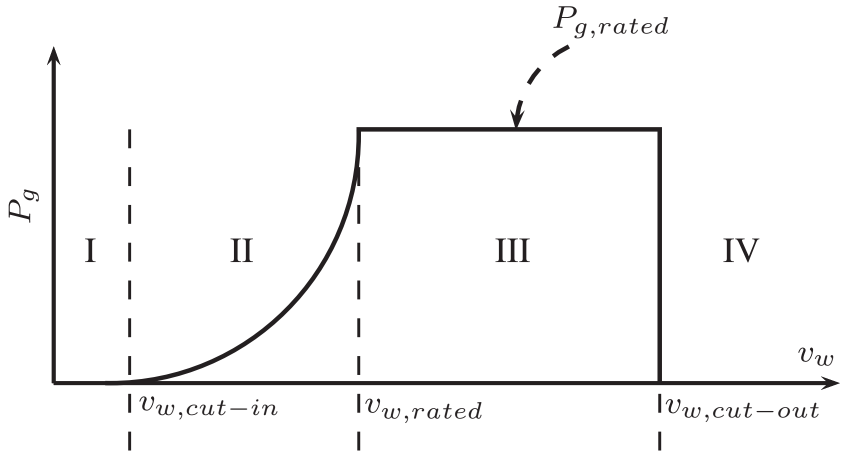


Figure 7.7: Illustration of the reference power curve for the wind turbine depending on the wind speed

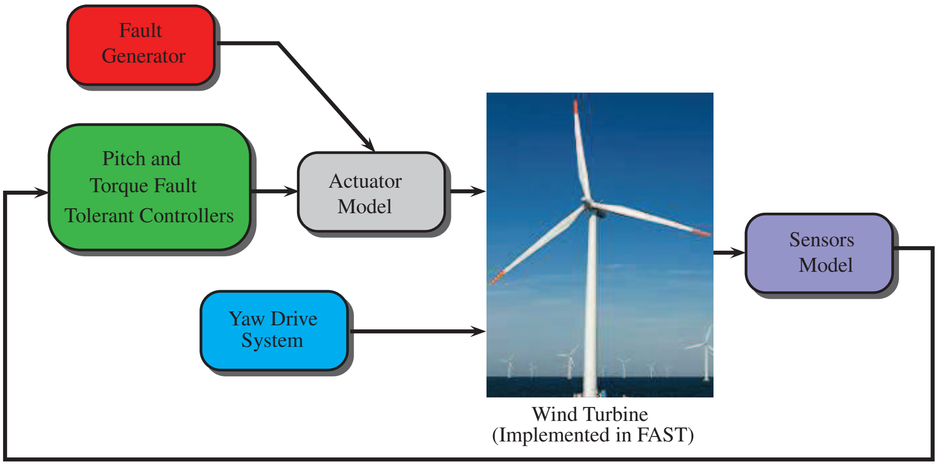

Fault Generator
Pitch and Torque Fault Tolerant Controllers
Actuator Model
Yaw Drive System
Wind Turbine (Implemented in FAST)
Sensors Model

Figure 7.8: Block diagram showing major elements of the Simulink-based Wind Turbine FTC Systems

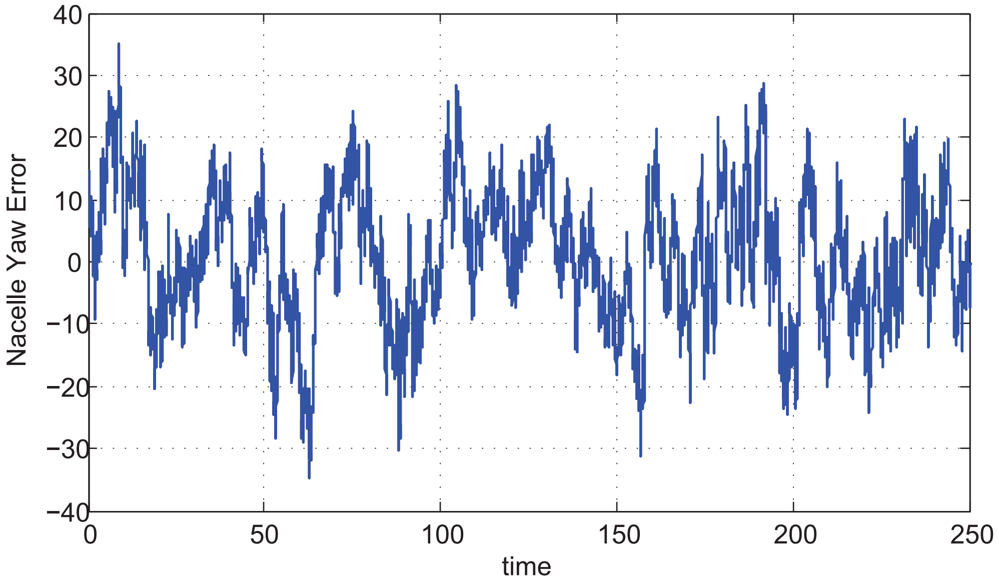


Figure 7.9: Nacelle Yaw Error

7.3.4 Projection-based FTC

The two control inputs to the wind turbine are the generator torque $T_{g,r}$, and the blade pitch angle β_r. Since, we are working in Zone-3, the control objective is to track the generator power at its rated value of 5 units. In addition, the other requirements that are described in the subsection 7.3.3 must also be satisfied. As suggested in [OJ12], within the full load region the main control scheme is developed in the torque control and the pitch controls from the industrial standpoint. FAST also requires a yaw angular velocity and a yaw angular position as inputs. Geometrically, the speed of the wind is represented by the vector $\vec{v_w}(t) = v_{w_x}(t)\,\vec{i} + v_{w_y}(t)\,\vec{j} + v_{w_z}(t)\,\vec{k}$. The $v_{w_x}(t)$ component flows perpendicular to the rotor plane. Here, all other components are assumed zero, and hence no yaw system is considered. However, if other components are also considered then the job of the yaw controller is to ensure the zero mean value of nacelle yaw error. To demonstrate that only $v_{w_x}(t)$ component is considered, the nacelle yaw error is plotted in Fig. 7.9. With this consideration, the system can be viewed as a multi-variable two-input two-output system.

In the AD phase, the parameter space $\Theta(t) \in [0, 1]$ was gridded with a 0.1 step yielding eleven points. Note that within a grid, a controller is able to perform well, to some extent, against faults in the WT which directly corresponds to the case of minor faults. For testing the proposed FTC scheme, we will focus on three significant faulty dynamics given by the grid values of $\Theta(t) \in \{0, 0.7, 1\}$. Consequently, we construct a bank of three controllers where the control structure is composed of a multi-variable Proportional+Integral (PI) controller:

$$\begin{aligned} \mathcal{C}_{\Theta_f(t)} \equiv C_e^{\Theta_f(t)}(\xi)u = C_u^{\Theta_f(t)}(\xi)(r-y), \qquad \text{where} \\ u = \begin{bmatrix} \beta_r \\ T_{g,r} \end{bmatrix}, r = \begin{bmatrix} \omega_{g,ref} \\ P_{rated} \end{bmatrix}, y = \begin{bmatrix} \omega_{g,m} \\ P_{g,m} \end{bmatrix} \end{aligned} \tag{7.32}$$

with $r = \texttt{col}(1173.7, 5)$ units. The controllers' polynomials are given by $C_e^0(\xi) = C_e^{0.7}(\xi) = C_e^1(\xi) = \xi$ and

$$C_u^0(\xi) = \begin{bmatrix} 2.746 \times 10^{-3}\xi + 6.76 \times 10^{-2} & 0 \\ 0 & 2.6\xi + 104 \end{bmatrix} \tag{7.33a}$$

$$C_u^{0.7}(\xi) = \begin{bmatrix} 1.563 \times 10^{-3}\xi + 3.38 \times 10^{-2} & 0 \\ 0 & 2.6\xi + 104 \end{bmatrix} \tag{7.33b}$$

$$C_u^1(\xi) = \begin{bmatrix} 0.1008\xi + 6.76 \times 10^{-2} & 0 \\ 0 & 2.6\xi + 104 \end{bmatrix} \tag{7.33c}$$

In addition, the implementable desired behavior is also provided by the AD phase for the WT operating in different modes. The control performance functional is given by

$$J = \int_{n\tau}^{(n+1)\tau} \|(\hat{r} - y)(\varsigma)\|_2^2 d\varsigma \times 10^{-3}, \tag{7.34}$$

where $\|\bullet\|_2$ denotes the Euclidean norm. The supervisor is constructed within the pitch and torque controller block, since this block contains all the trajectories, where the parameters of supervisor are taken as $\lambda = 2, \tau = 5sec$ [JYS13a].

An experimental setup considers the wind profile varying around the mean speed of 14 units, which is illustrated Fig. 7.10, together with fault scenarios as discussed in subsection 7.3.2, where the parametric values of the pitch system are taken as $\zeta_n = 0.6 = \zeta_f, \omega_{n,n} = 11.11, \omega_{n,f} = 0.2$. An experiment is run with an initial value of natural frequency as $\omega_n(0)$, and the operating controller in the closed-loop is $\mathcal{C}_0$. The first power drop fault appears within the WT at time 70 units, which changes the value of natural frequency to $\omega_n(0.7)$. Based on the theory developed in previous sections, controller $\mathcal{C}_{0.7}$ is then switched into the closed-loop at time 75 units. The second power drop fault appears at time 150 units, which affects the dynamics of the system by changing the value of natural frequency to $\omega_n(1)$. According to the controller selection logic, controller $\mathcal{C}_1$ is switched at time 160 units. The Simulink architecture of the fault-tolerant wind turbine control system is illustrated in Fig. 7.11, and the closed-loop signals are illustrated in Fig. 7.12. It has been shown here that

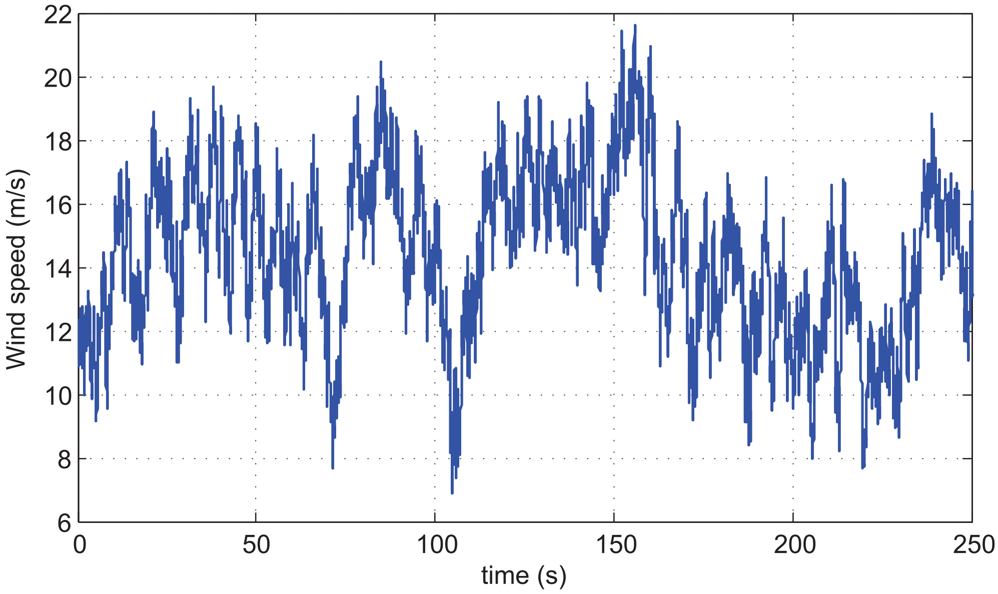


Figure 7.10: Wind profile

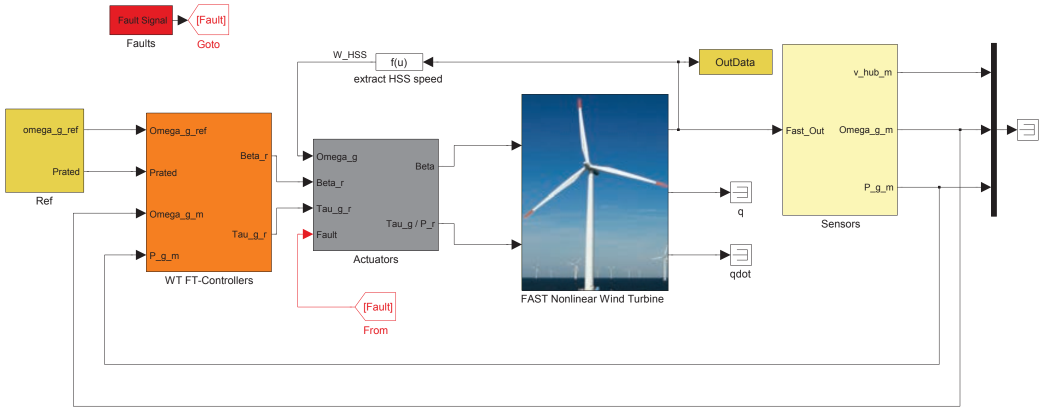


Figure 7.11: System-level Simulink Architecture of Projection-based Fault-Tolerant Wind Turbine Control

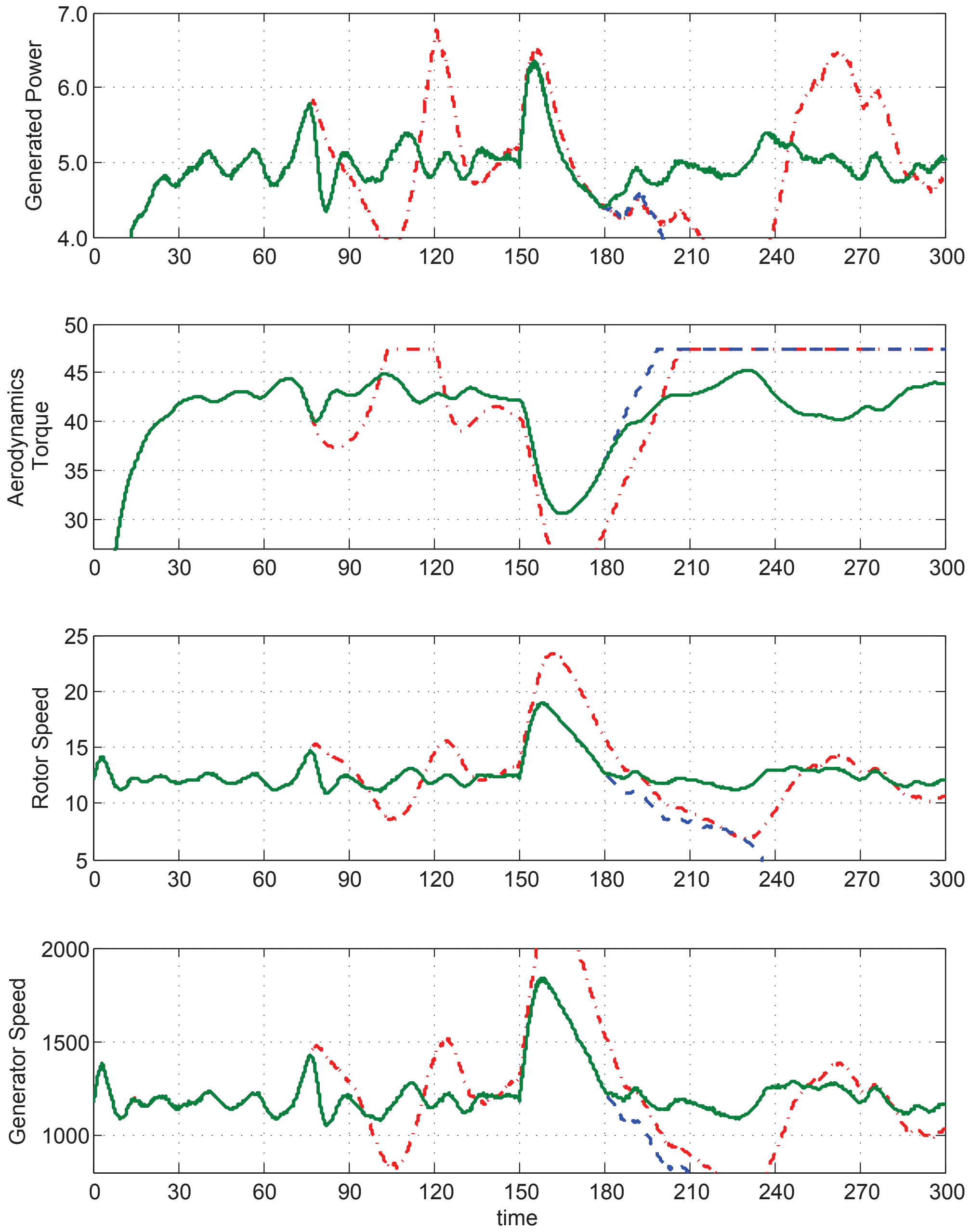


Figure 7.12: Closed-loop Signals of Fault-Tolerant Wind Turbine System: Dashed-dotted line denotes the case of without any FTC mechanism; Dashed line denotes the second faulty case; Solid line denotes the case of using the proposed FTC mechanism

without any fault-tolerant strategy, the dynamics of the system oscillates, and can even saturates that might damage some internal components of the Wind Turbine. However, with the proposed real-time fault-tolerant strategy, the behavior of the system satisfies the desired behavior at anytime.

7.3.5 Online-redesign based FTC

The WT model described in the previous sections is just discussed to show how the dynamics of the WT system evolves. However, we do not use any of this knowledge to demonstrate the proposed real-time fault-tolerant control mechanism. The two control inputs to the wind turbine are the generator torque $T_{g,r}$, and the blade pitch angle β_r. Since, we are working in Zone-3, the control objective is to track the generator power at its rated value of 5 units. As suggested in [OJ12], the main control scheme is developed in the torque control and the pitch controls from the industrial standpoint. FAST also requires a yaw angular velocity and a yaw angular position as inputs. Here no yaw system is installed within the benchmark model since the direction of the wind is considered to be perpendicular to the blades. The torque controller is a nonlinear controller which depends on the wind speed, and the pitch controller is a PI (Proportionl + Integral) controller. In the proposed fault-tolerant controller design, the control objective can be achieved by reconfiguring only the PI pitch controller, described by

$$\begin{gathered}
\mathcal{C}_{\Theta_f(t)} \equiv C_e^{\Theta_f(t)}(\xi)u = C_u^{\Theta_f(t)}(\xi)(r-y) \quad \text{where} \qquad (7.35)\\
u = \beta_r, r = \omega_{g,ref}, y = \omega_{g,m}\\
C_e^{\Theta_f(t)}(\xi) = \xi, C_u^{\Theta_f(t)}(\xi) = k_p^{\Theta_f(t)}\xi + k_i^{\Theta_f(t)}.
\end{gathered}$$

The nonlinear torque controller, embedded within the benchmark model, is not reconfigured here. With this consideraition, the closed-loop system can be viewed as a single-input single-output system.

In the AD phase, the parameter space $\Theta(t) \in [0,1]$ was gridded with a 0.1 step yielding eleven points. For testing the proposed FTC scheme, we will focus on two significant faulty dynamics given by the grid values of $\Theta(t) \in \{0, 0.9\}$. The AD phase also provides the implementable desired behavior $\mathcal{D}$ covering the parameter space $\Theta(t)$,

$$\mathcal{D} = \mathtt{ker}(\begin{bmatrix} D_r & -D_y \end{bmatrix}), \qquad (7.36)$$

where $D_r(\xi) = 49, D_y(\xi) = \xi^2 + 12.6\xi + 49$ with a measurement set observed during the interval of length $\tau = 10sec$.

The experimental setup considers the wind profile varying around the mean speed of 17 units, which is illustrated Fig. 7.13, together with fault scenarios as

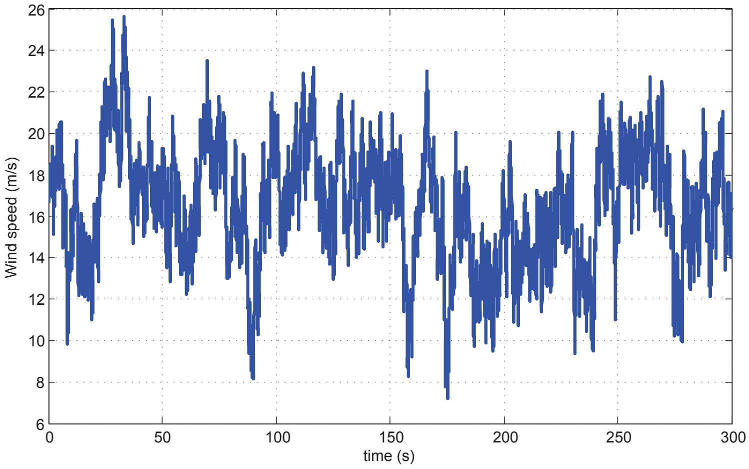


Figure 7.13: Wind profile used for simulation

discussed in subsection 7.3.2, where the parametric values of the pitch system are taken as $\zeta_n = 0.6, \omega_{n,n} = 11.11, \zeta_f = 0.1, \omega_{n,f} = 1$. An experiment is run with an initial value of pitch actuator parameters as $\omega_n(\Theta(t) = 0), \zeta(\Theta(t) = 0)$. The parameters of the initial controller operating in the closed-loop are computed as

$$k_p^0 = 2.746 \times 10^{-3}, \quad k_i^0 = 6.76 \times 10^{-2}. \tag{7.37}$$

A pressure power drop fault appears within the WT system at time 80 units, which changes the value of pitch parameters to $\omega_n(0.9), \zeta(0.9)$. Based on the theory developed in previous sections, a new controller is determined at every interval of length τ [JYS13d, JYS13b]. Thus, a controller $\mathcal{C}_{0.9}$ is then computed online and switched into the closed-loop at time 90 units. The computed parameters of $\mathcal{C}_{0.9}$ are given by

$$k_p^{0.9} = 5.113 \times 10^{-4}, \quad k_i^{0.9} = 7.044 \times 10^{-3}. \tag{7.38}$$

The Simulink architecture of a Fault-tolerant Wind Turbine system is illustrated in Fig. 7.14, and the closed-loop signal of the WT system is shown in Fig. 7.15. It has been shown here that without any fault-tolerant strategy, the dynamics of the system oscillates that might damage some internal components of the Wind Turbine. However, with the proposed real-time fault-tolerant strategy, the behavior of the system satisfies the desired behavior at anytime.

From the practical implementation point of view, avoiding the appearance of large transients during the switching of controllers is of utmost importance. These unpermitted transients due to an instant switching deteriorates the system performance. This effect is clearly visible in the illustrated fig-

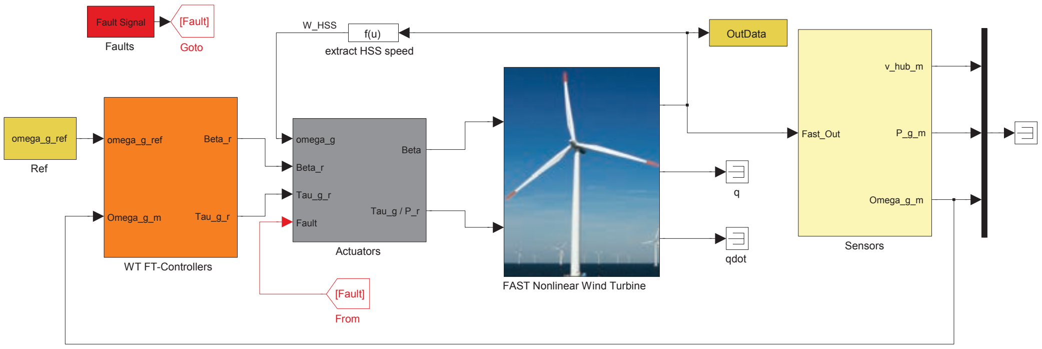


Figure 7.14: System-level Simulink Architecture of Online-redesign based Fault-Tolerant Wind Turbine Control

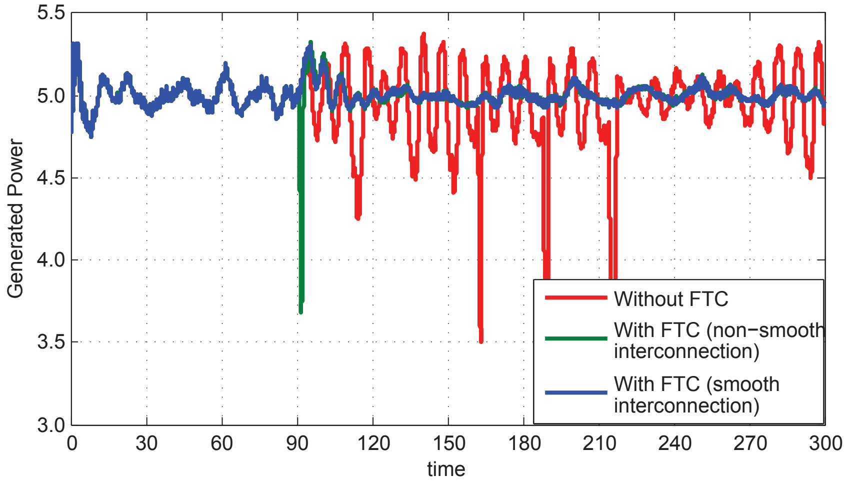


Figure 7.15: Closed-loop Signals of Fault-Tolerant Wind Turbine System

ure. Throughout running all the simulations, the interconnection between the unknown plant and the controller is considered to be smooth interconnection.

Chapter 8

Conclusions and future research

The work addressed in this thesis intended to develop novel approaches to deal with fault-tolerant control (FTC) systems. In this research, the novelty of demonstrated approaches lies in using the time-trajectory based viewpoint of behavioral system theory. Within this mathematical framework, a dynamical system is defined in terms of real-time trajectories generated by the system, which captures the *behavior* of the system. Acquiring this behavioral viewpoint, two novel approaches are presented lying under the taxonomy of active fault-tolerant systems, namely, the projection-based approach and the online redesign based approach. In the former approach, one of the controllers from the predefined controllers' bank is switched into the closed-loop, while in the latter, a new online controller is designed for the operating plant.

Generally, in the existing literature of fault-tolerant systems, FTC objectives are achieved by using the two cascaded operating modules as follows, fault diagnosis module and controller reconfiguration module. The latter module reconfigures the controller after incurring from the former module an accurate information about an occurred fault. It has been shown in the first part of the thesis that to obtain this information a precise knowledge of the operating plant is required during the fault diagnosis. This involves a substantial amount of time to be utilized at a runtime, which is one of the drawbacks considering the real-time constraints. However, in the proposed approaches, no *a priori* information about the plant's model is required during fault accommodation in real-time. This clearly excludes the need of the fault diagnosis module. The second part of the thesis demonstrated the design and implementation of the aforementioned approaches. It was shown within the proposed solutions, the controller reconfiguration process is based entirely on the trajectories generated by the system in real-time. Therefore, the issues related to the use of integrated FDD-CR approach for FTC systems, which are raised in Chapter 1 are not experienced here. In addition, the fault accommodation delay in these active FTC schemes is always smaller than the delay as experienced in the traditional schemes of AFTC systems. It is worth mentioning that the underlying aim of this thesis is not to recommend to totally abandon the use of the fault diagnosis module, but to emphasize together with resolving aforementioned issues that whenever an FTC problem is feasible, the fault accommodation delay can be reduced by using the proposed

FTC schemes.

In the projection-based approach, satisfying the desired behavior at any time is considered equivalent to the real-time fault-tolerant control objectives, which the system has to satisfy irrespective of an occurrence of a fault. However, the undesirable transients appearing at the switching instants, i.e. whenever a new controller is introduced in the closed-loop and makes an interconnection with the unknown plant, deteriorate the system performance as shown in the last part of the thesis. Further, it has been demonstrated that these transients can be suppressed by using the approach presented in Chapter 6, thereby guaranteed the real-time smooth interconnection. In addition, to establish the effectiveness of the above developed approaches, they are successfully implemented on the two-tank system, the aircraft during the landing phase, and the NREL's 5MW wind turbine system.

This work opens many pathways for possible future research and development. First of among all is the application of behavioral system theory to deal with fault-tolerant systems. To the best of our knowledge, not much work has been concretely reported in the literature, which deals with solving an FTC problem by taking this trajectory-based viewpoint apart from the work addressed in this manuscript. This research will certainly catch an attention from the scientific community so that further application of this nice mathematical theory can be investigated in the field of FTC systems dealing with several inherent issues. As an initiative step, the classical results of fault diagnosis are formulated within the behavioral system theory in [BVW06]. One of the most crucial steps in any fault-tolerant scheme against occurring fault is the Failure Mode and Effects Analysis. This analysis is performed at the outset, which is mandatory for several reasons, particularly how all known faults can be treated, how faults propagate within the system, how to change the properties of programmable components, etc. In the first chapter, basic ideas of analyzing the fault propagation are referred. Since this analysis is performed only in terms of system variables, it would be worth investigating this analysis procedure by taking the behavioral point of view.

Secondly, it has been shown in this work that the desired behavior captures the fault-tolerant control objectives. That is to say, whenever an FTC controller implements the desired behavior, the interconnected system satisfies the FTC objectives. However, in the projection based approach, a formal proof of measurement-based real-time stability without using any *a priori* information about the plant's model remains an open question. It has been stressed here that the closed-loop system is implicitly a "stand-alone" stable system in the pre-fault mode and the post-fault mode of an operating plant. The only source of instability at a run-time of the above FTC scheme arises due to the switching of controllers. Nevertheless, an *ad hoc* way of resolving this issue is to impose dwell-time constraints (see Appendix) on the switching criteria. An interesting question is: how this constraint can be relieved, equally guaranteeing the stability? Several works have been initiated in the direction of establishing data-based stability of LTI systems. In [PI09], dynamics of an LTI discrete-time system are represented as a subspace of a finite-dimensional vector space, called the data space, whose vectors correspond to all the subsequences of the time series. It is assumed that this data space contain all the dynamic behaviors of the system, hence, by using a basis matrix of the data space, data-based stability conditions for autonomous systems were established using the Lyapunov stability theory. All discussions in this so-called data space approach, were carried out under the assumptions of noise-free, and *a priori* known system orders. However, it is suggested that these assumptions can be exempted by considering a high-dimensional data space. Indeed, this leads to use a large amount of data that requires high computational costs. Another work on the stability test is initiated in [DLLA09], where a limited amount of experimental data and possibly noisy data is utilized, which is obtained with an existing known stabilizing controller connected to an unknown plant for verifying that the introduction of a new controller will stabilize the plant. The above discussion depends on the assumptions that the unknown plant is stabilized by a known controller and that some knowledge of the closed-loop system, such as noisy frequency response data, is available.

In the online redesign based approach, we have designed a data-driven controller based on the implementability of the given desired behavior. The relating question is: which desired behaviors can be implemented and can this be chosen arbitrarily? According to Willem's theorem, the implementability of the desired behavior depends entirely on the behavior of the plant. For a stable and minimum phase plant as defined in the classical sense, an arbitrarily chosen stable and minimum phase desired behavior can be implemented. However, the problem arises when the plant is an unstable and non-minimum phase. A deeper look at the interconnected system satisfying the desired behavior within Chapter 5 reveals that the controller implementing the desired behavior in the case of unity feedback configuration cancels the entire dynamics of the plant, see [YS12]. This clearly demands to embed the knowledge of at least unstable poles and unstable zeros of the plant in the desired behavior, which is, in fact a classical issue in the system theory. On the other side, when the approach is considered within the 2-DOF feedback configuration as presented in the later section of Chapter 5, the knowledge of unstable poles of the plant is not required. However, the knowledge of unstable zeros is still needed to be embedded within the desired behavior.

In this thesis, we have solved many prevailing issues related to the integrated FDD-CR design of fault-tolerant systems by shifting from the classical input/output processor standpoint of representing a dynamical system to an equivalent set of solutions. While solving these issues with a successful practical implementation, we have come across many other aspects as raised above, which call for a significant research to be done in this area.

Appendix A
Dwell-Time Switching

Consider a compact subset $\mathfrak{P}$ of a finite dimensional space, a parameterized family of $n \times n$ matrices $\mathcal{A} = \{A_p : p \in \mathfrak{P}\}$, and a family $\mathfrak{S}$ of piecewise-constant switching signals $\sigma : [0, \infty) \to \mathfrak{P}$, whose switching times are separated by τ times unit where τ is a pre-specified positive number called a *dwell time*. More precisely, $\sigma \in \mathfrak{S}$ is said to have dwell time τ if and only if σ switches values at most once, or if it switches more that once, the set of time differences between any two successive switches is bounded below τ.

Note that the class $\mathfrak{S}$ just defined contains constant switching signal $\sigma(t) = p, t?0$ for any value of $p \in \mathfrak{P}$. A necessary condition for A_σ to be exponentially stable for every $\sigma \in \mathfrak{S}$, is therefore that each $A_p \in \mathcal{A}$ is exponentially stable. In other words, if A_σ to be exponentially stable for every $\sigma \in \mathfrak{S}$, then for each $p \in \mathfrak{P}$ there must exist non-negative numbers t_p and ν_p, with ν_p positive such that $|e^{A_p t}| \leq e^{\nu_p(t_p - t)}, t \geq 0$. The symbol $| \bullet |$ denotes any norm on a finite dimensional linear space. It is quite easy to show by example that this condition is not sufficient unless τ is large. An estimate of how large τ has to be in order to guarantee exponential stability, is provided by the following lemma.

Lemma A.1 ([Mor08]). *Let $A_p : p \in \mathfrak{P}$ be a set of real, $n \times n$ matrices for which there are non-negative numbers t_p and ν_p with ν_p positive such that*

$$|e^{A_p t}| \leq e^{\nu_p(t_p - t)}, t \geq 0 \tag{A.1}$$

Suppose that τ is a finite number satisfying

$$\tau > t_p, p \in \mathfrak{P} \tag{A.2}$$

For any switching signal $\sigma : [0, \infty) \to \mathfrak{P}$ with dwell time τ, the state transition matrix of A_σ satisfies

$$|\Phi(t, \mu)| \leq e^{\nu(T - (t - \mu))}, \forall t \geq \mu \geq 0 \tag{A.3}$$

where ν is a positive number defined by

$$\nu = \inf_{p \in \mathfrak{P}} \left\{ \nu_p \left(1 - \frac{t_p}{\tau} \right) \right\} \tag{A.4}$$

and

$$T = \frac{2}{\nu} \sup_{p \in \mathfrak{P}} \{\nu_p t_p\}. \tag{A.5}$$

Moreover,

$$\nu \in (0, \nu_p], \quad p \in \mathfrak{P}. \tag{A.6}$$

Proof. Since $\mathfrak{P}$ is closed, bounded set, $\sup_{p \in \mathfrak{P}} t_p < \infty$, a finite τ satisfying (A.2) exists. Clearly $\nu_p(1 - \frac{t_p}{\tau}) > 0, p \in \mathfrak{P}$. From this and the definition of ν it follows that (A.6) holds and that

$$e^{\nu_p(t_p - \tau)} \leq e^{-\nu\tau}, \quad p \in \mathfrak{P}.$$

This and (A.1) imply that for $t \geq \tau$

$$\begin{aligned} |e^{A_p t}| &\leq e^{\nu_p(t_p - t)} = e^{\nu_p(t_p - \tau)} e^{-\nu_p(t - \tau)} \leq e^{-\nu\tau} e^{-\nu_p(t - \tau)} \\ &\leq e^{-\nu_p \tau} e^{-\nu(t - \tau)} \leq e^{-\nu t}, \quad t \geq \tau, p \in \mathfrak{P}. \end{aligned} \tag{A.7}$$

It also follows from (A.1) and the definition of T that

$$|e^{A_p t}| \leq e^{\nu(\frac{T}{2} - t)}, \quad t \in [0, \tau), \quad p \in \mathfrak{P}. \tag{A.8}$$

Set $t_0 = 0$ and let $t_1, t_2, \ldots$ denote the times at which σ switches. Write p_i for the value of σ on $[t_{i-1}, t_i$. Note that for $t_{j-1} \leq \mu \leq t_j \leq t_i \leq t \leq t_{i-1}$,

$$\Phi(t, \mu) = e^{A_{p_{i+1}}(t - t_i)} \left(\prod_{q=j+1}^{i} e^{A_{p_q}(t_q - t_{q-1})} \right) e^{A_{p_j}(t_j - \mu)}.$$

In view of (A.7) and (A.8)

$$\begin{aligned} |\Phi(t, \mu)| &\leq |e^{A_{p_{i+1}}(t - t_i)}| \left(\prod_{q=j+1}^{i} |e^{A_{p_q}(t_q - t_{q-1})}| \right) |e^{A_{p_j}(t_j - \mu)}| \\ &\leq e^{\nu(\frac{T}{2} - (t - t_i))} \left(\prod_{q=j+1}^{i} e^{-\nu(t_q - t_{q-1})} \right) e^{\nu(\frac{T}{2} - (t_j - \mu))} \\ &= e^{\nu(T - (t - \mu))} \end{aligned}$$

On the other hand, for $i > 0$, $t_{i-1} \leq \mu \leq t \leq t_i$, (A.8) implies that

$$|\phi(t, \mu)| \leq e^{\nu(\frac{T}{2} - (t - \mu))} \leq e^{\nu(T - (t - \mu))}$$

and so (A.3) is true. □

Appendix B

Direct Continuous-time Model Identification

B.1 The Traditional SVF Method

A continuous-time model of the system takes the form of a constant coefficient differential equation

$$\frac{d^n}{dt^n}y(t) + a_1\frac{d^{n-1}}{dt^{n-1}}y(t) + \cdots + a_n y(t) = b_0\frac{d^m}{dt^m}u(t) + \cdots + b_m u(t) \tag{B.1}$$

where $\frac{d^i}{dt^i}x(t)$ denotes the $i-$th time derivative of the continuous-time signal $x(t)$. Equation B.1 can be written alternatively as

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots + a_n y(t) = b_0 u^{(m)}(t) + \cdots + b_m u(t) \tag{B.2}$$

where $x^{(i)}(t)$ denotes the $i-$th time derivative of the continuous-time signal $x(t)$. Equation B.1 or B.2 can be written in the alternative time-domain differential operator form

$$A(\xi)y(t) = B(\xi)u(t) \tag{B.3}$$

with

$$B(\xi) = b_0\xi^m + b_1\xi^{m-1} + \cdots + b_m \tag{B.4}$$

$$A(\xi) = \xi^n + a_1\xi^{n-1} + \cdots + a_n, n \geq m \tag{B.5}$$

and ξ denoting the differential operator [GW08].

Assume now that a state-variable filter (SVF) with operator model $F(\xi)$ is applied to both sides of B.3. Then, ignoring transient initial conditions

$$A(\xi)F(\xi)y(t) = B(\xi)F(\xi)u(t) \tag{B.6}$$

The minimum-order SVF filter is typically chosen to have the following operator model form[1]

$$F(\xi) = \frac{1}{(\xi + \lambda)^n} \tag{B.7}$$

[1] The filter dc gain can be made unity if this is thought desirable.

where λ is the parameter that can be used to define the bandwidth of the filter.

Equation B.6 can then be rewritten, in expanded form, as

$$\left(\frac{\xi^n}{(\xi+\lambda)^n} + a_1\frac{\xi^{n-1}}{(\xi+\lambda)^n} + \cdots + a_n\frac{1}{(\xi+\lambda)^n}\right) y(t)$$
$$= \left(b_0\frac{\xi^m}{(\xi+\lambda)^n} + \cdots + b_m\frac{1}{(\xi+\lambda)^n}\right) u(t) \quad \text{(B.8)}$$

Let $F_i(\xi)$ for $i = 0, 1, \ldots, n$ be a set of filters defined as

$$F_i(\xi) = \frac{\xi^i}{(\xi+\lambda)^n} \quad \text{(B.9)}$$

By using the filters defined in B.9, equation B.8 can be rewritten, as

$$(F_n(\xi)+a_1F_{n-1}(\xi)+\ldots+a_nF_0(\xi))y(t) = (b_0F_m(\xi)+\ldots+b_mF_0(\xi))u(t). \quad \text{(B.10)}$$

Equation B.10 can also be written as

$$y_f^{(n)}(t) + a_1 y_f^{(n-1)}(t) + \ldots + a_n y_f^{(0)}(t) = b_0 u_f^{(m)}(t) + \ldots + b_m u_f^{(0)}(t) \quad \text{(B.11)}$$

with

$$y_f^{(i)}(t) = f_i(t) * y(t)$$
$$u_f^{(i)}(t) = f_i(t) * u(t)$$

where $f_i(t)$, for $i = 0, \ldots, n$ represent the impulse responses of the filters defined in B.9 and $*$ denotes the convolution operator. The filter outputs $y_f^{(i)}(t)$ and $u_f^{(i)}(t)$ provide *prefiltered* time derivatives of the inputs and outputs in the bandwidth of interest, which may then be exploited for model parameter estimation.

At time instant $t = t_k$, equation B.11 can be rewritten in standard linear regression-like form as

$$y_f^{(n)}(t_k) = \varphi_f^T(t_k)\theta \quad \text{(B.12)}$$

where

$$\varphi_f^T(t_k) = \begin{bmatrix} -y_f^{(n-1)}(t_k) & \cdots & -y_f^{(0)}(t_k) & u_f^{(m)}(t_k) & \cdots & u_f^{(0)}(t_k) \end{bmatrix} \quad \text{(B.13)}$$

$$\theta = \begin{bmatrix} a_1 & \cdots & a_n & b_0 & \cdots & b_m \end{bmatrix}^T. \quad \text{(B.14)}$$

Now, from N available samples of the input and output signals observed at discrete times $t_1, \ldots, t_N$, not necessarily uniformly spaced, the linear least-squares (LS)based SVF parameter estimates are given by

$$\hat{\theta}_{LSSVF} = \left[\frac{1}{N}\sum_{k=1}^{N}\varphi_f(t_k)\varphi_f^T(t_k)\right]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi_f(t_k)y_f^{(n)}(t_k). \quad \text{(B.15)}$$

Bibliography

[AAB$^+$01] K. Astrom, P. Albertos, M. Blanke, A. Isidori, W. Schaufelberger, and R. Sanz. *Control of Complex Systems*. Springer Verlag, 2001. (Cited on page 21.)

[AKKH09] M. Al-Kuwaiti, N. Kyriakopoulos, and S. Hussein. A comparative analysis of network dependability, fault-tolerance, reliability, security, and survivability. *IEEE Communications Letters Surveys & Tutorials*, 11(2):106–124, 2009. (Cited on page 12.)

[AT03] M. Araki and H. Taguchi. Tutorial paper on two-degree-of-freedom pid controllers. *International Journal of Control, Automation, and Systems*, 1:401–411, 2003. (Cited on page 95.)

[AW95] K. J. Åström and B. Wittenmark. *Adaptive Control*. Addison-Wesley, 1995. (Cited on page 11.)

[BBM05] J. Bošković, S. F. Bergstrom, and R. Mehra. Adaptive accommodation of failures in second-order fight control actuators with measurable rates. In *IEEE American Control Conference*, 2005. (Cited on page 24.)

[Bel03] M. N. Belur. *Control in Behavioral context*. PhD thesis, University of Groningen, 2003. (Cited on page 45.)

[BIZL98] M. Blanke, R. I-Zamanabadi, and T.F. Lootsma. Fault monitoring and re-configurable control for a ship propulsion plant. *International Journal of Adaptive Control and Signal Processing*, 12(8):671–688, 1998. (Cited on page 12.)

[BKSL03] M. Blanke, M. Kinnaert, M. Staroswiecki, and J. Lunze. *Diagnosis and Fault tolerant control*. Springer-Verlag, 2003. (Cited on pages 11, 12, 22, 26, 29, 32, 74, and 80.)

[Bla96] M. Blanke. Consistent design of dependable control systems. *Control Eng. Practice*, 4:1305–1312, 1996. (Cited on page 27.)

[BM98] J. Bošković and R. Mehra. A multiple model-based reconfigurable flight control system design. In *37th IEEE Conference on Decision and Control (CDC98)*, 1998. (Cited on page 24.)

[BPD99] S.M. Bennett, R. J. Patton, and S. Daley. Sensor fault-tolerant control of a rail traction drive. *Control Eng. Practice*, 7(2):217–225, 1999. (Cited on page 12.)

[BT02] M. N. Belur and H. L. Trentelman. Stabilization, pole placement and regular implementability. *IEEE Transactions on Automatic Control*, 47(5):735–744, 2002. (Cited on pages 65 and 71.)

[BVW06] M. Bisiacco, M. E. Valcher, and J. C. Willems. A behavioral approach to estimation and dead-beat observer design with applications to state-space models. *IEEE Transactions on Automatic Control*, 51(11):1787–1797, 2006. (Cited on pages 32, 48, and 140.)

[CDZ03] D. U. Campos-Delgado and K. Zhou. Reconfigurable fault-tolerant control using gimc structure. *IEEE Transactions on Automatic Control*, 48:832–838, 2003. (Cited on page 23.)

[Che99] Chi-Tsong Chen. *Linear Systsem Theory and Design*. Harcourt Brace College Publishers (Oxford University Press), 1984 (1999). (Cited on pages 35 and 49.)

[CP99] J. Chen and R. J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999. (Cited on pages 38 and 74.)

[DLLA09] A. Dehghani, A. Lecchini, A. Lanzon, and B. D. O. Anderson. Validating controllers for internal stability utilizing closed-loop data. *IEEE Transactions On Automatic Control*, 54:2719–2725, 2009. (Cited on page 141.)

[Don09] J. Dong. *Data Driven Fault Tolerant Control: A Subspace Approach*. PhD thesis, Delft Center for Systems and Control, TU Delft, 2009. (Cited on page 25.)

[DZN$^+$09] S.X. Ding, P. Zhang, A. Naik, E.L. Ding, and B. Huang. Subspace method aided data-driven design of fault detection and isolation systems. *Journal of Process Control*, 19(9):1496–1510, 2009. (Cited on pages 25 and 29.)

[EWLW85] J. S. Eterno, J. L. Weiss, D. P. Looze, and A. S. Willsky. Design issues for fault tolerant-restructurable aircraft control. In *24th IEEE conference on decision and control*, 1985. (Cited on pages 12 and 13.)

[GA91] Z. Gao and P. J. Antsaklis. Stability of the pseudo-inverse method for reconfigurable control systems. *International Journal of Control*, 53:717–729, 1991. (Cited on page 19.)

[GCW$^+$03] M. Guler, S. Clements, L.M. Wills, B.S. Heck, and G.J. Vachtsevanos. Transition management for reconfigurable hybrid control systems. *Control Systems, IEEE*, 23(1):36 – 49, feb 2003. (Cited on page 28.)

[GW08] H. Garnier and L. Wang, editors. *Identification of Continuous-time Models from Sampled Data*. Springer-Verlag London Limited, 2008. (Cited on pages 95, 100, 122, and 145.)

[Him78] D.M. Himmelblau. *Fault detection and diagnosis in chemical and petrochemical processes.* Elsevier Scientific Pubs., 1978. (Cited on pages 9 and 12.)

[HKH87] R. Hanus, M. Kinnaert, and J.L. Henrotte. Conditioning technique, a general anti-windup and bumpless transfer method. *Automatica*, 23:729–739, 1987. (Cited on page 28.)

[HKKS10] I. Hwang, S. Kim, Y. Kim, and C.E. Seah. A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, 18(3):636–653, May 2010. (Cited on page 12.)

[HTYJLM09] Wang Hong, Chai Tian-You, Ding Jin-Liang, and Brown Martin. Data driven fault diagnosis and fault tolerant control: Some advances and possible new directions. *ACTA AUTOMATICA SINICA*, 35:739–747, 2009. (Cited on pages 25 and 74.)

[Ise84] R. Isermann. Process fault detection based on modeling and estimation methods - a survey. *Automatica*, 20(4), 1984. (Cited on pages 9, 12, and 74.)

[Ise97] R. Isermann. Trends in the application of model based fault detection and diagnosis of technical processes. *Control Eng. Practice*, 5(5):709–719, 1997. (Cited on page 3.)

[JB09] J. Jonkman and M. Buhl. FAST user's guide. Technical report, NREL/EL-500-46198, 2009. (Cited on page 125.)

[JWBT05] A.A. Julius, J. C. Willems, M. N. Belur, and H. L. Trentelman. The canonical controllers and regular interconnection. *Systems*

& Control Letters, 54(8):787–797, 2005. (Cited on pages 71 and 92.)

[JY12] Jin Jiang and Xiang Yu. Fault-tolerant control systems: A comparitive study between active and passive approaches. *Annual Reviews in Control*, xx:xxxx, 2012. (Cited on page 104.)

[JYS10a] T. Jain, J. J. Yamé, and D. Sauter. A model based 2-dof fault tolerant control strategy. In *18th IEEE Med. Conference on Control and Automation*, pages 1073–1078, June 2010. (Cited on page 11.)

[JYS10b] T. Jain, J. J. Yamé, and D. Sauter. A model based 2-dof fault tolerant control strategy. In *18th Mediterranean Conference on Control & Automation (MED'10), Morocco*, pages 1073–1078. IEEE, 2010. (Cited on pages 73 and 87.)

[JYS10c] T. Jain, J. J. Yamé, and D. Sauter. A real time router fault accommodation. In *Conference on Control and Fault-Tolerant Systems (SysTol'10), France*, pages 899–903. IEEE, 2010. (Cited on page 87.)

[JYS11a] T. Jain, J. J. Yamé, and D. Sauter. Data-driven fault tolerant control. In *PAPYRUS Workshop on Fault Diagnosis and Fault Tolerant Control in large scale processing industries, Corsica-France*, page CDROM, 2011. (Cited on page 87.)

[JYS11b] T. Jain, J. J. Yamé, and D. Sauter. A note on performance evaluation in data-driven fault tolerant control. In *Conférence Méditerranéenne sur l'Ingénierie Sûre des Systèmes Complexes (MISC 2011), Morocco*, 2011. (Cited on page 73.)

[JYS11c] T. Jain, J. J. Yamé, and D. Sauter. Synergy of canonical control and unfalsified control concept to achieve fault tolerance. In *18th World IFAC Congress, Italy*, pages 14832–14837, 2011. (Cited on pages 28, 73, and 87.)

[JYS12a] T. Jain, J. J. Yamé, and D. Sauter. A behavioral approach to fault-tolerant control. *to be submitted*, 2012. (Cited on page 87.)

[JYS12b] T. Jain, J. J. Yamé, and D. Sauter. Case study on behavioral approach to fault-tolerant control: Application to an electric circuit. In *12th IEEE International Conference on Control, Automation, Robotics and Vision, China*, 2012. (Cited on page 87.)

[JYS12c] T. Jain, J. J. Yamé, and D. Sauter. Model-free reconfiguration mechanism for fault tolerance. *International Journal of Applied Mathematics and Computer Sciences*, 22(1):125–137, 2012. (Cited on pages 28, 73, and 116.)

[JYS12d] T. Jain, J. J. Yamé, and D. Sauter. On active fault-tolerant control in behavioral context. In *International Symposium on Security and Safety of Complex Systems (2SCS), Morocco*, **Best Theoretical Paper**, 2012. (Cited on pages 28 and 87.)

[JYS12e] T. Jain, J. J. Yamé, and D. Sauter. On implementing on-line designed controller for smooth interconnection in the behavioral framework. In *7th IFAC Symposium on Robust Control Design (ROCOND'12), Denmark*, 2012. (Cited on page 103.)

[JYS12f] T. Jain, J. J. Yamé, and D. Sauter. Role of performance evaluator in data-driven fault tolerant control. In *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SafeProcess 2012), Mexico*, 2012. (Cited on pages 30 and 73.)

[JYS12g] T. Jain, J. J. Yamé, and D. Sauter. Time-trajectory based active fault tolerant control. In *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SafeProcess 2012), Mexico*, 2012. (Cited on pages 87 and 121.)

[JYS13a] T. Jain, J. J. Yamé, and D. Sauter. A novel approach to real-time fault accommodation in nrel's 5mw wind turbine systems. *IEEE Transactions on Sustainable Energy*, submitted, 2013. (Cited on pages 73 and 131.)

[JYS13b] T. Jain, J. J. Yamé, and D. Sauter. On-line redesign based approach for fault accommodation in wind turbines. In *IEEE Conference on Innovative Smart Grid Technologies (ISGT 2013)*, 2013. (Cited on pages 87 and 135.)

[JYS13c] T. Jain, J. J. Yamé, and D. Sauter. On real-time smooth interconnection of on-line synthesized controllers in the behavioral framework. *submitted*, 2013. (Cited on page 103.)

[JYS13d] T. Jain, J. J. Yamé, and D. Sauter. Online redesign based real-time fault accommodation in nrel's 5mw wind turbine systems. In *IEEE International Conference on Industrial Technology (ICIT)*, 2013. (Cited on pages 87 and 135.)

[JYS13e] T. Jain, J. J. Yamé, and D. Sauter. Projection-based fault accommodation in nrel's 5mw wind turbine. In *IEEE American Control Conference*, 2013. (Cited on pages 73 and 128.)

[JYS13f] T. Jain, J. J. Yamé, and D. Sauter. Trajectory-based real-time control of electrical circuit against unknown faults. *International Journal of Electronics and Communications, Elsevier*, to be submitted, 2013. (Cited on page 87.)

[JZ00] J. Jiang and Q. Zhao. Design of reliable control systems possessing actuator redundancies. *Journal of Guidance, Control, and Dynamics*, 23(4):709–718, 2000. (Cited on page 17.)

[KA96] I. K. Konstantopoulos and P. J. Antsaklis. Eigenstructure assignment in reconfigurable control systems. Technical Report ISIS-96-001, ISIS Group at the University of Notre Dame, January 1996. (Cited on page 19.)

[Kan04] S. Kanev. *Robust Fault Tolerant Control*. PhD thesis, University of Twente, The Netherlands, 2004. (Cited on page 16.)

[KDY09] M. Kinnaert, T. Delwiche, and J. Yamé. State resetting for bumpless switching in supervisory control. In *Proc. European Control Conference (EUCA'09)*, pages 2097–2102, 2009. (Cited on page 108.)

[KM99] E. Kerrigan and J. Maciejowski. Fault-tolerant control of a ship propulsion system using model predictive control. In *Proceedings of the 5th European Control Conference (ECC99)*, 1999. (Cited on page 22.)

[KPH97] M. Keating, M. Pachter, and C. Houpis. Fault tolerant flight control system: Qft design. *International Journal of Robust and Non-Linear Control*, 7(6):551–559, 1997. (Cited on page 17.)

[KV00] S. Kanev and M. Verhaegen. Controller reconfiguration for nonlinear systems. *Control Eng. Practice*, 8(11):1223–1235, 2000. (Cited on page 23.)

[Lib03] D. Liberzon. *Switching in systems and control*. Boston: Birkauser, 2003. (Cited on page 25.)

[LWEB85] D. Looze, J. Weiss, J. Eterno, and N. Barrett. An automatic redesign approach for restructurable control systems. *Control*

Systems Magazine, IEEE, 5(2):16 –22, may 1985. (Cited on page 14.)

[MGB08] T. Miksch, A. Gambier, and E. Badreddin. Real-time performance comparison of fault-tolerant controllers. In *IEEE International Conference on Control Applications (CCA 2008)*, pages 492 – 497, 2008. (Cited on page 22.)

[MHRC89] D. D. Moerder, N. Halyo, Broussard J. R., and A. K. Caglayan. Application of pre-computed control laws in a reconfigurable aircraft flight control system. *Journal of Guidance, Control, and Dynamics*, 12:325–333, 1989. (Cited on page 14.)

[MJ03] J. M. Maciejowski and C.N. Jones. MPC fault-tolerant flight control case study: Flight 1862. In *SAFE Process 2003: 5th Symposium on Fault Detection and Safety for Technical Processes*, 2003. (Cited on pages 5, 21, and 22.)

[MJL08] E. Mazars, I. M. Jaimoukha, and Z. Li. Computation of a reference model for robust fault detection and isolation residual generation. *Journal of Control Science and Engineering*, Volume 2008, 2008. (Cited on page 90.)

[Mor08] A.S. Morse. Lectures notes on logically switched dynamical systems. In *Nonlinear and Optimal Control Theory, Lectures Notes in Mathematics*. Springer-Verlag, 2008. (Cited on page 143.)

[NP09] H. Niemann and N. K. Poulsen. Fault tolerant control - a residual based set-up. In *Joint 48th IEEE Conference of Decision and Control and 28th Chinese Control Conference*, 2009. (Cited on page 23.)

[NS03] H. Niemann and J. Stoustrup. Passive fault tolerant control of a double inverted penduluma case study example. In *Proceedings of the 5th Symposiumon Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS2003).*, pages 1029–1034, 2003. (Cited on page 17.)

[NS05] H. H. Niemann and J. Stoustrup. An architecture for fault tolerant control. *International Journal of Control*, 78:1091–1110, 2005. (Cited on page 22.)

[NSA04] H. Niemann, J. Stoustrup, and R. B. Abrahamsen. Switching between multivariable controllers. *Optim. Control Appl. Meth.*, 25:51–66, 2004. (Cited on page 28.)

[NSHT00] H. Noura, D. Sauter, F. Hamelin, and D. Theilliol. Fault tolerant control in dynamic systems: Application to a winding machine. *IEEE Control Systems Magazine*, 20(1):33–49, 2000. (Cited on pages 8, 12, and 19.)

[OJ12] P. F. Odgaard and K. E. Johnson. Wind turbine fault detection and fault tolerant control - a second challenge. In *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, 2012. (Cited on pages 127, 130, and 134.)

[OMB$^+$02] M. Oishi, I. Mitchell, A. Bayen, C. Tomlin, and A. Degani. Hybrid verification of an interface for an automatic landing. In *41st IEEE Conference on Decision and Control*, 2002. (Cited on page 113.)

[Ost85] Aaron J. Ostroff. Techniques for accommodating control effector failures on a mildly statically unstable airplane. In *American Control Conference, 1985*, pages 906 –913, june 1985. (Cited on page 18.)

[PAGB10] M. Pasamontes, J.D. Álvarez, J.L. Guzmán, and M. Berenguel. Bumpless switching in control - a comparative study. In *15th IEEE Conf. on Emerging Technologies and Factory Automation*, 2010. (Cited on page 28.)

[Pat93] R. J. Patton. Robustness issues in fault-tolerant control. In *IEE Colloquium on Fault Diagnosis and Control System Reconfiguration*, 1993. (Cited on page 13.)

[Pat97] R. J. Patton. Fault-tolerant control systems: The 1997 situation. In *Proc. 3rd IFAC symposium on fault detection, supervision and safety for technical processes*, pages 1033–1055, 1997. (Cited on pages ix, 10, 13, 21, and 28.)

[PI09] U. S. Park and M. Ikeda. Stability analysis and control design of LTI discrete-time systems by the direct use of time series data. *Automatica*, 45:1265–1271, 2009. (Cited on page 141.)

[Pol00] J. W. Polderman. Sequential continuous time adaptive control: A behavioral approach. In *Proc. 39th IEEE Conf. Decision Control*, pages 2484–2487, 2000. (Cited on page 71.)

[PTA10] J.C. Ponsart, D. Theilliol, and C. Aubrun. Virtual sensors design for active fault tolerant control system applied to a winding machine. *Control Engineering Practice*, 18:1037–1044, 2010. (Cited on page 27.)

[PW97] J. W. Polderman and J. C. Willems. *Introduction to Mathematical Systems Theory: A Behavioral Approach*. Springer-Verlag, 1997. (Cited on pages 40, 41, 42, 43, 44, 45, 46, 47, 50, 51, 52, 53, 56, 64, and 107.)

[RW01] P. Rocha and J. Wood. Trajectory control and interconnection of 1D and nD systems. *SIAM Journal of Control Optimization*, 40:107–134, 2001. (Cited on pages 32, 64, and 65.)

[sit] http://www.bhopal.com/faq.htm. (Cited on page 4.)

[SKP00] G. Simon, T. Kovacshazy, and G. Peceli. Transients in reconfigurable control loops. In *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference*, 2000. (Cited on page 28.)

[SP97] J.T. Spooner and K.M. Passino. Fault tolerant control for automated highway systems. *IEEE Transaction on Vehicular Technology*, 46(3):770–785, 1997. (Cited on page 12.)

[ST97] M.G. Safonov and T-C. Tsao. The unfalsified control concept and learning. *IEEE Transactions on Automatic Control*, 42(6):843–847, 1997. (Cited on pages 28 and 84.)

[Sta02] M. Staroswiecki. On reconfigurability with respect to actuator failures. In *15th Triennial World Congress*, 2002. (Cited on page 26.)

[Sta04] M. Staroswiecki. Progressive accommodation of actuator faults in the linear quadratic control problem. In *43rd IEEE Conference on Decision and Control*, 2004. (Cited on pages 27, 30, 74, and 78.)

[Sta05] M. Staroswiecki. Fault tolerant control: the pseudo-inverse method revisited. In *16th IFAC World Congress*, 2005. (Cited on page 19.)

[Ste05] M. Steinberg. Historical overview of research in reconfigurable flight control. *Proceedings of IMechE, Part G: Journal of Aerospace Engineering*, 219:263–275, 2005. (Cited on page 12.)

[TCJ02] G. Tao, S. Chen, and S. M. Joshi. An adaptive control scheme for systems with unknown actuator failures. *Automatica*, 38:1027 1034, 2002. (Cited on pages 8 and 21.)

[TMM97] T.T. Tay, I.M.Y. Mareels, and J.B. Moore. *High Performance Control*. Birkhäuser, 1997. (Cited on page 22.)

[TNS98] D. Theilliol, H. Noura, and D. Sauter. Fault-tolerant control method for actuator and component faults. In *Proceedings of the 37th IEEE Conference on Decision and Control*, 1998. (Cited on pages 9 and 19.)

[TSP03] D. Theilliol, D. Sauter, and J. Ponsart. A multiple model based approach for fault tolerant control in non linear systems. In *Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes*, 2003. (Cited on page 23.)

[TW02] H. L. Trentelman and J. C. Willems. Synthesis of dissipative systems using quadratic differential forms: Part II. *IEEE Transactions On Automatic Control*, 47:70–86, 2002. (Cited on pages 32 and 71.)

[vdS03] A.J van der Schaft. Achievable behavior of general systems. *Systems & Control Letters*, 49(2):141–149, 2003. (Cited on pages 32, 71, and 91.)

[VMP92] R. Veillette, J.B. Medanic, and W.R. Perkins. Design of reliable control systems. *IEEE Transactions on Automatic Control*, 37(3):290–304, 1992. (Cited on page 17.)

[VRYK03a] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri. A review of process fault detection and diagnosis part i: Quantitative model-based methods. *Computers and Chemical Engineering*, 27:293–311, 2003. (Cited on page 15.)

[VRYK03b] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri. A review of process fault detection and diagnosis part ii: Qualitative model and search strategies. *Computers and Chemical Engineering*, 27:313–326, 2003. (Cited on page 15.)

[VW99] M. E. Valcher and J. C. Willems. Observer synthesis in the behavioral approach. *IEEE Transactions on Automatic Control*, 44(12):2297–2307, 1999. (Cited on pages 32 and 48.)

[Wil91] J. C. Willems. Paradigms and puzzles in the theory of dynamic systems. *IEEE Transactions on Automatic Control*, 36:259–294, 1991. (Cited on pages 32, 35, 36, and 97.)

[Wil97] J. C. Willems. On interconnections, control, and feedback. *IEEE Transactions on Automatic Control*, 42:326–339, 1997. (Cited on pages 32, 49, 65, and 66.)

[Wil07] J. C. Willems. The behavioral approach to open and interconnected systems: Modeling by tearing, zooming, and linking. *IEEE Control Systems Magazine*, 27:46–99, 2007. (Cited on pages 32 and 61.)

[WT02] J. C. Willems and H. L. Trentelman. Synthesis of dissipative systems using quadratic differential forms: Part I. *IEEE Transactions On Automatic Control*, 47:53–69, 2002. (Cited on pages 70, 91, and 97.)

[Wu04a] E. Wu, editor. *IFAC Proceedings: Fault Detection, Supervision and Safety of Technical Processes*. Elsevier, 2004. (Cited on page 12.)

[Wu04b] N. E. Wu. Coverage in fault-tolerant control. *Automatica*, 40(4):537 – 548, 2004. (Cited on pages 26 and 80.)

[Yam05] J. J. Yamé. Modeling and simulation of an aircraft in landing approach. Technical report, Centre de Recherche en Automatique de Nancy (CRAN), Nancy, France, 2005. (Cited on page 115.)

[YCJ10] H. Yang, V. Cocquempot, and B. Jiang. Supervisory fault tolerant control design via switched system approach. In *IEEE Conference on Control and Fault-Tolerant Systems, Systol'10*, 2010. (Cited on pages 24 and 25.)

[YIZB03] Z. Yang, R. Izadi-Zamanabadi, and M. Blanke. On-line multiple-model based adaptive control reconfiguration for a class of non-linear control systems. In *SAFEPROCESS*, 2003. (Cited on page 23.)

[YJ11] X. Yu and J. Jiang. Hybrid fault-tolerant flight control system design against partial actuator failures. *Control Systems Technology, IEEE Transactions on*, PP(99):1 –16, 2011. (Cited on page 29.)

[YJB76] D.C. Youla, H.A. Jabr, and J.J. Bongiorna. Modern Wiener-Hopf design of optimal controllers - Part I: The single-input-output case. *IEEE Transactions on Automatic Control*, 21(3):319–338, 1976. (Cited on page 22.)

[YJC11] H. Yang, B. Jiang, and V. Cocquempot. Supervisory fault-tolerant regulation for nonlinear systems. *Nonlinear Analysis: Real World Applications*, 12:789–798, 2011. (Cited on page 25.)

[YK04a] J. J. Yamé and M. Kinnaert. A fault accommodation strategy based on closed-loop performance monitoring. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, 2004. (Cited on page 28.)

[YK04b] J. J. Yamé and M. Kinnaert. Performance-based switching for fault-tolerant control. In Eva Wu and Marcel Staroswiecki, editors, *Fault Detection, Supervision And Safety of Technical Processes*, 2004. (Cited on page 74.)

[YK05] J.J. Yamé and M. Kinnaert. Performance-based supervisory fault-tolerant control with application to the ifatis two-tanks benchmark. In *16th IFAC World Congress*, 2005. (Cited on page 25.)

[YK07] J. J. Yamé and M. Kinnaert. On bumps and reduction of switching transients in multicontroller systems. *Mathematical Problems in Engineering*, Volume2007, 2007. (Cited on pages 28 and 106.)

[YS08] J.J. Yamé and D. Sauter. A real-time model-free reconfiguration mechanism for fault-tolerance: Application to a hydraulic process. In *Proc. 10th International Conference on Control, Automation, Robotics and Vision(ICARV)*, pages 91–96, 2008. (Cited on page 119.)

[YS12] J. J. Yamé and D. Sauter. A note on the canonical controller in the behavioral system-theoretic approach. In *12th International Conference on Control, Automation, Robotics and Vision*, 2012. (Cited on page 141.)

[YWDC06] L. Yew-Wen and L. Der-Cherng. Common stabilizers for linear control systems in the presence of actuators outage. *Applied Mathematics and Computation*, 177(2):635 – 643, 2006. (Cited on page 26.)

[YWS00] G.H. Yang, J.L. Wang, and Y.C. Soh. Reliable LQG control with sensor failures. *IEE Proc.Control Theory Appl.*, 147(4):428–432, 2000. (Cited on page 17.)

[YY06] D. Ye and G-H. Yang. Adaptive fault-tolerant tracking control against actuator faults with application to flight control. *IEEE Transactions on Control Systems Technology*, 14:1088–1096, 2006. (Cited on page 75.)

[ZJ98] Q. Zhao and J. Jiang. Reliable state feedback control systems design against actuator failures. *Automatica*, 34(10):1267–1272, 1998. (Cited on page 17.)

[ZJ01] Y.M. Zhang and J. Jiang. Integrated active fault-tolerant control using IMM approach. *IEEE Transactions on Aerospace and Electronic Systems*, 37(4):1221–1235, 2001. (Cited on pages 19 and 23.)

[ZJ02] Y. Zhang and J. Jiang. Design of reconstructable active fault-tolerant control systems. In *15th Triennial IFAC World Congress*, 2002. (Cited on page 21.)

[ZJ08] Y. Zhang and J. Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32:229–252, 2008. (Cited on pages 4, 5, 18, 26, 27, 28, and 103.)

[ZR01] K. Zhou and Z. Ren. A new controller architecture for high performance, robust, and fault-tolerant control. *IEEE Transactions on Automatic Control*, 46(10):1613–1618, 2001. (Cited on pages 17 and 22.)