Mogens Blanke · Michel Kinnaert · Jan Lunze · Marcel Staroswiecki

Diagnosis and Fault-Tolerant Control

Mogens Blanke · Michel Kinnaert
Jan Lunze · Marcel Staroswiecki

Diagnosis and Fault-Tolerant Control

With contributions by Jochen Schröder

2nd Edition

With 270 Figures, 121 Examples, 5 Application Studies, and 36 Exercises

Springer

Prof. Dr. Mogens Blanke
Technical University of Denmark
Section of Automation at Ørsted · DTU
2800 Lyngby, Denmark
and
Norwegian University
of Science and Technology
7491 Trondheim, Norway
mb@oersted.dtu.dk

Prof. Dr. Michel Kinnaert
Université Libre de Bruxelles
Laboratoire d'Automatique
CP 165
50 Ave. F.D. Roosevelt
1050 Bruxelles, Belgium
Kinnaert@labauto.ulb.ac.be

Prof. Dr.-Ing. Jan Lunze
Ruhr-Universität Bochum
Lehrstuhl für Automatisierungstechnik
und Prozessinformatik
44780 Bochum, Germany
Lunze@atp.rub.de

Prof. Dr. Marcel Staroswiecki
Université Lille I
Ecole Polytechnique Universitaire de Lille
59655 Villeneuve d'Ascq Cedex, France
and
Ecole Normale Supérieure de Cachan
94235 Cachan, France
marcel.staroswiecki@univ-lille1.fr

Library of Congress Control Number: 2006927814

ISBN-10 3-540-35652-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-35652-3 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springer.com

Printed in Germany

Typesetting: Digital data supplied by data
Final processing: PTP-Berlin Protago-TEX-Production GmbH, Berlin (www.ptp-berlin.com)
Cover-Design: estudio Calamar, Frido Steinen-Broo, Spain
Printed on acid-free paper 62/3141/Yu – 5 4 3 2 1 0

Preface

Technological systems are vulnerable to faults. Actuator faults reduce the performance of control systems and may even cause a complete break-down of the system. Erroneous sensor readings are the reason for operating points that are far from the optimal ones. Wear reduces the efficiency and quality of a production line. In many fault situations, the system operation has to be stopped to avoid damage to machinery and humans.

As a consequence, the detection and the handling of faults play an increasing role in modern technology, where many highly automated components interact in a complex way such that a fault in a single component may cause the malfunction of the whole system. Due to the simultaneously increasing economic demands and the numerous ecological and safety requirements to be met, high dependability of technological systems has become a dominant goal in industry.

This book introduces the main ideas of fault diagnosis and fault-tolerant control. It gives a thorough survey of the new methods that have been developed in the recent years and demonstrates them by application examples. To the knowledge of the authors, all major aspects of fault-tolerant control are treated for the first time in a single book from a common viewpoint.

Scope. Whereas fault diagnosis has been the subject of intensive research since the 1970s and there are several good books about this subject, systematic methods for fault handling is a new area of automatic control. The book considers both steps of fault-tolerant control together and shows how the information gained by model-based diagnosis can be used to find remedial actions that adapt the control algorithms to the faulty conditions in order to keep the system in operation. Basically, such actions can be classified as *fault accommodation*, which deals with the autonomous adaptation of the controller parameters to the faulty plant behaviour, and *control reconfiguration*, which includes the selection of a new control configuration and the on-line re-design of the controller.

The solution of these problems necessitates new analysis tasks like the test of the reconfigurability of the system or the search for redundant sensors and actuators,

which can replace faulty components. The aim is to close the control loop after a break-down of such components has brought the controller out of operation. With respect to fault accommodation and control reconfiguration, the book presents the current state of the art.

The fault diagnostic parts of the book describe those methods and ideas which can be used to identify the fault with sufficient detail for fault accomodation or reconfiguration. The detection of a fault alone is not sufficient for fault-tolerant control, but the fault location and, possibly, the fault magnitude have to be known to activate appropriate remedial actions.

The design and implementation of fault-tolerant control necessitates a variety of techniques. The search for redundancies concerning the information and the possible control activities in a system, the selection of a reasonable control configuration, and the combination of diagnostic methods with controller design methods are some of the problems to be tackled. This set of different tasks cannot be dealt with by a single analytical model of the system under consideration, but different viewpoints have to be combined. For this reason, the book introduces several models of dynamical systems and describes how these models can be used in fault-tolerant control. A component-oriented description of the system architecture is used to find the cause-effect chains from the primary faults to the measured fault symptoms. A structural analysis is introduced to elaborate the analytical redundancies that can be used for fault diagnosis and fault-tolerant control actions. For the well known continuous system representations like the state-space model and the transfer function, diagnostic methods and their extensions to fault-tolerant control algorithms are explained. With the presentation of diagnostic and reconfiguration methods for discrete-event systems and quantised systems, the book provides further novel material that has not yet been described in monographs or textbooks.

Structure of the book. The book is organised according to the different models used. As each of these models requires its own mathematical background and the methods based on these models follow different lines of thinking, the book cannot present the methods in all details. The aim is to give the readers a broad view on the field and provide them with bibliographical notes for further reading. A further reason for the different depth with which the chapters tackle the fault-tolerant control problems is given by the current status of research. Whereas for continuous-variable systems, fault diagnostic and fault-tolerant control methods have been developed for long, discrete-event systems became the subject of substantial research with respect to the topic of this book only recently. Hence, this field has not yet reached the same maturity as for continuous systems.

The chapters start with a verbal explanation of the main ideas and illustrate all results by **two running examples** that concern a simple tank system and a ship autopilot. The common use of these examples in all chapters makes a comparison of the alternative approaches very easy. It is the knowledge of the aims, models, ideas and methods used for different problems of fault diagnosis and fault-tolerant control that enables a control engineer to tackle practical problems under the circumstances

given by the particular field of application. To introduce him to this knowledge is the primary aim of this book.

Level of the book. The intended readers of the book are graduate students of control, electrical, mechanical or process engineering with knowledge in control, continuous system theory and filtering. The authors use the text in regular courses at the Université Lille 1, at the Universitè Libre de Bruxelles, and at the universities of Bochum, Lyngby and Trondheim.

In the introductory parts of all chapters the problems to be solved are posed in a framework that is familiar to practising engineers. They describe the new ideas and concepts of fault diagnosis and fault-tolerant control in an intuitive way, before these ideas are brought into a strict mathematical form, which requires a firm systems theoretic background. Interesting practical examples illustrate the applicability of the methods. Bibliographical notes at the end of each chapter point to the origins of the presented ideas and the current research lines. The evaluation of the methods and the application results should help the readers to assess the available methods and the limits of the present knowledge about fault-tolerant control with respect to their particular field of application.

The book is self-contained with a review on some basics in the appendices. Its understanding requires knowledge about dynamical systems and controller design. Many figures illustrate the problems, methods and results in an intuitive way and make the interpretation of the rigorous mathematical treatment easier.

Common research. The idea of the authors to bring together their different views and principles for fault diagnosis and fault-tolerant control originated in the project "Control of Complex Systems (COSY)", which was funded by the European Science Foundation between 1995 and 1999. Some of the material has been collected for a PhD course at Aalborg University (Denmark) held in April 1999. In the later DAMADICS Training and Research Network funded by the European Union beween 2001 and 2004 several chapters of the book have been presented at the summer schools and workshops. The interest of people from industry in this new subject has been attracted in courses organised by the Carl-Cranz-Gesellschaft Oberpfaffenhofen (Germany), where preliminary versions of this book have been used.

The large scope of the book was made possible by the close cooperation and by common research of the four authors together with their PhD students and colleagues. The introductory part (Chapters 1 through 3) and the application examples (Chapter 10) describe ideas and results of these four groups. The presentation of the methods for dealing with the system architecture (Chapter 4) is common work of the groups of MOGENS BLANKE in Aalborg and Lyngby (Denmark) and MARCEL STAROSWIECKI in Lille (France). The part on structural analysis (Chapter 5) introduces the methods developed in Lille. Diagnostic methods for continuous systems have been elaborated by many groups. The presentation of those ideas that can be used in fault-tolerant control (Chapter 6) resulted from common work and teaching

experiences of MOGENS BLANKE, MICHEL KINNAERT (Brussels, Belgium) und MARCEL STAROSWIECKI. Chapter 7 on fault accommodation and control reconfiguration describes ideas of the groups in Lyngby, Bochum, Brussels and Lille. The methods for dealing with discrete-event systems and quantised systems (Chapters 8 and 9) have been elaborated by the group of JAN LUNZE in Hamburg and Bochum (Germany), where particularly JOCHEN SCHRÖDER has made substantial contributions not only to the research work but also to the presentation of the results in this book.

Acknowledgements. The authors express their gratitude to the European Science Foundation and the European Union for financial support of the collaboration of the four groups in the COSY and the DAMADICS projects and to the national science funding organisations (Statens Teknisk-Videnskabelige Forkningsråd, Denmark; Région Wallonne, Belgium; Deutsche Forschungsgemeinschaft, Germany; Centre National de la Recherche Scientifique and Ministère de la Recherche, France) for supporting numerous projects in the field of fault diagnosis and fault-tolerant control during the last years.

Special thanks are due to our former and current PhD students and research associates, particularly to ROOZBEH IZADI-ZAMANABADI, JAKOB STOUSTRUP and JESPER S. THOMSEN (Aalborg), HENRIK NIEMANN and TORSTEN LORENTZEN (Lyngby), JOCHEN SCHRÖDER (Hamburg), THOMAS STEFFEN (Hamburg/Bochum), JÖRG NEIDIG, JAN RICHTER and THORSTEN SCHLAGE (Bochum), and ANNE-LISE GÉHIN and BELKACEM OULD BOUAMAMA (Lille).

Several industrial applications have been accomplished with the interest and help of, among others, CLAUS THYBO (Danfoss) and MARTIN FRITZ (R. Bosch GmbH, Stuttgart). Further, the support from American Power Conversion Denmark A/S to develop tools for structural analysis and from Sauer-Danfoss A/S to develop the fault-tolerant steering-by-wire system introduced in Chapter 10 are gratefully acknowledged.

We are grateful for the valuable help of Ms. KATRIN LUNZE (Stuttgart) for giving the manuscript a uniform layout and of Ms. ANDREA MARSCHALL (Bochum) for drawing many of the figures.

Second edition. This new edition has been extended by a lot of new material, which has appeared in the three-year period after the first edition was finished. Among them is the description of new diagnostic structures like decentralised vs. cooperative diagnosis, and remote diagnosis. The uniform look at continuous and discrete-event systems diagnosis has been deepened. New results on the diagnosis of continuous systems and the reconfiguration of the control loop after sensor or actuator failures have been included. The application examples are extended by a steering-by-wire system and the air path of a diesel engine, both of which include experimental results. The list of references and the bibliographical remarks at the end of all chapters have been up-dated.

The presentation has been improved according to the authors' experience from using this book in their lectures and courses in industry. Many chapters finish with exercises to be used in lectures or for self-repetition of the material.

Lyngby, Brussels, Bochum and Lille M.B., M.K., J.L., M.S.
May 2006

The book homepage at `www.rub.de/atp` → `Books` provides further information such as the description of the COSY benchmark problems of fault-tolerant control, exercises, and files with the figures of this book for their use as slides in lectures.

Contents

Appendices

The authors

Mogens Blanke is professor of automatic control at the Section of Automation at Ørsted·DTU of the Technical University of Denmark and is adjoin professor at the CESOS center of excellence at the Norwegian University of Science and Technology. His research interests comprise autonomous and fault-tolerant systems, fault diagnosis, systems architecture design to obtain desired safety properties, system modelling, identification and control. His experiences result from the development of fault-tolerant design methods for the Danish Ørsted satellite, from the design of several marine automation systems and from the design of fault-tolerant steering-by-wire architectures for vehicles.

Michel Kinnaert is professor in the Department of Control Engineering and System Analysis at the Université Libre de Bruxelles (Belgium). He has held a visiting professor position at the LAGEP at the Université Claude Bernard Lyon 1 and a post-doctoral position at the University of Newcastle (Australia). His research interests include fault diagnosis and fault-tolerant control for linear and nonlinear systems with applications in the process industry and in mechatronics. Professor Kinnaert is currently the chairman of the IFAC Technical Committee SAFEPROCESS.

Jan Lunze is professor of automatic control and head of the Institute of Automation and Computer Control at the Ruhr-Universität Bochum (Germany). His research interests include fault diagnosis and reconfigurable control of discrete-event and hybrid systems, qualitative modelling of dynamical systems, linear control theory with applications in the automotive and process industries, and applications of symbolic information processing to control systems. He is author of two monographs on robust and decentralised control and of several textbooks on control theory, discrete-event systems and artificial intelligence.

Marcel Staroswiecki is professor of automatic control at the Université des Sciences et Technologies de Lille (France). He has been heading the Laboratoire d'Automatique et d'Informatique Industrielle de Lille (LAIL-CNRS) and is currently with the Laboratoire SATIE-CNRS at Ecole Normale Supérieure de Cachan. Professor Staroswiecki has been working on fault detection, isolation and recovery algorithms since 1986. His research group addresses model, signal and data-based approaches to the supervision of complex and embedded systems with emphasis on structural analysis, intelligent instruments and components, and applications in the process industry and to transportation systems.

Chapter 1

Introduction to diagnosis and fault-tolerant control

This chapter introduces the aims, notions, concepts and ideas of fault diagnosis and fault-tolerant control and outlines the contents of the book.

1.1 Technological processes subject to faults

Our modern society depends strongly upon the availability and correct function of complex technological processes. This can be illustrated by numerous examples. Manufacturing systems consist of many different machine tools, robots and transportation systems all of which have to correctly satisfy their purpose in order to ensure an efficient and high-quality production. Economy and every-day life depend on the function of large power distribution networks and transportation systems, where faults in a single component have major effects on the availability and performance of the system as a whole. Mobile communication provides another example where networked components interact so heavily that component faults have far reaching consequences. For automobiles strict legal regulations for protecting the environment claim that the engine has to be supervised and shut off in case of a fault.

In the general sense, a *fault* is something that changes the behaviour of a system such that the system does no longer satisfy its purpose. It may be an internal event in the system, which stops the power supply, breaks an information link, or creates a leakage in a pipe. It may be a change in the environmental conditions that causes an ambient temperature increase that eventually stops a reaction or even destroys the reactor. It may be a wrong control action given by the human operator that brings the system out of the required operation point, or it may be an error in the design of the

system, which remained undetected until the system comes into a certain operation point where this error reduces the performance considerably. In any case, the fault is the primary cause of changes in the system structure or parameters that eventually leads to a degraded system performance or even the loss of the system function.

In large systems, every component has been designed to accomplish a certain function and the overall system works satisfactorily only if all components provide the service they are designed for. Therefore, a fault in a single component usually changes the performance of the overall system.

In order to avoid production deteriorations or damage to machines and humans, faults have to be found as quickly as possible and decisions that stop the propagation of their effects have to be made. These measures should be carried out by the control equipment. Their aim is to make the system *fault tolerant*. If they are successful, the system function is satisfied also after the appearance of a fault, possibly after a short time of degraded performance. The control algorithm adapts to the faulty plant and the overall system satisfies its function again.

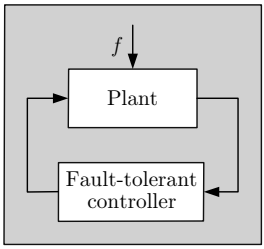

Fig. 1.1. Fault-tolerant system

From a systems-theoretic viewpoint, fault-tolerant control concerns the interaction between a given system (plant) and a controller (Fig. 1.1). The term "controller" is used here in a very general sense. It does not only include the usual feedback or feedforward control law, but also the decision making layer that determines the control configuration. This layer analyses the behaviour of the plant in order to identify faults and changes the control law to hold the closed-loop system in a region of acceptable performance.

Controllers are usually designed for the faultless plant so that the closed loop meets given performance specifications and, hence, satisfies its function. Fault-tolerant control concerns the situation that the plant is subject to some fault f, which prevents the overall system to satisfy its goal in the future. A fault-tolerant controller has the ability to react on the existence of the fault by adjusting its activities to the faulty behaviour of the plant. Hence, for an observer who evaluates the function of the closed-loop system shown in Fig. 1.1, the system is fault-tolerant if it may be subject to some fault, but the fault is not "visible", because the system remains satisfying its designated goal.

Generally, the way to make a system fault-tolerant consists of two steps:

1. **Fault diagnosis:** The existence of faults has to be detected and the faults have to be identified.
2. **Control re-design:** The controller has to be adapted to the faulty situation so that the overall system continues to satisfy its goal.

These steps are not carried out by the usual feedback controller, but by a supervision system that prescribes the control structure and selects the algorithm and parameters of the feedback controller.

Engineers have been using this principle for a long time. Traditional methods for fault diagnosis include limit-checking or spectral analysis of selected signals, which make the detection of specific faults possible. In the case of faults, the controller switches to a redundant component. For example, important elements of an aircraft use this principle with a threefold redundancy.

These means for fault tolerance can only be applied to safety-critical systems. Indeed, for a more general use they are unnecessarily complicated and too expensive for two reasons. First, the traditional methods for fault diagnosis presuppose that for every fault to be detected there is a measurable signal that indicates the existence of the fault by, for example, the violation of a threshold or by changing its spectral properties. In complex systems with many possible faults, such a direct relation between a fault and an associated signal does not exist or it is too expensive to measure all such signals. Second, this kind of fault tolerance is based on *physical redundancy*, where important components are implemented more than once. Industry cannot afford to use such a kind of fault tolerance on a large scale.

The methods described in this book are based on *analytical redundancy*. An explicit mathematical model is used to perform the two steps of fault-tolerant control. The fault is diagnosed by using the information included in the model and in the on-line measurement signals. Then the model is adapted to the faulty situation and the controller is re-designed so that the closed-loop system including the faulty plant satisfies the given specifications. Model-based fault-tolerant control is a cheaper way to enhance the dependability of systems than traditional methods based on physical redundancy.

The aim of the book is to describe the existing methods for model-based fault-tolerant control and to demonstrate their applicability by prototypical practical examples. Fault-tolerant control is a new, rapidly developing field. A lot of interesting ideas have already been elaborated, which are presented here.

1.2 Faults and fault tolerance

1.2.1 Faults

A *fault* in a dynamical system is a deviation of the system structure or the system parameters from the nominal situation. Examples for structural changes are the

blocking of an actuator, the loss of a sensor or the disconnection of a system component. In all these situations, the set of interacting components of the plant or the interface between the plant and the controller are changed by the fault. Parametrical changes are brought about, for example, by wear or damage. All these faults yield deviations of the dynamical input/output (I/O) properties of the plant from the nominal ones and, hence, change the performance of the closed-loop system which further results in a degradation or even a loss of the system function.

System behaviour. For a more detailed analysis of the impact of faults consider the plant in Fig. 1.1 from the viewpoint of the controller. The fault is denoted by f. $\mathcal{F}$ is the set of all faults for which the function of the system should be retained. To simplify the presentation, the faultless case is also included in the fault set $\mathcal{F}$ and denoted by f_0. For the performance of the overall system it is important with which output $y(t)$ the plant reacts if it gets the input $u(t)$. The pair (u, y) is called input/output pair (*I/O pair*) and the set of all possible pairs that may occur for a given plant define the *behaviour* $\mathcal{B}$. Note that for a single-input single-output system u and y denote the functions $u : \mathbb{R} \rightarrow \mathbb{R}$ and $y : \mathbb{R} \rightarrow \mathbb{R}$, which describe the input or output signals rather than the values of these functions for given points in time.

Figure 1.2 gives a graphical interpretation. The behaviour $\mathcal{B}$ is a subset of the space $\mathcal{U} \times \mathcal{Y}$ of all possible combinations of input and output signals. The dot A in the figure represents a specific I/O pair that may occur for the given system whereas $C = (u_C, y_C)$ represents a pair that is not consistent with the system dynamics. That is, for the input u_C the system produces an output $y \neq y_C$.

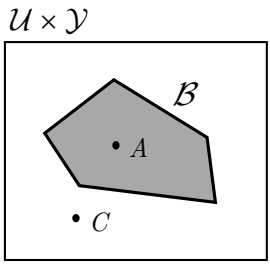

Fig. 1.2. Graphical illustration of the system behaviour

To illustrate the system behaviour in some more detail, consider a static system

$$y(t) = k_s\, u(t), \tag{1.1}$$

where k_s is the static gain. The input and the output are elements of the set $\mathbb{R}$ of real numbers. The set of all I/O pairs is given by

$$\mathcal{B} = \{(u,\, y) \,:\, y = k_s u\},$$

which can be graphically represented as a straight line in the u/y-coordinate system. Equation (1.1) describes, which values of u and y belong together. Faults are found,

if this equation is not satisfied, i.e. if the measured I/O pair $(u,\, y)$ does not belong to the behaviour $\mathcal{B}$ like the pair depicted by the point C in Fig. 1.2.

For a dynamical system the behaviour becomes more involved because the I/O pairs have to include the whole time functions $u(\cdot)$ and $y(\cdot)$ that represent the input and output signals. In a discrete-time setting, the input u is represented by the sequence

$$U = (u(0),\, u(1),\, u(2), ...,\, u(k_h))$$

of input values that occur at the time instances 0, 1,..., k_h, where k_h denotes the time horizon over which the sequence is considered. Often, k_h is the current time instant, until which the input sequence is stored. Likewise, the output is described by the sequence

$$Y = (y(0),\, y(1),\, y(2), ...,\, y(k_h)).$$

Consequently, the signal spaces $\mathbb{R}$ used for the static system have to be replaced by $\mathcal{U} = \mathbb{R}^{k_h}$ and $\mathcal{Y} = \mathbb{R}^{k_h}$ for single-input single-output systems and by signal spaces of higher dimensions if the system has more than one input and one output. Then the behaviour is a subset of the cartesian product $\mathcal{U} \times \mathcal{Y} = \mathbb{R}^{k_h} \times \mathbb{R}^{k_h}$

$$\mathcal{B} \subset \mathbb{R}^{k_h} \times \mathbb{R}^{k_h}$$

(Fig. 1.2). $\mathcal{B}$ includes all sequences U and Y that may occur for the faultless plant. For dynamical systems, the I/O pair is a pair $(U,\, Y)$ of sequences rather than a pair $(u,\, y)$ of current signal values.

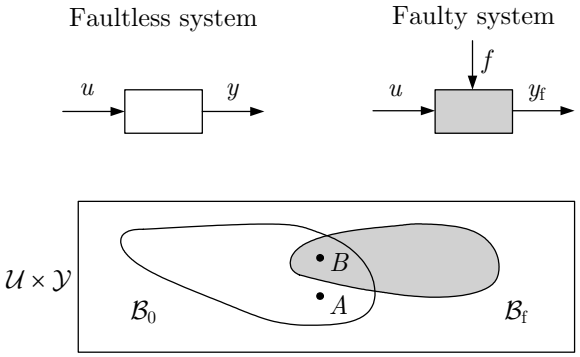

Fig. 1.3. System subject to faults

Fault effects on the system behaviour. A fault changes the system behaviour as illustrated in Fig. 1.3. Instead of the white set, the system behaviour is moved by the fault towards the grey set. If a common input u is applied to the faultless and the faulty system, then both systems answer with the different outputs Y_A or Y_B, respectively. The points $A = (U,\, Y_A)$ and $B = (U,\, Y_B)$ differ and lie in the white or

the grey set, respectively. This change in the system behaviour makes the detection and isolation of the fault possible, unless the faulty I/O pair lies in the intersection of $\mathcal{B}_0$ and $\mathcal{B}_f$.

In the strict sense, the fault is the primary cause of a misfunction. It has to be distinguished from the effects of the fault, which are described by the change of the I/O behaviour. Therefore, fault diagnosis has to trace back the cause-effect relations from the measured I/O pair, which is found to be different from the nominal one, to the primary cause of this change, which is the fault to be identified.

Modelling of faulty systems. For fault-tolerant control, dynamical models have to describe the plant subject to the faults $f \in \mathcal{F}$. These models will play a major role throughout this book. They describe the behaviour of the faultless and the faulty system, i.e. they restrict the possible I/O pairs to those that appear in the behaviour $\mathcal{B}_0$ or $\mathcal{B}_f$ in Fig. 1.3. Therefore, models represent *constraints* on the signals U and Y that appear at the plant. The notion of constraints will be used synonymously with the notion of model equations in this book.

In dependence upon the kind of systems considered, constraints can have the form of algebraic relations, differential or difference equations, automata tables or behavioural relations of automata. A set of such constraints constitutes a model, which can be used as a generator of the system behaviour. For a given input U the model yields the corresponding output Y. If the model is used for a specific fault, it shows how the system output Y is affected by this fault.

In fault diagnosis, the constraints can also be used to check the consistency of measured I/O pairs with the behaviour of the faultless or the faulty system. In this situation, not only the input U, but also the output Y is known and it is checked whether the pair (U, Y) belongs to the behaviour $\mathcal{B}$:

$$(U, Y) \stackrel{?}{\in} \mathcal{B}.$$

Faults versus disturbances and model uncertainties. Like faults, disturbances and model uncertainties change the plant behaviour. In order to explain their distinction, consider a continuous-variable system that is described by an analytical model (e.g. differential equation). For this kind of systems, faults are usually represented as additional external signals or as parameter deviations. In the first case, the faults are called *additive faults*, because in the model the faults are represented by an unknown input that enters the model equation as addend. In the second case, the faults are called *multiplicative faults* because the system parameters depending on the fault size are multiplied with the input or system state.

In principle, disturbances and model uncertainties have similar effects on the system. Disturbances are usually represented by unknown input signals that have to be added up to the system output. Model uncertainties change the model parameters in a similar way as multiplicative faults.

The distinction is given by the aim of fault-tolerant control. The faults are those elements which should be detected and whose effects should be removed by remedial actions. Disturbances and model uncertainties are nuisances, which are known to exist but whose effects on the system performance are handled by appropriate measures like filtering or robust design. Control theory has shown that controllers can be designed so as to attenuate disturbances and tolerate model uncertainties up to a certain size. Faults are more severe changes, whose effects on the plant behaviour cannot be suppressed by a fixed controller. Fault-tolerant control aims at changing the control law so as to cancel the effects of the faults or to attenuate them to an acceptable level.

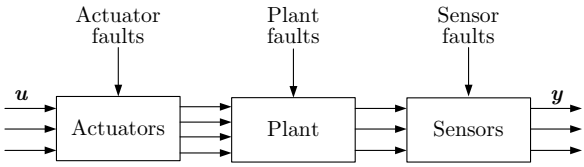

Fig. 1.4. Distinction between actuator faults, plant faults and sensor faults

Classification of faults. The faults are often classified as follows (cf. Fig. 1.4):

- **Plant faults:** Such faults change the dynamical I/O properties of the system.
- **Sensor faults:** The plant properties are not affected, but the sensor readings have substantial errors.
- **Actuator faults:** The plant properties are not affected, but the influence of the controller on the plant is interrupted or modified.

Due to the "location" of sensor and actuator faults at the end or the beginning of the cause-effect-chain of the plant, there are specific methods for detecting them. For example, in Section 8.7, observer schemes for sensor and actuator fault diagnosis will be developed and fault-tolerant control will be treated specifically for these cases.

Faults can be distinguished concerning their size and temporal behaviour. Abrupt faults occur, for example, in a break-down of the power supply whereas steadily increasing faults are brought about by wear, and intermittent faults by an intermitted electrical contact. All these different kinds of faults will be considered in this book, although not all methods are suitable to tackle all kinds of faults.

Fault versus failure. A short note is necessary concerning the distinction of the notions of fault and failure with respect to their current use in the engineering terminology. As explained above, a fault causes a change in the characteristics of a com-

ponent such that the mode of operation or performance of the component is changed in an undesired way. Hence the required specifications on the system performance are no longer met. However, a fault can be "worked around" by fault-tolerant control so that the faulty system remains operational.

In contrast to this, the notion of a *failure* describes the inability of a system or a component to accomplish its function. The system or a component has to be shut off, because the failure is an irrecoverable event. With these notions the idea of fault-tolerant control can be stated as follows:

> Fault-tolerant control has to prevent a fault from causing a failure at the system level.

1.2.2 Requirements and properties of systems subject to faults

As faults may cause substantial damage on machinery and risk for human life, engineers have investigated their appearance and impacts for decades. Different notions like safety, reliability, availability and dependability have been defined and investigated. In this section, the aims of fault-tolerant control is related to these notions, which result from different views on faulty systems.

- **Safety** describes the absence of danger. A safety system is a part of the control equipment that protects a technological system from permanent damage. It enables a controlled shut-down, which brings the technological process into a safe state. To do so, it evaluates the information about critical signals and activates dedicated actuators to stop the process if specified conditions are met. The overall system is then called a *fail-safe system*.

- **Reliability** is the probability that a system accomplishes its intended function for a specified period of time under normal conditions. Reliability studies evaluate the frequency with which the system is faulty, but they cannot say anything about the current fault status. Fault-tolerant control cannot change the reliability of the plant components, but it improves the reliability of the overall system, because with a fault-tolerant controller the overall system remains operational after the appearance of faults.

- **Availability** is the probability of a system to be operational when needed. Contrary to reliability it also depends on the maintenance policies, which are applied to the system components.

- **Dependability** lumps together the three properties of reliability, availability and safety. A dependable system is a fail-safe system with high availability and reliability.

As explained earlier, a fault-tolerant system has the property that faults do not develop into a failure of the closed-loop system. In the strict form, the performance remains the same. Then the system is said to be *fail-operational*. In a reduced form, the system remains in operation after faults have occurred, but the system has degraded performance. Then it is called to be *fail-graceful*.

Safety versus fault tolerance. Due to its importance, the relation between safety and fault tolerance is elaborated now in more detail. Assume that the system performance can be described by the two variables y_1 and y_2. Then Fig. 1.5 shows the different regions that have to be considered.

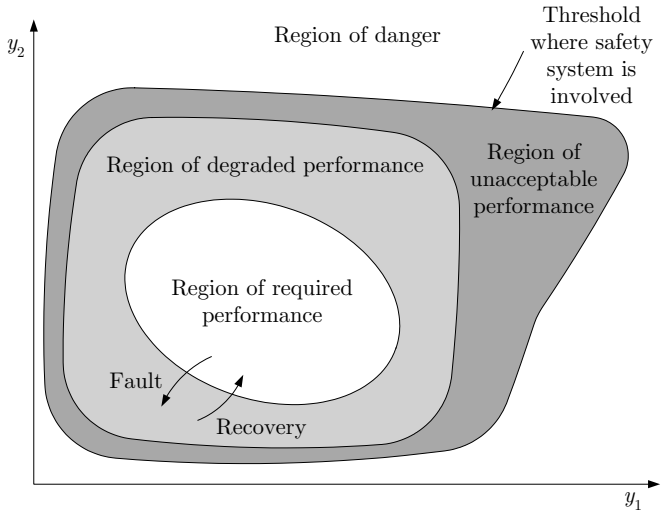

Fig. 1.5. Regions of required and degraded performance

In the region of required performance, the system satisfies its function. This is the region where the system should remain during its time of operation. The controller makes the nominal system remain in this region despite of disturbances and uncertainties of the model used in the controller design. The controller may even hold the system in this region if small faults occur, although this is not its primary goal. In this case, the controller "hides" the effect of faults, which is not its intended purpose but makes the fault diagnostic task more difficult.

The region of degraded performance shows where the faulty system is allowed to remain, although in this region the performance does not satisfy the given requirements but may be considerably degraded. Faults bring the system from the region of the required performance into the region of degraded performance. The fault-tolerant controller should be able to initiate recovery actions that prevent a further degradation of the performance towards the unacceptable or dangerous regions and it should move the system back into the region of required performance. At the bor-

der between the two regions, the supervision system is invoked, which diagnoses the faults and adjusts the controller to the new situation.

The region of unacceptable performance should be avoided by means of fault-tolerant control. This region lies between the region of acceptable performance in which the system should remain and the region of danger, which the system should never reach.

A safety system interrupts the operation of the overall system to avoid danger for the system and its environment. It is invoked if the outer border of the region of unacceptable performance is exceeded. This shows that the safety system and the fault-tolerant controller work in separate regions of the signal space and satisfy complementary aims. In many applications, they represent two separate parts of the control system. For example, in the process industry, safety systems and supervision systems are implemented in separate units. This separation makes it possible to design fault-tolerant controllers without the need to meet safety standards.

1.3 Elements of fault-tolerant control

1.3.1 Structure of fault-tolerant control systems

The architecture of fault-tolerant control is depicted in Fig. 1.6. The two blocks "diagnosis" and "controller re-design" carry out the two steps of fault-tolerant control introduced on page 3.

1. The diagnostic block uses the measured input and output and tests their consistency with the plant model. Its result is a characterisation of the fault with sufficient accuracy for the controller re-design.

2. The re-design block uses the fault information and adjusts the controller to the faulty situation.

Since the term of the controller is used here in a very broad sense, the input u to the plant includes all signals that can be influenced by the control decision units. The aims and methods associated with both blocks will be discussed in more detail below.

In the figure, all simple arrows represent signals. The connection between the controller re-design block and the controller is drawn by a double arrow in order to indicate that this connection represents an information link in a more general sense. The re-design of the controller may not only result in new controller parameters, but also in a new control configuration. Then the old and the new controller differ with respect to the input and output signals that they use (cf. Section 1.3.3).

The figure shows that fault-tolerant control extends the usual feedback controller by a supervisor, which includes the diagnostic and the controller re-design blocks. In the faultless case, the nominal controller attenuates the disturbance d and ensures set-point following and other requirements on the closed-loop system. The main

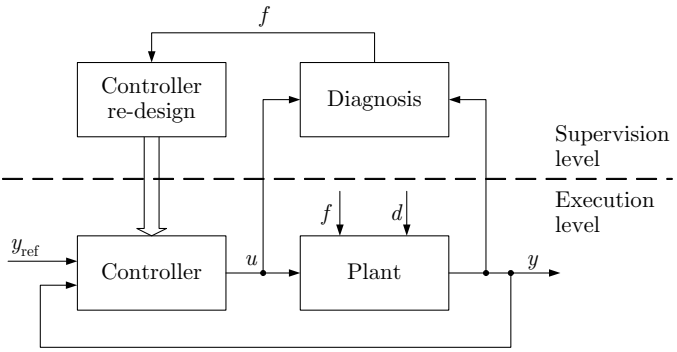

Fig. 1.6. Architecture of fault-tolerant control

control activities occur on the execution level. On the supervision level the diagnostic block simply recognises that the closed-loop system is faultless and no change of the control law is necessary.

If a fault f occurs, the supervision level makes the control loop fault-tolerant. The diagnostic block identifies the fault and the controller re-design block adjusts the controller to the new situation. Afterwards the execution level alone continues to satisfy the control aims.

In Fig. 1.6 as well as in the next figures the diagnostic result f is assumed to be identical to the fault f occurring in the system. This reflects an idealised situation, because in many applications disturbances d or model uncertainties bring about uncertainties of the diagnostic results such that instead of the fault f only an approximate fault $\hat{f}$ or a set $\mathcal{F}$ of fault candidates is obtained. This fact will be investigated in detail in all chapters of this book. Here, however, the idealised situation is considered in order to explain the basic ideas of fault diagnosis and fault-tolerant control.

Established methods for ensuring fault tolerance. To a certain extent, fault tolerance can also be accomplished without the structure given in Fig. 1.6 by means of well established control methods. As this is possible only for a restricted class of faults, these methods will not be dealt with in more detail in this book, but they should be mentioned here.

- **Robust control:** A fixed controller is designed that tolerates changes of the plant dynamics. The controlled system satisfies its goals under all faulty conditions. Fault tolerance is obtained without changing the controller parameters. It is, therefore, called *passive fault tolerance*. However, the theory of robust control has shown that robust controllers exist only for a restricted class of changes of the plant behaviour that may be caused by faults. Further, a robust controller works suboptimal for the nominal plant because its parameters are fixed so as to get a trade-off between performance and robustness.

- **Adaptive control:** The controller parameters are adapted to changes of the plant parameters. If these changes are caused by some fault, adaptive control may provide *active fault tolerance*. However, the theory of adaptive control shows that this principle is particularly efficient only for plants that are described by linear models with slowly varying parameters. These restrictions are usually not met by systems under the influence of faults, which typically have a nonlinear behaviour with sudden parameter changes.

From a structural point of view, adaptive control has a similar structure as fault-tolerant control, if the diagnostic block is replaced by a block that identifies the current plant parameters and the controller re-design block adapts the controller parameters to the identification result (Fig. 1.6). However, in fault-tolerant control the size of the changes of the plant behaviour are larger and not restricted to parameter changes and to continous-variable systems.

If the modifications of the plant dynamics brought about by faults satisfy the requirements that are necessary to apply robust or adaptive control schemes, then these schemes provide reasonable solutions to the fault-tolerant control problem. However, for severe or sudden faults, these methods are not applicable and the ideas presented in this book have to be used.

Fault-tolerant control at the component level and the overall system level. Modern technological systems consist of several, often many subsystems, which are strongly connected. The effect of a fault in a single component propagates through the overall system. In Fig. 1.7 the fault occurring in Component 2 influences all other components.

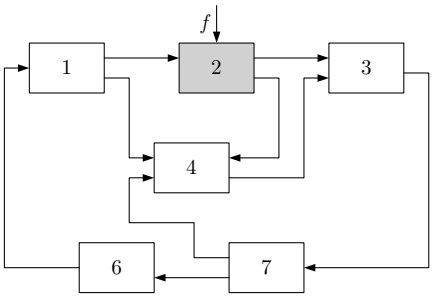

Fig. 1.7. Fault propagation in interconnected systems

The effects of a fault in a single component may be of minor importance to this component. However, due to their propagation throughout the overall system, the fault may eventually initiate the safety system to shut off the whole system. In the terms defined above, the fault has then caused a system failure.

There are two possibilities to stop the propagation of the fault. Either the fault propagation is stopped inside the affected component by making the component fault-tolerant or the propagation of the fault among the components has to be stopped.

The propagation of the fault effects through the overall system usually takes time. This time gives the controller of the affected component the chance to adjust its behaviour to the faulty situation and, hence, to keep the overall system in operation.

1.3.2 Main ideas of fault diagnosis

The first task of fault-tolerant control concerns the detection and identification of existing faults. Figure 1.8 illustrates the diagnostic problem. A dynamical system with input u and output y is subjected to some fault f. The system behaviour depends on the fault $f \in \mathcal{F}$ where the element f_0 of the set $\mathcal{F}$ symbolises the faultless case. The diagnostic system obtains the I/O pair $(U,\, Y)$, which consists of the sequences

$$\begin{aligned} U &= (u(0),\, u(1),\, u(2), ...,\, u(k_h)) \\ Y &= (y(0),\, y(1),\, y(2), ...,\, y(k_h)) \end{aligned}$$

of input and output values measured at discrete time points k within a given time horizon k_h. It has to solve the following problem:

Diagnostic Problem. *For a given I/O pair* $(U,\, Y)$*, find the fault* f.

If the unique result is f_0, the diagnostic system indicates that the system is faultless.

It should be emphasised that the problem considered here concerns on-line diagnosis based on the available measurement data. No inspection of the process is possible. The diagnostic problem has to be solved under real-time constraints by exploitation of the information included in a dynamical model and in the time evolution of the signals. Therefore, the term *process diagnosis* is used if these aspects should be emphasised.

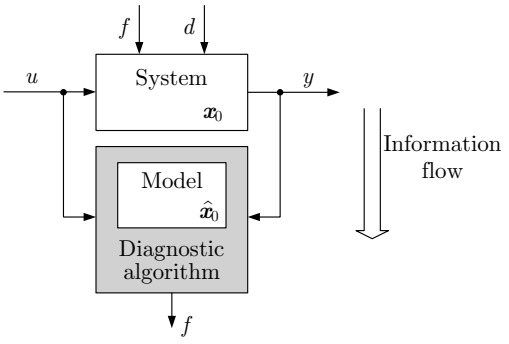

Fig. 1.8. Fault diagnosis

Diagnostic steps. For fault-tolerant control, the location and the magnitude of the fault have to be found. Different names are used to distinguish the diagnostic steps according to their "depth":

- **Fault detection:** Decide whether or not a fault has occurred. This step determines the time at which the system is subject to some fault.
- **Fault isolation:** Find in which component a fault has occurred. This step determines the location of the fault.
- **Fault identification and fault estimation:** Identify the fault and estimate its magnitude. This step determines the kind of fault and its severity.

Consistency-based diagnosis. Different diagnostic methods are explained throughout this book. Although they use different kinds of dynamical models and have different assumptions concerning the measurement information available, they follow a common principle, which can be explained by using the notion of the system behaviour.

In order to be able to detect a fault, the measurement information $(U,\, Y)$ alone is not sufficient, but a reference, which describes the nominal plant behaviour, is necessary. This reference is given by a plant model, which describes the relation between the possible input sequences and output sequences. This model is a representation of the plant behaviour $\mathcal{B}$.

The idea of consistency-based diagnosis should be explained now by means of Fig. 1.2 on page 4. Assume that the current I/O pair $(U,\, Y)$ is represented by point A in the figure. If the system is faultless (and the model is correct) then A lies in the set $\mathcal{B}$. However, if the system is faulty, it generates a different output $\hat{Y}$ for the given input U. If the new I/O pair $(U,\ \hat{Y})$ is represented by point C, which is outside of $\mathcal{B}$ then the fault is detectable. If the faulty system produces the I/O pair represented by point B, no inconsistency occurs despite of the fault. Hence, the fault is not detected.

The principle of *consistency-based diagnosis* is to test whether or not the measurement $(U,\, Y)$ is consistent with the system behaviour. If the I/O pair is checked with respect to the nominal system behaviour, a fault is detected if $(U,\, Y) \notin \mathcal{B}$ holds. If the I/O pair is consistent with the behaviour $\mathcal{B}_f$ of the system subject to the fault f, the fault f may occur. In this case, f is called a *fault candidate*. The diagnostic result is usually a set $\mathcal{F}_c \subseteq \mathcal{F}$ of fault candidates.

To illustrate this result, assume that the system behaviour is known for the faults f_0, f_1 and f_2. The corresponding behaviours $\mathcal{B}_0$, $\mathcal{B}_1$ and $\mathcal{B}_2$ are different, but they usually overlap, i.e. there are I/O pairs that may occur for more than one fault. If the I/O pair is represented by the points A, C or D in Fig. 1.9, the faults found are f_0, f_1 or f_2, respectively. If, however, the measurement sequences are represented by point B, the system may be subjected to one of the faults f_0 or f_1. The diagnostic algorithm cannot distinguish between these faults because the measured I/O pair may occur for both faults. Hence, the ambiguity of the diagnostic result is caused