Fault Diagnosis and Performance Recovery Based on the Dynamic Safety Margin

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Mostafa A. M. Abdel-Geliel Shahin
aus Ägypten

Mannheim, 2006

Dekan: Prof. Dr. M. Krause, Universität Mannheim
Referent: Prof. Dr. E. Badreddin, Universität Mannheim
Koreferent: Prof. Dr. H. P. Geering, ETH Zurich

Tag der mündlichen Prüfung: 20. Dezember 2006

ABSTRACT

The complexity of modern industrial processes makes high dependability an essential demand for reducing production loss, avoiding equipment damage, and increasing human safety. A more dependable system is a system that has the ability to: 1) detect faults as fast as possible; 2) diagnose them accurately; 3) recover the system to the nominal performance as much as possible. Therefore, a robust Fault Detection and Isolation (FDI) and a Fault Tolerant Control (FTC) system design have attained increased attention during the last decades. This thesis focuses on the design of a robust model-based FDI system and a performance recovery controller based on a new performance index called Dynamic Safety Margin (DSM).

The DSM index is used to measure the distance between a predefined safety boundary in the state space and the system state trajectory as it evolves. The DSM concept, its computation methods, and its relationship to the state constraints are addressed. The DSM can be used in different control system applications; some of them are highlighted in this work.

Controller design based on DSM is especially useful for safety-critical systems to maintain a predefined margin of safety during the transient and in the presence of large disturbances. As a result, the application of DSM to controller design and adaptation is discussed in particular for model predictive control (MPC) and PID controller.

Moreover, an FDI scheme based on the analysis of the DSM is proposed. Since it is difficult to isolate different types of faults using a single model, a multi-model approach is employed in this FDI scheme. The proposed FDI scheme is not restricted to a special type of fault.

In some faulty situations, recovering the system performance to the nominal one cannot be fulfilled. As a result, reducing the output performance is necessary in order to increase the system availability. A framework of FTC system is proposed that combines the proposed FDI and the controllers design based on DSM, in particular MPC, with accepted degraded performance in order to generate a reliable FTC system.

The DSM concept and its applications are illustrated using simulation examples. Finally, these applications are implemented in real-time for an experimental two-tank system. The results demonstrate the fruitfulness of the introduced approaches.

ZUSAMMENFASSUNG

Die Komplexität moderner Industrieanlagen macht hohe Verlässlichkeit zu einer notwendigen Anforderung um Produktausfall, Beschädigung der Anlage und Sicherheit zu gewährleisten. Ein verlässliches System kann: 1) Fehler so schnell wie möglich detektieren; 2) Die Ursache des Fehlers genau diagnostizieren; 3) Die Systemleistung so nah wie möglich am Nominalverhalten wiederherstellen. Deswegen wuchs das Interesse an robuster Fehlerdetektion und Isolierung (Fault Detection and Isolation FDI) und fehlertoleranter Regelung (Fault Tolerant Control FTC) in den letzten Jahren erheblich. In dieser Dissertation wird, basierend auf einem neuen Gütekriterium der „Dynamic Safety Margin“ (DSM), der Entwurf eines robusten modellbasierten FDI-Systems und eines Reglers zur Systemwiederherstellung entwickelt.

Das DSM-Gütekriterium wird benutzt um die Entfernung zwischen dem Rand vordefinierten Sicherheitsgebietes im Zustandsraum und der sich entwickelnden Systemtrajektorie zu bewerten. Es werden das DMS-Konzept, seine Berechnung und die Beziehung zu den Zustandsbeschränkungen behandelt. DSM kann für verschiedene regelungstechnische Anwendungen eingesetzt werden. Einige dieser Anwendungen werden in dieser Arbeit vorgestellt.

Ein Reglerentwurf mit Hilfe von DSM ist speziell nützlich für sicherheitskritische Systeme um einen vordefinierten Sicherheitsabstand sowohl während des Transientenverhaltens als auch während großer Störungen einzuhalten. Aus diesem Grund wird die Anwendung des DSM bei Reglerentwurf und Regleranpassung speziell für modellbasierte prädiktive Regelung und PID-Regler betrachtet.

Zusätzlich wird ein FDI-Schema anhand der Analyse des DSMs vorgeschlagen. Da es schwierig ist, verschiedene Fehler unter Verwendung eines einzelnen Modells zu isolieren, wird ein Multi-Modell Ansatz in diesem Schema eingesetzt. Die Anwendung des DSMs um Fehler zu entdecken und zu isolieren verringert die Anzahl der Diagnosevariablen, die der gemessene Zustand oder Ausgangsvektoren der anderen Methoden sind. Dazu ist das vorgeschlagene FDI-Schema nicht auf spezielle Fehlertypen beschränkt.

In einigen fehlerverursachten Situationen kann es unmöglich werden, die Systemleistung vollständig wiederherzustellen. Deswegen muss die Ausgangsleistung verringert werden um die Verfügbarkeit des Systems zu steigern. Die beiden auf dem

DSM basierenden Verfahren zur FDI und FTC, speziell die für den MPC, werden in einem Framework kombiniert um ein zuverlässiges FTC-System mit einer akzeptablen Leistungsminderung zu erhalten.

Das DSM-Konzept und seine Anwendungen werden anhand von Simulationsbeispielen erklärt. Schließlich werden diese Anwendungen in Echtzeit auf einer Zwei-Tank-Laboranlage implementiert. Die Ergebnisse zeigen die Leitungsfähigkeit der eingeführten Ansätze auf.

ACKNOWLEDGMENTS

This work has been carried out at the LS Automation, Computer Engineering department, faculty of mathematics and computer science, university of Mannheim.

I would like to thank Prof. E. Badreddin (Head of LS Automation) for his support, encouragements, fruitful discussion, and guidance during my stay in Mannheim. It is a great honor for me to work under his supervision and to be a member of this research group. I would like to thank also Prof. H. Geering (ETH, Zurich, Switzerland) for accepting to be the co-referee of my work, for his powerful comments and suggestion. It is a great honor that he judges my work. I am thankful for Prof. N. Fliege and Prof. P. Fischer (university of Mannheim) the members of my exam committee beside Prof. Badreddin and Prof. Geering.

Great thanks to the Arab Academy for Science and Technology (AAST) for giving me the chance to study in Germany with a complete financial support.

I am thankful for all people stand beside me and provide direct or indirect support during my work in this thesis. In particular, Dr. A. Gambier (LS Automation) for his support and inspiring discussion; In addition, all staff at LS for creating a positive atmosphere and their co-operation; my colleagues in my working university AAST for their support and encouragements; Eng. Sherine Rady (LS automation) for her help in reviewing the writing of this work.

I would like to express my deepest gratitude and admiration to my parents for their infinite support and unconditional love and engorgements. I wish to tell my father, who has died during my study in Germany, I missed you very much, God's mercy for you.

Furthermore, I would like to thank my wife for her support and encouragements. Despite being in Germany and my family in Egypt most of the time, she took care of my girls and gave me her love.

I wish to express my appreciation to my uncle Mr. Ezzat El-Alfy for his encouragement, aids, and efforts to my family and me during the work. Finally, I am thankful for all my relatives, who always look forward to finish my work, and wish the best for me.

Mannheim, December 2006

Mostafa Abdel-Geliel

CONTENTS

NOMENCLATURE

Some of the terminology used in this thesis is given below. Most of these terminologies were made by the safe process technical committee of IFAC.

Active fault tolerant control systems	Control systems where faults are explicitly detected and accommodated through changing of the control laws
Analytical redundancy	Use of more than one, not necessary identical, way to determine a variable, where one way uses a mathematical process model in analytical form
Availability	Probability that a system or equipment will operate satisfactory and effectively at any point in time
Dependability	Ability of the system to successfully and safely complete its mission
Dependable system	A system that has a high reliability in terms of high availability and where the consequences of a fault are limited to the system it self, i.e. Local faults do not developed into failure at plant level
Disturbance	An unknown and uncontrolled input acting on a system
Error	A deviation between a measured or computed value of an output variable and it's true or theoretically correct one
Failure	A Permanent interruption of a systems ability to perform a required function under a specified operating condition
Failure Modes	The various ways in which failures occur
Fault	An unpermitted deviation of at least one characteristic property or variable of the system from acceptable/normal/ standard condition
Fault Detection	Determination of faults present in a system and time of detection

Fault Diagnosis	Determination of kind, size, location, and time of detection of a fault. Follows fault detection. Includes fault isolation and identification
Fault Identification	Determination of the size and time-variant behavior of a fault. Usually, follows isolation
Fault Isolation	Determination of kind, location, and time of detection of a fault. Follows fault detection. Follows fault detection
Fault Tolerant System	A system where a fault can be accommodated, so that a single fault at subsystem level does not developed into a failure on a system level
Malfunction	An intermittent irregularity in the fulfillment of a system's desired function
Passive Fault Tolerance	A fault tolerant system where faults are not explicitly detected and accommodated, but the controller is designed to be insensitive to a certain set of faults in the system
Quantitative Model	Uses of static and dynamic relations among system variables and parameters in order to describe a system's behavior in quantitative mathematical terms
Reconfiguration	Ability of a system to modify its structure/parameters to account for the detected fault in the system
Reliability	Ability of a system to perform a recurred function under stated conditions, within a given period of time
Residual	A fault indicator, based on a deviation between measurements and model-equation-based computations
Robustness	Ability of a system to maintain satisfactory performance in the presence of parameter variations
Safety	Ability of a system not to cause danger to human operators, equipment or the environment
Symptom	A change of an observable quantity from normal behavior

ABBREVIATION

AFTC	Active Fault Tolerant Control
DSM	Dynamic Safety Margin
EA	Eigenstructure Assignment
EKF	Extended Kalman Filter
ETA	Event Tree Analysis
FDE	Fault Detection and Estimators
FDI	Fault Detection and Isolation
FMEA	Failure Mode Effect Analysis
FTA	Fault Tree Analysis
FTC	Fault Tolerant Control
IMM	Interacting Multiple-Model
LMI	Linear Matrix Inequalities
LP	Linear Programming
LQR	Linear Quadratic Regulator
LQT	Linear Quadratic Tracking
MI	Matrix Inequalities
MIMO	Multi-Input Multi-Output
MM	Multiple Model
MMAE	Multiple Model Adaptive Estimator
MM-FDI	Multiple Model- Fault Detection and Isolation
MPC	Model Predictive Control
mp-QP	multi-Parametric Quadratic Program
PCA	Principle Component Analysis
PFTC	Passive Fault Tolerant Control
QP	Quadratic Programming
SISO	Single-Input Single-Output
UIO	Unknown Input Observer

CHAPTER 1

INTRODUCTION AND PROBLEM STATEMENT

1.1 Background and Motivation

Typical industrial processes are of large and complex nature, involving a huge number of components. The complexity makes systems more vulnerable to faults. A fault changes the behaviour of an industrial process such that the system does no longer satisfy its purpose. It may arise due to component aging and wear, or human errors in connection with installation, operation, and maintenance. It may also arise due to the environmental conditions change that causes, for instance, a temperature increase, which eventually stops a reaction or even destroys the reactor in chemical process. In any case, a fault is the primary cause of changes in the system structure or parameters that leads to a degraded system performance or even the loss of the system function.

In large systems, every component is designed to provide 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.

A fault can be very costly in terms of production loss, equipment damage and human safety. In order to maintain a high level of safety, performance and availability in controlled processes it is important that the system errors, component faults and abnormal system operation are detected promptly, and that the source and severity of each malfunction is diagnosed so that the corrective action can be taken. The human operator can correct some system “errors”, e.g., by closing down the part of the process which has malfunctioned or by re-scheduling the feedback control or the set point parameters. The complexity and fast response required in the system made the manual supervision, to detect a fault, isolate its cause and accommodate the system to a new condition, is hard. Therefore, it is necessary to move the more basic supervision to be automated and become more autonomous.

As a consequence, attention has changed towards increased dependability, a synonym for high degree of availability, reliability, and safety under changing operating

conditions. A more dependable system is the system that has the ability to tolerate faults and prevents them to develop into failures at a subsystem or plant level. Furthermore, it should be guaranteed that all essential faults are detected and all critical faults are accommodated. Hence, modern technological systems rely on sophisticated control functions to meet increased performance requirements.

1.1.1 Reliability and Dependability

The dependability of a system reflects the user's degree of trust in that system. It reflects the extent of the user's confidence that it will operate as users expect and that it will not 'fail' in normal use. For critical systems, it is usually the case that the most important system property is the dependability of the system [1]. Dependability is the ability of the system to successfully and safely complete its mission. In particular, a dependable system implies the ability of the system to:

- Deliver services when requested (Availability).
- Deliver services as specified (Reliability).
- Operate without catastrophic failure (Safety).
- Satisfy mission constraints on performance and time.

Reliability is one of the important properties of a dependable system. Reliability is the probability of failure-free system operation over a specified time in a given environment for a given purpose. Reliability studies evaluate frequency with which the system is faulty, but they cannot say anything about the current fault status [2].

1.1.1.1 Reliability Achievement

The reliability of the system can be achieved by [1], [3]:

- Fault avoidance: Development techniques are used that either minimize the possibility of errors or trap errors before they result in the introduction of system faults.
- Fault detection and removal: Verification and validation techniques that increase the probability of detecting and correcting errors before the system goes into service are used.
- Fault tolerance: Run-time techniques that accommodate the diagnosed faults and prevent them to develop into failure,

- Autonomous supervision and protection: Run-time techniques that reconfigure the system in order to isolate faults.

1.1.2 Safety Critical Systems

Safety is a property of a system that reflects the system's ability to operate, normally or abnormally, without danger of causing human injury or death and without damage to the system's environment [1]. It describes the absence of danger. A safety system is a part of the control equipment that protects a controlled system from permanent damage. It enables a controlled shut-down, which brings the controlled system into a safe state [2].

A critical system is a system that failures can result in significant economic losses, physical damage or threats to human life.

Critical systems can be classified into [1]:

- *Safety-critical system:* A system whose failure may result in injury, loss of life or major environment damage. For example, a control system for a chemical manufacturing plant and nuclear power plant.
- *Mission-critical system:* A system whose failure may result in the failure of some goal-directed activity. For example, a navigational system for a spacecraft.
- *Business-critical system:* A system whose failure may result in the failure of the business using that system. For example, customers account system in a bank.

Safety and reliability are related but distinct. In general, reliability and availability are necessary but not sufficient conditions for system safety.

Reliability is concerned with conformance to a given specification and delivery of service. Whereas safety is concerned with ensuring that the system will not cause damage, irrespective of whether or not it conforms to its specification.

1.1.2.1 Safety Achievement

The safety of system can be achieved by [1]:

- Hazard avoidance: The system is designed so that some classes of hazard simply cannot arise.
- Hazard detection and removal: The system is designed so that hazards are detected and removed before they result in an accident.

- Damage limitation: The system includes protection features, which minimize the damage that may result from an accident.

Reliability and safety analysis can be performed by Fault Tree Analysis (FTA) [5], Failure Mode Effect Analysis (FMEA) [6], Event Tree Analysis (ETA), Cause-Consequence Analysis (CCA), Fault Hazard Analysis (FHA), etc. see for example [3], and [4].

1.1.3 Down-time in the Process Industries

Down time in process industries causes significant economic losses. Moreover, restarting the process takes a long time (hours or days), mainly in critical systems such as petrochemical industries, power plants, etc. Therefore, the availability of the system should be high. Contrarily, the downtime should be reduced. Availability is the probability of a system to be operational and able to deliver the requested services when needed. Contrary to reliability it also depends on the maintenance policies, which are applied to the system components. Figure 1-1 explains the availability and down-time [1], [5].

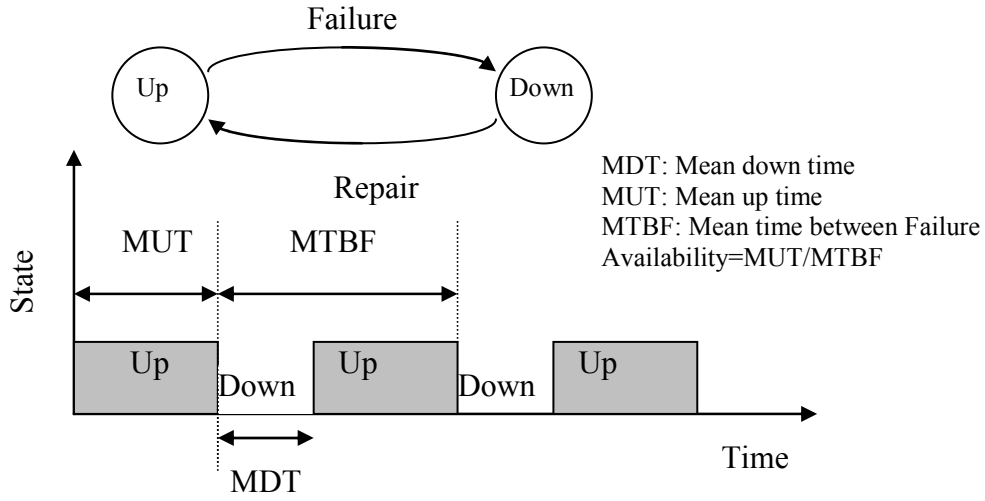


Figure 1-1: Availability and down-time

Here, it can be concluded that early fault detection, accurate fault diagnosis, and fault tolerant capability enhance the overall system safety and availability besides reliability of the monitored system, i.e. enhance the overall system dependability.

1.2 Model Based Fault Detection and Diagnosis

The complexity and sophistication of the new generation of engineered systems, along with growing demands for their reliability, safety and low cost operation, is being met by the use of more automated monitoring and Fault Detection and Isolation (FDI) subsystems. The goal is to accurately isolate problems and restore the system to the nominal operation by making control changes to bring system behavior back to desired operating ranges or at least safe mode of operation. This defines the needs for fault detection, isolation, and recovery.

A fault detection system compares expected behavior of the system with the actual behavior. If the actual behavior deviates from the expected behavior, a symptom is detected and the detection system generates an alarm. The diagnosis system is able to determine the type, size and location of the fault, based on observed analytical symptoms and heuristic symptoms, knowledge of faulty behaviors. This is called fault isolation. Fault diagnosis methods broadly consist of statistical pattern recognition and decision making, such as classification and fuzzy rule-based technique [7].

In general, fault detection methods can be grouped into: (a) model based, (b) knowledge based, and (c) signal based. Further, model-based approaches are typically grouped into quantitative and qualitative models. Quantitative models (differential equations, state space methods, transfer functions, etc.) are used to generally utilize results from the field of the control theory [7]. In qualitative models, the relation between the variables to obtain the expected system behavior is expressed in terms of qualitative functions centered around different units in the process such as causal models and abstraction hierarchy [8], [9]. They are used, in particular, for large and nonlinear systems. The analysis methods used in the qualitative model are FTA, FMEA, ETA, structure analysis, etc. The formal approach uses qualitative reasoning and qualitative modeling [7], [8].

Knowledge-based approaches are based on the use of artificial intelligence methods, neural networks, fuzzy logic, and combination of these methods. These approaches utilize deep understanding of process structure, process unit functions and qualitative models of the process units under various faulty conditions. It is used when it is difficult to obtain a model for the system in case of nonlinear and uncertain systems [10]-[12]. Recent developments in empirical modeling, such as the use of neural networks and fuzzy, have broadened the scope of the quantitative modeling to include ‘data based

model', in additional to the traditional models based on physical principle [13]-[15], [11]. A class of model-free-based FDI approaches has also been developed. Various algorithms have been implemented employing fuzzy logic [16], [17], [10], [11], and artificial neural networks [18]-[20]. In many other techniques, different operating conditions including normal and abnormal ones are treated as patterns. Neural networks are then applied to analyze the online measurement data and map them to a known pattern directly so that the current system condition is identified [18], [21], [13].

Signal processing methods, such as spectral analysis, the wavelet decomposition [22], and Principle Component Analysis (PCA) [23], [24], which do not incorporate any model, can be used for fault detection and diagnosis. Integration of fault detection methods are used to detect system faults in some applications. A combination of self-organized neural network (knowledge base) with wavelet analysis and statistical analysis techniques is used in [25].

There is another classification of FDI in literature, which classifies the FDI methods into only two main categories, model-based and signal-based approaches. Each of which is grouped into quantitative and qualitative methods [9]. In signal-based methods, quantitative methods use signal processing methods, such as spectral analysis, PCA, etc. while qualitative methods use knowledge based method such as fuzzy and neural classification, etc. The signal-based methods, whether quantitative or qualitative, do not incorporate model. The fault detection method, which employs model based on artificial intelligent (knowledge based), is classified under the qualitative model-based FDI methods.

Any of the methods presented above has its own strength and field of application. However, it is widely recognized that in many cases, the design of diagnosis systems for complex plants calls for a wise combination of various techniques, see for example [26] and [27]. The use of Finite State Automata (FSA) to describe a complex industrial plant under diagnosis has been considered in [28]-[30], where the fault observer was derived using the information provided by the sequence of events registered under working conditions. The results of the method in [28] were in agreement with those provided by a standard FMEA, but it has less effort for its developments than FMEA. Fault diagnosis using stochastic FSA is introduced in [31]. A combination of model based with signal processing in fault detection of a hybrid system was introduced in [32].

The block diagram of Figure 1-2 shows the classification of fault detection methods. A comparison of various diagnostic methods based on the desirable characteristics is explained in [9], [33], and [8].

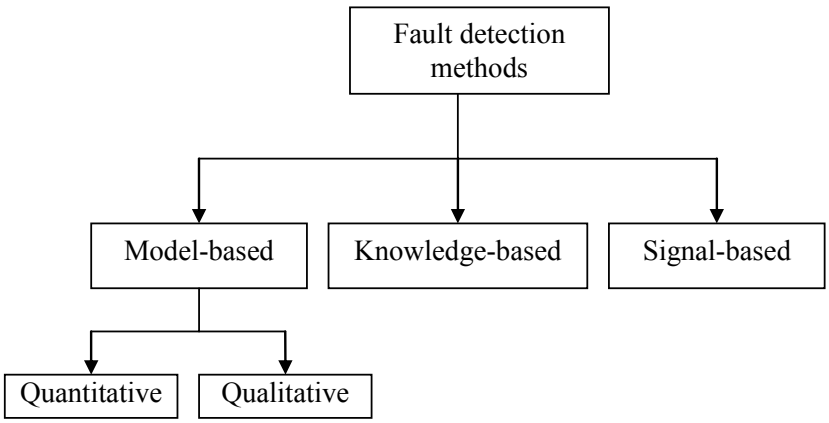


Figure 1-2: Classification of fault detection methods

1.2.1 Model-based Fault Detection Methods

In this section, a more detailed description of analytical model-based fault detection and isolation is introduced. Increasing usage of explicit models in FDI has a large potential due to the following advantages [34]:

- Higher FDI performance can be obtained, for example, more types of faults can be detected and the detection time is shorter.
- FDI can be performed over a large operating range.
- FDI can be performed passively without disturbing the operation of the process.
- Increased possibilities to perform isolation.
- Disturbances can be compensated, i.e. high diagnosis performance can be obtained in spite of presence of disturbances.
- Reliance on hardware redundancy can be reduced, which means that the cost and weight can be reduced.

The disadvantage of model-based FDI is, quite naturally, the need for a reliable model and possibly a more complex design procedure.

The accuracy of the model is usually the major limiting factor of the performance of a model based FDI system. Compared to model-based control, the quality of the model

is much more important in FDI. The reason is that the feedback, used in control, tends to be forgiving with respect to model errors. Diagnosis should be compared to open-loop control since no feedback is involved. All model errors propagate through the diagnosis performance [34].

Model-based methods are normally performed in two steps: residual generation and residual evaluation (decision-making). Residuals are generated by comparing the expected behavior of the system with the measured behavior, where the expected behavior is obtained from a model of the system. Figure 1-3 shows the basic structure of model based fault detection and diagnosis.

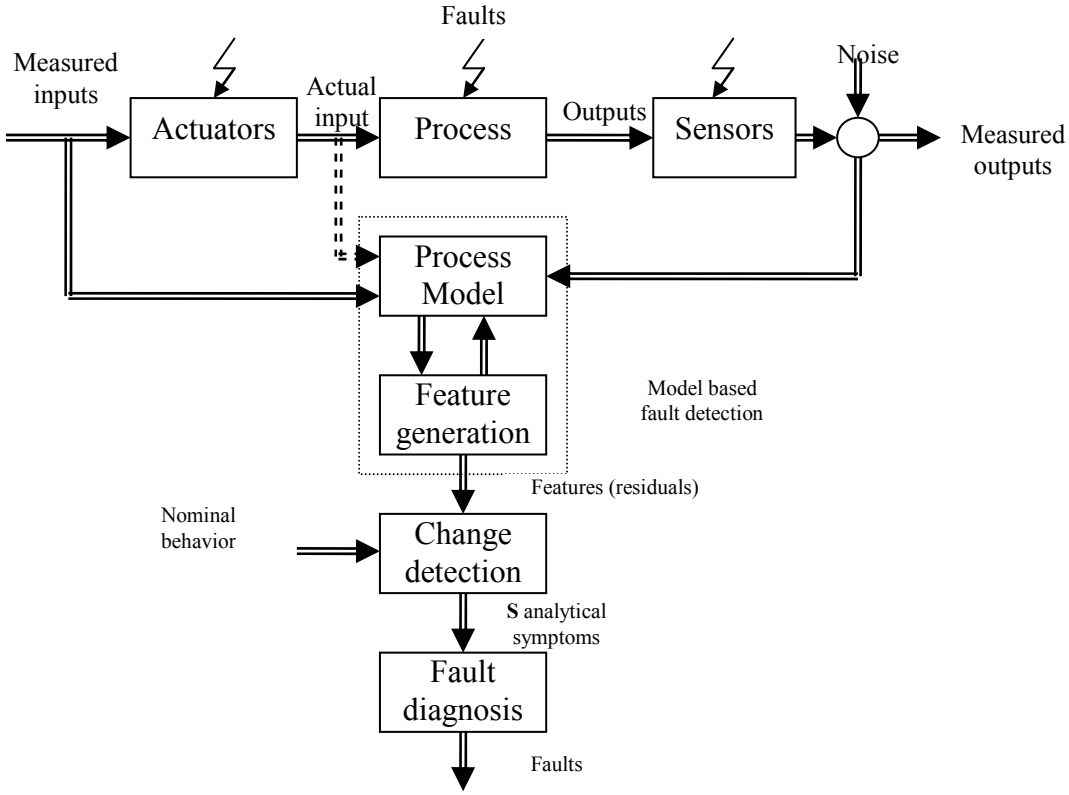


Figure 1-3: General scheme of process model-based fault detection and diagnosis [35]

The selection of model-based FDI method depends on the type of faults and available information of the model. A fault is defined as an unpermitted deviation of at least one characteristic property of a variable from acceptable behavior. Therefore, the fault is a state that may lead to malfunction or failure of the system. The time dependency of faults can be distinguished as abrupt fault (stepwise), incipient fault (drift-like) or intermitted fault. With regard to the process models, the faults can be further classified as additive or multiplicative faults. Additive faults appear, e.g., as offsets of sensors, whereas multiplicative faults are parameter changes within a process [7], [13].

The residual generators of model-based FDI are classified into three main categories; observer-based approaches, parity space approaches, and parameter estimation approaches [7]-[9], [35], [36]. More details about residual generation methods are described in C*hapter 3*. The principle of observer-based approaches is to estimate the system variables (state or outputs) with Luenberger observer for the deterministic case or a Kalman filter for the stochastic case, and use the estimate errors as residuals. The observer based method can be applied if the process parameters are known. Fault modeling is performed with additive faults at the input (additive actuator or process faults) and at the output (sensor offset faults). The design of proper observer gain design has suggested by various methods, such as Eigenstructure assignment [37]-[39], unknown input observer [7], [40], [41], Kronecker canonical form [7], fault sensitive filter [43], and frequency domain optimization approach [44]. Some recent developments in the application of Kalman filter in FDI are found in [45], [46], and [47]. A bank of observer or kalman filters with distinct properties, which is defined as a class of multi-model FDI system, can be used in parallel to isolate faults [7], [48], [13]. Recently, a bank of Extended Kalman Filter (EKF) is used to detect and estimate the faults based on the Multiple Model Adaptive Estimator (MMAE) is presented in [49] and [50]. The number and nature of faults to be detected and isolated necessitate different structures [51]-[53]. Methods of nonlinear observer design are addressed in [54], and [55]. A recent approach to detect and isolate the fault by reconstructing the fault value instead of generating the residuals using observer has been discussed in [56] and the references therein.

In the parity space approaches, using the input-output model of the system, residuals are computed as a difference of the measured outputs and estimated outputs and their associated derivatives. The parity space approach has been developed in frequency domain in [57] and in time domain in [58]. The residual then depends only on the additive input faults and output faults. It is simpler to design and to implement than output observer-based approaches and lead approximately to the same results [35]. The primary residual signals could be reshaped using a transformation matrix to make the residual insensitive to unknown disturbances and to increase fault identification ability; this process is defined as a structure residual generation. A structure residuals generation, based on parity approach in order to obtain good isolation patterns for the residuals, is discussed in [10]. Fault detection in a hybrid system, using structure parity residuals, is discussed in [59], [60]. A lower order parity vector means a simple online

realization but a poorer performance index, while a higher order vector brings a better performance index but leads to higher computational load and a higher rate of misdetection. Therefore, parity space fault detection based on stationary Wavelet Transform (WT) is introduced in [61]. In that contribution, stationery WT is introduced into the residual signal in order to ensure a good performance index of detection, a satisfactory low misdetection rate, and a suitable response speed to faults with low order parity vector and a simple online implementation form. A comparison between parity space approach and a signal base PCA method is discussed in [62].

The concept of parameter estimation methods for FDI is that faults typically affect the physical coefficient of the process. By continuously estimating the parameters of the process model, residuals are computed as the parameters estimation error. To isolate faults successfully, the mapping from the model coefficients to the process parameters must exist and known. Different methods for parameter estimation in FDI have been studied: least squares estimation, output error methods [63], [64], [65], [66], [67], sliding mode estimation [68], neural network estimation [69] and extended Kalman filters [70]. Moving horizon method for detecting and estimating parameter changes is described in [71]. Parameter estimation methods usually need a process input excitation and are especially suitable for the detection of the multiplicative faults. A fault detection using parameter estimation employing fuzzy clustering to diagnosis the fault is addressed in [64] and [65].

Several interesting approaches have been utilized to design and implement FDI algorithms scattered in literature, such as, Linear Matrix Inequality (LMI) approach [72], frequency domain approaches [73], H_2/H_∞ approach [74], and geometric approach for bilinear system [75].

A fault decision is taken, if the residual has changed sufficiently from the nominal behavior. Several decision-making methods have been used, such as binary decision and statistical decision.

1.2.2 Fault Diagnosis Methods

The task of fault diagnosis consists of the determination of the type of fault with as many details as possible such as the fault size, location and time of detection. The diagnostic procedure is based on the observed analytical and heuristic symptoms and the heuristic knowledge of the process, as shown in Figure 1-3. The symptoms may be

presented just as binary values [0,1] or as, e.g., fuzzy sets to consider gradual sizes [35]. The analytical symptoms in the model-based fault detection are the residuals. If the relationship between the residuals and the faults are completely known due to the design of residuals method, then the fault information can be extracted from the residuals directly. For instance, unknown input observer [7] ,[40], fault sensitive filter [43], [50], a bank of observer or kalman filters [7], [48], [50] and a bank of extended Kalman filter to detect and estimate the faults [49], [50] in case of observer fault detection methods, and structure residuals generation based on parity-space approach [10].

The relationship between the symptom and the faults may be unknown or partially known. Therefore, classification and inference methods are used for fault diagnosis [7], [35].

1.2.2.1 Classification Methods

Classification or pattern recognition methods can be used, if no further knowledge is available for the relationships between features (residuals) and faults. The features are determined experimentally for certain faults. The relation between features and faults is therefore learned (or trained) experimentally and stored, forming an explicit knowledge base. Faults can be concluded by comparing of the observed features with the nominal feature.

The classification methods can be grouped as statistical or geometrical classification [7], [35]. A further possibility is the use of neural networks because of their ability to approximate non-linear relations and to determine flexible decision regions for faults in continuous or discrete form [68], [18], [21]. By fuzzy clustering, the use of fuzzy separation areas is possible [64], [65].

1.2.2.2 Inference Methods

Inference methods can be used if the basic relationships between faults and symptoms are at least partially known. This prior knowledge can be represented in causal relations: fault→ events → symptoms. The establishment of these causalities follows the FTA, or the ETA. To perform a diagnosis, this qualitative knowledge can now be expressed in the form of rules: IF <condition> THEN <conclusion>. The condition part contains facts in the form of symptoms as inputs, and the conclusion part includes events and faults as a logical cause of the facts. If several symptoms indicate an event or fault, the facts are associated by AND and OR connections. In this case, the symptoms and events

are considered as binary variables, and the condition part of the rules can be calculated by Boolean equations for parallel serial connection [35], [7]. Because of the continuous natural of the faults and symptoms, this procedure has not proved to be successful. For this reason, approximate reasoning and fuzzy logic are more appropriate for the diagnosis of technical processes, see [35] and the references therein for more details. The use of Transferable Belief Model (TBM) in fault diagnosis and its performance in comparison to Boolean and fuzzy logic approaches are investigated in [76], and [77].

1.2.3 Robustness in Fault Detection System

Usually, the parameters of the system vary with time, and the characteristics of the disturbances and noises are unknown so that they can not be modeled accurately. Since an accurate mathematical model of a physical process is not always available, there is often a mismatch between the actual process and its mathematical model, even if no fault in the process occurs. This constitutes a source of false alarm, which can corrupt the performance of the fault detection and diagnosis system. The effect of modeling uncertainties, disturbances, and noise is therefore the most crucial point in the model-based FDI concept, and the solution to these problems is the key for its practical applicability [78].

To overcome these difficulties, FDI system has to be made robust to such modeling errors and disturbances. In the context of automatic control, the term robustness is used to describe the insensitivity or invariance of the performance of control systems with respect to disturbances, model-plant mismatches or parameter variations. Fault diagnosis schemes, on the other hand, must of course also be robust to the mentioned disturbances, but, in contrast to automatic control systems, they must not be robust to actual faults. On the contrary, while generating robustness to disturbances, the designer must maintain or even enhance the sensitivity of fault diagnosis schemes to faults. The robustness as well as the sensitivity properties must moreover be independent of the particular fault and disturbance mode [7], [13].

An FDI system, which is designed to provide both sensitivity to faults and robustness to modeling errors and disturbances, is called a robust FDI scheme [42]. During the last decades, much FDI research has focused on robust fault diagnosis of uncertain systems. Adaptive threshold can be used to increase the robustness to modeling uncertainties [79]. Surveys of adaptive threshold technique are provided in [37]. One of the most

successful robust FDI approaches is the use of disturbance decoupling principle. This can be done by using unknown input observers [7], [40], [13]. Nevertheless, in some cases such as unstructured uncertainties or structured uncertainties, which does not enter the system as an additive disturbance, perfect decoupling is not possible [80]. An adaptive observer technique for robust FDI with independent effects on the system outputs is introduced in [81]. A game-theoretic approach for robust FDI system is introduced in [82] and [83]. An integrated design approach of FDI in time-frequency based on WT is introduced in [84]. A robust FDI relies on H_∞ filters is suggested in [73], [85]. Recently, FDI for an imprecise model of a system is performed by partitioning the uncertainty space of the imprecise model into smaller subspace models [86]. When new measurements become available, inconsistent subspace models are refuted resulting in a smaller uncertainty space. When all subspace models are refuted, then a fault has been detected. Robust FDI for nonlinear system is discussed in different works, see for example [87] and [88]. Robust FDI problem is defined in details in C*hapter 3*.

1.3 Fault Tolerant Control System and Performance Recovery

The reliability of systems can be increased by insuring that faults will not occur, however, this objective is unrealistic and often unattainable because faults may arise not only due to component aging and wear, but also as human errors in connection with installation and maintenance. In addition, there are some faults that arise due to uncontrollable external effects and sources such as surges, accidences, etc. Therefore, it is necessary to design control systems that are able to tolerate possible faults in systems to improve reliability and availability. This type of control system is often known as Fault Tolerant Control (FTC) systems, which can be classified into two categories: Active Fault Tolerant Control (AFTC) and Passive Fault Tolerant Control (PFTC) [89].

1.3.1 Definition of Fault Tolerant Control System

An FTC system is a control system that can accommodate system component faults and is able to maintain stability and acceptable degree of performance when not only the system is fault-free, but also when there are component malfunctions. FTC system prevents faults in a subsystem from developing into failure at the system level [89].

An FTC system may be called upon to improve system reliability, maintainability, and survivability [90], [91], [2]. The objectives of an FTC system may be different for different applications. An FTC system is said to improve reliability if it allows normal completion of tasks, even after component faults. FTC system could improve maintainability by increasing the time between maintenance actions and allowing the use of simpler repair procedures [89].

Although FTC is a recent research topic in control theory, the idea of controlling a system that deviates from its nominal operating conditions has been investigated by many researchers. The methods for dealing with this problem usually stem from linear quadratic, adaptive, or robust control [92]. The problems to be considered in FTC are quite particular; first, the number of possible faults and consequently action; second, the correct isolation of the faulty components; finally, the accommodation of the system after fault to recover the system to the nominal behavior.

1.3.2 Types of Fault Tolerant Control Systems

The design techniques for FTC system can be classified into two approaches: PFTC system and AFTC system [93], [2]. A particular approach, to be employed, depends on the ability to determine the faults that a system may undergo at the design phase, the behavior of fault-induced changes, and the type of redundancy being utilized in the system. Figure 1-4 shows classification of FTC system approaches.

1.3.2.1 Passive Fault Tolerant Control System

In this approach, a system may tolerate only a limited number of faults, which are assumed to be known prior to the design of the controller. Once the controller is designed, it can compensate for the anticipated faults without any access of on-line fault information. PFTC system treats the faults as if they were sources of modeling uncertainty [93].

PFTC system has a very limited fault tolerance capability. When running on-line, a passive controller is robust only to the presumed faults. Therefore, it is quite risky to rely on PFTC system alone [93]. When redundant hardware components are available, methods of PFTC are also called reliable control methods [94]-[96]. In general, PFTC system has the following characteristics [89]:

- Robust for anticipated faults.

- Utilize hardware redundancy (multiple actuators and sensors, etc.).
- More conservative.

Adaptive controller seems to be the most natural approach to accommodate faults; the faults effects appear as model parameter changes, and they are identified online, and the control law is reconfigured automatically based on new parameters [97],[98]. Robust control methods are used to compensate the effect of the fault in FTC system by assuming the faults as model uncertainties [99], [100].

Designing an output feedback controller as a fault tolerant compensator to stabilize the system, not only during its nominal operating but also in the case of sensors or actuators would fail, have been discussed in [101]. In which, it is concluded that, such compensator always exists, provided that the system is detectable from each output and stablizable from each input.

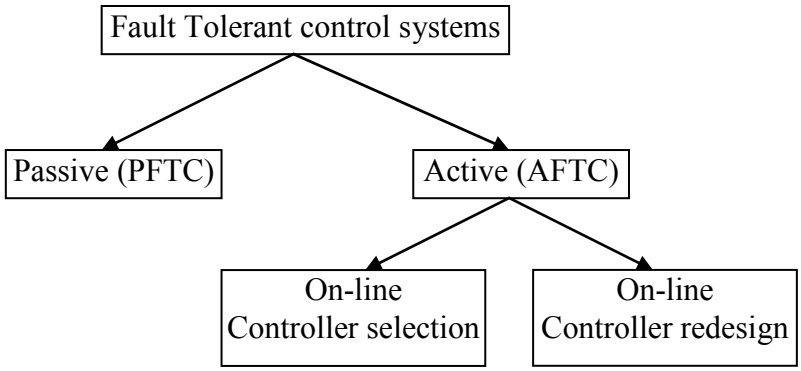


Figure 1-4: Classification of fault tolerant control systems [89]

1.3.2.2 Active Fault Tolerant Control System

In most conventional control systems, controllers are designed for fault-free systems without considering the possibility of fault occurrence. In other case, the system to be controlled may have a limited physical redundancy and it is not possible to increase or change the hardware configuration due to cost or physical restrictions. In these cases, an AFTC system could be designed using the available resources, and employing both physical and analytical system redundancy to accommodate unanticipated faults. Figure 1-5 shows a general schematic diagram of an AFTC system.

An AFTC system compensates for the effects of faults either by selecting a pre-computed control law, or by synthesizing a new control law on-line in real-time. Both approaches need a FDI algorithm to identify the fault-induced changes and to reconfigure the control law on-line [89].

An AFTC system involves significant amount of on-line fault detection, real-time decision making, and controller reconfiguration. It accepts a graceful degradation in overall system performance in the case of faults [2], [102]-[103]. Generally, AFTC system has the following characteristics [89]:

- Employs analytical redundancy in addition to the available hardware redundancy.
- Utilizes FDI algorithm and reconfigurable controller.
- Accepts degraded performance in the presence of a fault.
- Reduces conservationist.

AFTC system is a complex interdisciplinary field that covers a wide range of research areas, such as stochastic systems, applied statistics, risk analysis, reliability, signal processing, control and dynamic modeling [89].

Despite reducing hardware redundancy by using AFTC, the hardware redundancy is mandatory in some of catastrophic failures, which can not be accommodated using only analytical redundancy.

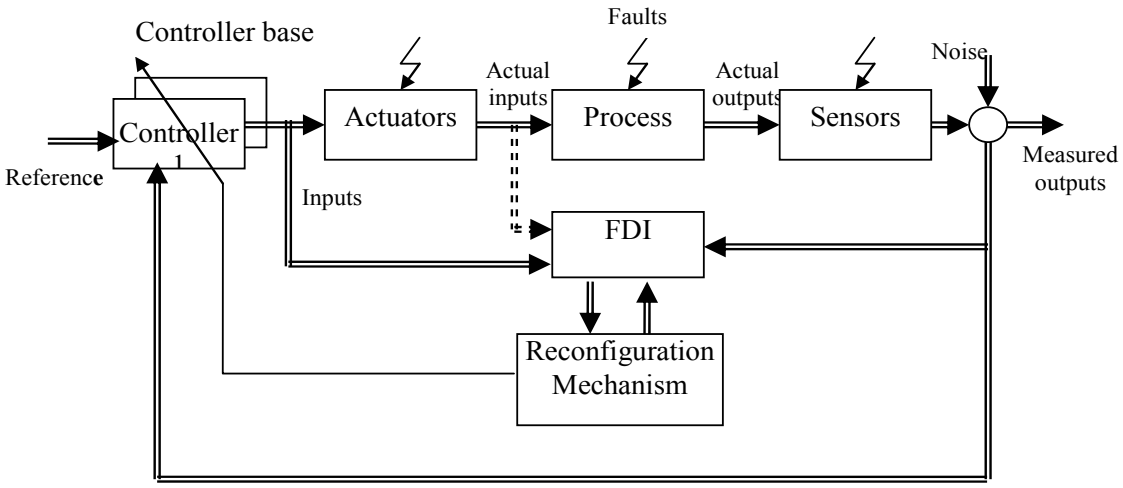


Figure 1-5: Schematic diagram for AFTC system

1.3.3 Control System Reconfiguration

In AFTC system, controller reconfiguration is necessary to compensate for the effects of the failed components. Reconfiguration mechanisms can be classified as on-line controller selection and on-line controller calculation methods [89]. In the first approach, controllers associated with presumed fault conditions are computed a priori in the design phase and selected on-line based on the real-time information from FDI algorithm. In the second approach, controllers are synthesized on-line and in real-time after the occurrence of faults [104].

Control law re-scheduling, multiple models and interacting multiple models approaches are examples of the on-line selection approach, [105]-[107], [108], [50]. This approach is highly dependent on prompt and correct operation of the FDI algorithm. Any false, missed, or error in detection may lead to degraded performance or even to a complete loss of stability of the closed-loop system. Therefore, methods have been proposed to deal with FDI robustness and to design a stability guaranteed AFTC system, see for example [109], [104], and [89].

The pseudo-Inverse method (PIM) is one of the on-line controller design methods. The principle of PIM is to re-compute the controller gain matrix such that the reconfigured system approximates the nominal system in some sense. A severe drawback of this method is that the stability of the reconfigured system is not guaranteed [110]. To overcome this stability problem, a modified PIM method was proposed, in which the difference between the closed-loop matrices is minimized subject to the stability constraints [111].

An Eigenstructure Assignment (EA) based algorithm was proposed in [112]. In this approach, the post-fault eigenvectors are assigned in an optimal way such that performance recovery of the original system is maximized. Extension to integrated FDI and reconfiguration control design using EA algorithm has been developed in [108], [109], and [113].

In [114] an FTC system is designed based on the on-line estimation of an eventual fault and the addition of new control law to the nominal control law, in order to reduce the fault effect once the fault is detected and isolated. The new control law is designed where the closed loop system stability is achieved.

Another on-line reconfiguration method is the model-following approach. In this approach, controller gains are calculated on-line either by enforcing system trajectories

to follow the desired trajectories (explicit model following [115]), or by minimizing a quadratic cost function of the actual and the modeled states (implicit model following [116]). Model Predictive Control (MPC) has been employed in FTC [117]-[119], where an adjustable objective function was optimized based on a simple linear model. Fault tolerant control with re-configuring sliding-mode schemes is discussed in [120].

Feedback controller design for FTC based on Youla parameterization is suggested in [121] and [122].

Control allocation, which manages the distribution of the control law requirements among multiple actuators in some optimal manner in case of actuator fault, for reconfiguration of the controller in particular for flight control application is addressed using constrained linear and quadratic programming in [124], [123], and [50]

Stabilizing of AFTC systems with imperfect fault detection and diagnosis is recently addressed in [104], [89], in which an algorithm that provides a necessary and sufficient condition for exponential stabilization is derived.

AFTC system design schemes with explicit consideration of graceful performance degradation using explicit model-following approach have been proposed in [102]. Recently, an Iterative Learning Observer (ILO) to estimate the state is used to reconfigure the controller in order to compensate the effect of stuck actuator [125].

Feedback linearization is an established on-line reconfiguration technique applied to non-linear system [126]-[127]. Here, an adaptive based on-line controller is modified on-line by the output of parameter estimation algorithm. AFTC has been developed in [128] based on adaptive tracking design that uses neural networks to approximate the unknown fault function for a class of nonlinear system. Recently, an FTC is investigated using an auto-tuning PID controller for nonlinear systems in [129], in which AFTC scheme composing an auto-tuning PID controller based on an adaptive neural network model is proposed. The model is trained on-line using the Extended Kalman Filter (EKF) algorithm.

To overcome difficulties in existing on-line methods, and to integrate the FDI scheme and on-line reconfiguration control law in a coherent manner without any pre-assumption of the knowledge of the post-fault system, several integrate design approaches have been proposed [108], [113]. An on-line reconfiguration method that does not require the use of FDI algorithms is the hybrid adaptive linear quadratic control proposed in [130]. Even though this design method does not need explicit fault information, it has an on-line accommodation capability. Another on-line

reconfiguration based on a model reference control with stabilized recursive least-square algorithm for adaptation is introduced in [131], [91] without explicit FDI.

Recently, designing an FTC unit able to automatically offset the effect of faults, without the need of an explicit FDI process and consequent explicit reconfiguration is discussed in [132]. In [133], stable indirect and direct adaptive controllers are applied to achieve fault tolerant engine control by using Takagi-Sugeno fuzzy systems to "learn" the unknown dynamics caused by faults, and to accommodate faults by updating the controller.

1.4 Problem Statement and Main Contribution

The problem of FDI has drawn increasing attention in a lot of work in the last decades. The disturbance and model uncertainties are the main source of error in the performance of FDI subsystem. For that reason, an FDI system must be insensitive to the model uncertainty and system disturbances with respect to generated features (residuals) and highly sensitive to faults, i.e. robust FDI system. Moreover, the controller should have the capabilities, after fault occurrence, to recover performance close to the nominal desired performance. In addition, it should have the ability to make the system well-behaved in a stable monotonic way during a transient period between the fault occurrence and the performance recovery, which is an important feature to increase system dependability.

1.4.1 Problem Statement

The problem of FDI design and performance recovery can be defined as:

For a system model given in the form of

$$M : \begin{cases} \Delta \mathbf{x}(t) = g(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}, \mathbf{f}, \mathbf{d}, \mathbf{v}) \\ \mathbf{y}(t) = h(\boldsymbol{\theta}, \mathbf{x}, \mathbf{u}, \mathbf{f}, \mathbf{d}, \mathbf{v}) \end{cases} \tag{1.1}$$

where $\mathbf{x} \in \Re^n$ is the state vector of the system model, $\mathbf{u} \in \Re^m$ is the input vector, $\mathbf{y} \in \Re^p$ is the output vector, $\mathbf{f} \in \Re^l$ is the unknown additive fault signal vector, $\mathbf{d}$ is the unknown disturbance, ν is the system noise, Δ is the time derivative operator in continuous system and shift operator in discreet one, g: $\Re^n \times \Re^m \times \Re^l \to \Re^n$, h: $\Re^n \times \Re^m \times \Re^l \to \Re^p$, $\boldsymbol{\theta} \in \Theta$ system parameters and Θ the set of system parameters in faulty and fault-free cases.

It is required to first, develop a robust FDI method that can be used for early detection and isolation of faults; second, design a fault-tolerant control system such that the impact of the fault is minimized, and the system dependability (safety, reliability, availability) is increased.

1.4.2 Main Contributions

A new performance index for the control system design, which is called "Dynamic Safety Margin" (DSM), is introduced in [134]. This index measures how far the system state trajectory is from a predefined safety boundary in the state space at any instance and answers the following questions: Does the system operate in a safe mode all the time even during the transient phase? If so, how far is the current state from a predefined safety boundary? Hence, the DSM value can be taken as a measure for the quality of the controller in this respect. As a result, the main contributions in this thesis concentrate on the DSM concept and its applications.

1.4.2.1 DSM in Contrast to State Constraints

In fault-free situation, the system state remains inside a closed region during the time of operation. This region is defined as a safe operation region. The instantaneous variation of the system state with respect to the safe operation region boundary is indicated by DSM. Therefore, the concept and the computation methods of DSM are discussed in [134] and [136]. An important question might come in mind; what is the difference between safe region boundary and individual state limits (constraints)? Operating the system within state limits does not always mean that the system is fault-free. It is necessary to distinguish between safety boundary, which is used to calculate DSM, and individual state limits. Therefore, the relation between DSM and state constraints are investigated in *Chapter 2* and [136].

1.4.2.2 Relation to Dependability

The DSM index indicates the system mode of operation, whether it is safe or not. Moreover, its value explains how far the system state is away from the safe mode. Therefore, in addition to using DSM as a quality measure to compare between different controllers performance, it can be used as a measure of dependability. Since the dependability analysis depends mainly on statistical models, it cannot reflect the system dynamics. On the other side, the DSM reflects the system dynamics. This is one of the main advantages of

using DSM as a dependability measure. Implementing DSM in different types of controller design is also discussed in [134]. It is concluded that controller design based on DSM permits to maintain a predefined margin of safety during transient and steady state of safety-critical systems. Since the system failure occurs mostly during the transient phase, designing a controller based on DSM to maintain a predefined margin of safety during transient period is a formidable task. Moreover, it can help speeding up performance recovery in some faults, which increases the system dependability [134]-[135].

1.4.2.3 Applications of DSM in Fault-Detection and Performance Recovery

A robust FDI method, based on the analysis of DSM instead of traditional residuals, is introduced in [135], [140], and [141]. One of advantages of dealing with DSM in FDI is that DSM value can be considered as a reduction of data, i.e. measured state variables or subset of them are transformed or projected to a single quantity (DSM).

Considering DSM in controller design is discussed in more details in [139]. In which, two controllers, PID and MPC, design and adapting based on DSM is addressed. DSM is taken as a performance index to adapt the PID controller parameters. Due to the advantage of MPC to deal with system constraints (state and input), DSM is considered as constraint in MPC design. The solution of MPC based on DSM is deduced. Moreover, the feasibility problem of MPC based on DSM is addressed.

An FTC scheme based on DSM is proposed in [138] and [139], in order to recover the system performance during the faulty period. The suggested FTC based on DSM is suitable to be applied in either AFTC or PFTC, according to the available fault information.

1.4.2.4 Practical Implementations and Experiments

The fruitfulness of DSM design and its applications in controller design, robust FDI, and FTC are demonstrated through several real-time experiments in C*hapter 5*. The experimental setup uses standard industrial components, which introduce more realism and robustness into the experiments.

1.5 Outline of the Thesis

The summaries of the different chapters, given below, indicate the scope of the thesis. The thesis consists of six chapters and the main contributions are in *Chapter 2*, *3*, and *4*.

The chapters are devoted to a dynamic safety margin definition and application, robust FDI system, and FTC. They are organized as follow:

Chapter 2 defines the DSM index, and explains the difference between state constraints and DSM. DSM computation methods are discussed as well. Moreover, the different applications of DSM especially in controller design and adaptation is highlighted. Using DSM in first, switching between pre-designed controllers; second, optimal control design as soft constraint; finally, adapting PID controller are tested in illustrating examples, in order to maintain a predefined margin of safety during transient period, steady state period, and in case of disturbance or fault.

Chapter 3 demonstrates the problem of robust FDI system. A robust FDI scheme based on DSM is introduced. The advantage of using DSM in robust FDI, based on multi-model fault isolation scheme, is also discussed. An illustration example is introduced to show the applicability of the proposed FDI scheme.

Chapter 4 discusses the application of DSM in controller design and adaptation, especially PID controller for SISO systems and MPC in case of MIMO systems. The method of adapting PID controller parameters based on DSM is deduced and tested on an illustration example. The solution of MPC based on DSM is discussed, and the adapting algorithm in order to find a feasible is introduced as well. Moreover, a general framework for FTC system based on DSM is introduced.

Chapter 5 illustrates the practical application of DSM in controller design (PID and MPC), FDI, and FTC for an experimental setup. Different types of controller design based on DSM are tested. Different types of faults such as actuator, sensor and internal faults are tested to indicate the applicability of the proposed FDI scheme. The proposed FTC scheme is tested for actuator fault considering AFTC and PFTC design. The practical results demonstrate the usefulness of DSM and its application.

Chapter 6 concludes the work in this thesis, in addition to some suggestions for possible future work as an extension of this work. It illustrates the reason and benefits of using DSM in control system in particular, FDI and FTC system design in order to enhance the overall system dependability. It is usual to find restriction conditions and disadvantages for applying a new approach. For that reason, the restrictions of the proposed approaches are discussed. Finally, open topics related to the analysis and application of DSM are highlighted.

CHAPTER 2

DYNAMIC SAFETY MARGIN DEFINITION AND PRINCIPLES

2.1 Introduction

The main goal of control system design is to achieve a desired performance of the controlled system, which can be specified e.g. according to the stability, rise and settling times or a general norm of the controlled variable. The evaluation of the control system depends mainly on a comparison between the desired performance and the actual performance. The selection of a controller also depends on the available information (quantitative or qualitative) about the controlled system. A quantitative controller is based on the accurate model of the system (model-based), while the qualitative controller depends on the information of the system behavior (knowledge-based) in case that a system model is not available or it is difficult to obtain [142].

Physical constraints exist in many control problems in industry. These constraints can be on inputs, due to actuator limitation, as well as on outputs and some intermediate variables, and can be due to safety limitations, product quality requirements, and efficiency consideration. For example, pressure in a chemical reactor must not be higher than some limits; movements of a robot arm may have been restricted in a certain region of space, and so on. Therefore, the system variables should satisfy the system constraints in order to maintain safe operation.

In this chapter, a new performance index for the control system design is proposed, which is called “Dynamic Safety Margin” (DSM) [134]. This index can also be considered as an additional term in a more general cost functional. This index measures the instantnous distance between the state trajectory and the boundry of a predefined safe operation region in state space. The sign of this index is used to indicate wether the sytem operates in the safe mode or not even if durng the transient phase. As a result, it measures how far the current state is from the predefined safety boundary. Hence, determining DSM can be taken as a measure for the quality of the controller in this respect.

Designing a controller based on DSM is important to maintain a predefined margin of safety during transient and disturbance actions. Moreover, it can help speeding up performance recovery in some cases of system faults. Here are some of DSM applications that will be discussed in this chapter.

2.2 Dynamic Safety Margin

Briefly to explain the idea, let X be the state space in $\Re^n$ and consider that a subspace $\Phi \subseteq X$, which defines the safe operation region for some crucial state variables $\mathbf{x} \in \Re^m$ in the state subspace Φ and $m \leq n$, can be specified by an inequality "$\phi(\mathbf{x}) \leq 0$" while $\phi(\mathbf{x}) > 0$ indicates unsafe operation (Figure 2-1)[1], where $\phi: \Re^m \rightarrow \Re$. It will be further assumed that the system is stable -in the sense of Lyapunov- with the safe region fully contained in the stability region. Starting with the initial condition $\mathbf{x}_o$, the system trajectory will evolve to the operating point $\mathbf{x}_s$ traversing the state space with varying distance to the safety boundary. DSM, in this case, is defined as the shortest distance, $\delta(t)$, between the system state of interest and a predefined boundary $\phi(\mathbf{x})=0$ in this subspace of the state variables. At the operating point $d\delta(t)/dt=0$ and $\delta(.)$ reaches a constant value, δss, indicating the Stationary Safety Margin (SSM). Most industrial designs are made to satisfy SSM of specified values. Figure 2-2 shows the idea of DSM for a system described by two state variable x_1 and x_2.

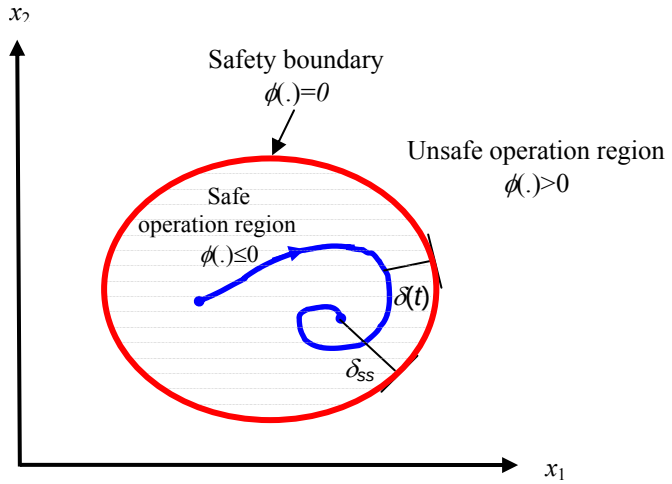


Figure 2-1: DSM definition

[1] Figure 2-1 explains the idea of DSM for a system described by two state variable x_1 and x_2. Safe operation means that there is no fault or large disturbance.

Most of the time the variables are dependent on one another and none of them adequately defines the system safety by itself. Thus, it is necessary to distinguish between safety boundary and individual state limits. Sometimes, some of the safety boundaries are defined by the state limits. Figure 2-2 shows the difference between variable limits and safety boundary. It is clear from the figure that all state variables within thier amplitude limits, but some state vectors, for instance $\mathbf{x}_o$, do not satisfy safety boundary constraints.

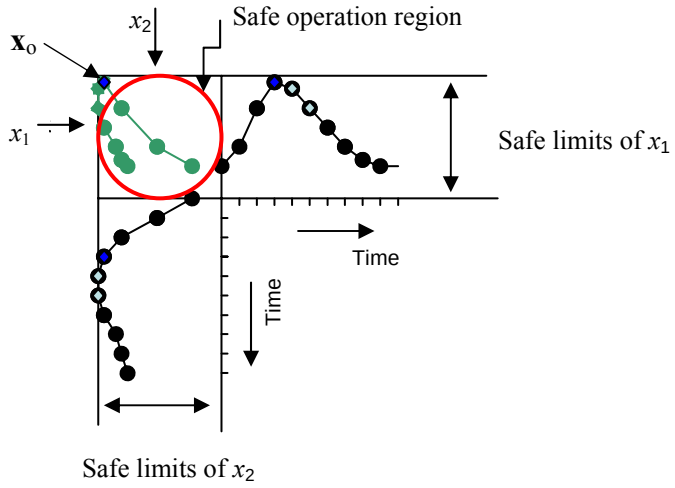


Figure 2-2: DSM and state limits

The boundary of the safe region is determined according to the available experience about the process operation and safety limitation. The system should remain during time of operation inside this region, which implies that the controller should make the nominal system remains in this region despite the existence of disturbance and uncertainties of the model used in the controller design. DSM is called dynamic, because the magnitude of DSM varies with time as the system trajectory evolves in the state space.

In general, the safe-operation region $\Phi \subseteq X$ is defined by a set of inequalities

$$\Phi = \left\{ \phi_i(\mathbf{x}) \le 0 \middle| i = 1,...,q \right\}, \tag{2.1}$$

in addition, the subspace $\mathbf{V} = \left\{ \mathbf{v} \middle| \phi_i(\mathbf{v}) = 0; i = 1,...,q \right\} \subset \Phi$, $\mathbf{v} \in \Re^m$, determines the boundary state of Φ. Therefore, DSM is given by

$$\delta(t) = s(t) \cdot \left\| \mathbf{v} - \mathbf{x} \right\|_{\min} \tag{2.2}$$

where $s(t) = \begin{cases} 1 & \text{if } \mathbf{x} \textbf{ inside} \text{ the safe operation region} \\ -1 & \text{if } \mathbf{x} \textbf{ outside} \text{ the safe operation region} \end{cases}$

$\|\cdot\|_{\min} \triangleq$ shortest distance from $\mathbf{x}(t)$ to ϕ, q is the number of defined inequalities and m is the number of state variables relevant to safety.

2.2.1 DSM Computation

The boundary constraints of the safe region can be defined by a set of either piece-wise linear or nonlinear functions. Therefore, the distance between the state vector and the safety boundaries, in general, can be defined as the solution of the optimization problem

$$\min_{\mathbf{v}} \|(\mathbf{x}-\mathbf{v})\|_2^2 \tag{2.3}$$

subject to

$$\mathbf{v} \in \{\mathbf{v} | \phi_i(\mathbf{v}) = 0; i = 1,...,q\} \tag{2.4}$$

where $\mathbf{x}$ is the current state, and $(\mathbf{x}\text{-}\mathbf{v})$ is the distance vector between $\mathbf{x}$ and $\mathbf{v}$.

The solution of the optimization problem is the state vector $\mathbf{v}_o$. where

$$\mathbf{v}_o = \arg\left(\min_{\mathbf{v}} \|(\mathbf{x}-\mathbf{v})\|_2^2\right)$$

Therefore, the minimum distance between $\mathbf{x}$ and safety boundaries ($\{\phi_i=0\}$) is given by

$$|\delta| = \|(\mathbf{x}-\mathbf{v}_o)\|_2 \tag{2.5}$$

2.2.1.1 DSM computation for safety region defined by linear boundaries

In many cases, the safe operation region can be defined by a set of linear inequalities $\{\phi_i = 0\}$. Furthermore, if the boundary function ϕ_i is nonlinear, it can be subdivided into two or more linear constraints (piecewise linear approximation).

The distance between a linear safety boundary equation and a certain state vector x in state space can be computed in different ways, for example linear algebra, vector algebra, etc., besides the optimization method described before. Linear algebra is more general and easier than an optimization method to obtain the solution. Therefore, the solution using linear algebra is deduced in this section. The vector algebra solution and the optimization method are proved in *Appendix* A as well, to insure the results.

Let the number of state variables of interest be all state variables (m=n) in order to generalize the algorithm. If the safe region is defined by q linear inequalities in the form of

$$\phi(\mathbf{x}) = \mathbf{a}_j^{\mathrm{T}}\mathbf{v}_i - c_i \le 0;\ \ i=1,2,\ldots,q \tag{2.6}$$

then the boundary equations can be written in the form of

$$\phi(\mathbf{v}_i) = \mathbf{a}_j^{\mathrm{T}}\mathbf{v}_i - c_i = 0 \tag{2.7}$$

where $\mathbf{a}_i \in \Re^n$ is a constant vector and $\mathbf{v}_i \in \mathrm{V}_i = \{\mathbf{v}_i |\ \mathbf{a}_i^{\mathrm{T}} .\ \mathbf{x}_i = \mathrm{c}_i\} \subset \Re^n$. Therefore, for any state vector $\mathbf{x}$, the following equation is valid

$$\mathbf{a}_j^{\mathrm{T}}(\mathbf{v}_i - \mathbf{x}) = c_i - \mathbf{a}_i^{\mathrm{T}}.\mathbf{x} \tag{2.8}$$

By taking the absolute value of both side of (2.8), it follows

$$\left|\mathbf{a}_i^{\mathrm{T}}(\mathbf{v}_i - \mathbf{x})\right| = \left|c_i - \mathbf{a}_i^{\mathrm{T}}.\mathbf{x}\right| \tag{2.9}$$

According to Cauchy-Schwarz inequality theorem [143]

$$\left|\mathbf{a}_i^{\mathrm{T}}(\mathbf{v}_i - \mathbf{x})\right| \le \|\mathbf{a}_i\|_2 \|(\mathbf{v}_i - \mathbf{x})\|_2 \tag{2.10}$$

then

$$\|(\mathbf{v}_i - \mathbf{x})\|_2 \ge \frac{\left|c_i - \mathbf{a}_i^{\mathrm{T}}.\mathbf{x}\right|}{\|\mathbf{a}_i\|_2}$$

where $\|(\mathbf{v}_i - \mathbf{x})\|_2$ is the distance between $\mathbf{x}$ and any state vector $\mathbf{v}_i \in \mathrm{V}_i$. Therefore, the minimum distance, the distance between $\mathbf{x}$ and the projection of $\mathbf{x}$ on $\phi_i(.)$, will be

$$|\delta_i| = \min_{\mathbf{x}_i}(\|(\mathbf{v}_i - \mathbf{x})\|_2) = \frac{\left|c_i - \mathbf{a}_i^{\mathrm{T}}.\mathbf{x}\right|}{\|\mathbf{a}_i\|_2} \tag{2.11}$$

Hence, in general if $\mathbf{x}(t)$ is the system state vector at time t then

$$\delta_i(t) = \frac{c_i - \mathbf{a}_i^{\mathrm{T}}.\mathbf{x}(t)}{\|\mathbf{a}_i\|_2} \begin{cases} \ge 0 & \text{iff } \phi_i(\mathbf{x}) < 0 \\ < 0 & \text{iff } \phi_i(\mathbf{x}) > 0 \end{cases} \tag{2.12}$$

The result of (2.12) is the same result which is obtained in *Appendix* A.

The distance vector for all boundaries $\mathbf{d}(t)=[\delta_1(t), \delta_2(t),\ldots, \delta_q(t)]^{\mathrm{T}}$ can be calculated from

$$\begin{aligned}\mathbf{d}(t) &= \mathbf{D}_{ia}\left(\mathbf{c}_c - \mathbf{A}_c\mathbf{x}(t)\right)\\ &= \mathbf{d}_c - \mathbf{D}_a\mathbf{x}(t)\end{aligned} \tag{2.13}$$

where $\mathbf{d}(.)\in\Re^q$, $\mathbf{c}_c\in\Re^q$,$\mathbf{d}_c \in \Re^q$, $\mathbf{D}_a \in \Re^{q\times n}$ and $\mathbf{D}_{ia} \in \Re^{q\times q}$

$$\mathbf{D}_{ia} = \begin{bmatrix} \frac{1}{\|\mathbf{a}_1\|_2} & 0 & 0\cdots\cdots 0 \\ 0 & \frac{1}{\|\mathbf{a}_2\|_2} & 0\cdots\cdots \\ \vdots & & \\ 0\cdots\cdots\cdots\cdots & & \frac{1}{\|\mathbf{a}_q\|_2} \end{bmatrix}, \mathbf{c}_c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ \vdots \\ \\ c_q \end{bmatrix}, \mathbf{A}_c = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \vdots \\ \\ \mathbf{a}_q^T \end{bmatrix}$$

$$\mathbf{d}_c = \mathbf{D}_{ia}\mathbf{c}_c \text{ , and } \mathbf{D}_a = \mathbf{D}_{ia}\mathbf{A}_c$$

Definition 2.1: If Φ is convex and the boundary constrains are linear, then the safe region is a polytope [144], [145].

Theorem 2.1: If Φ is a polytope, there are three possibilities of the component values of $\mathbf{d}$, δ_i, according to the current state position with respect to the safety boundaries:

1. All positive, i.e. $\mathbf{x} \in \Phi$. Then $\delta(.)$, DSM, is the minimum element in $\mathbf{d}(.)$ i.e.

$$\delta(t) = \min_{1\le i\le q} \delta_i(t) \tag{2.14}$$

2. Only one negative i.e. $\mathbf{x} \notin \Phi$ and only one constraint of the safe boundary is violated. Then, $\delta(.)$ is negative and can be calculated from (2.14), which is equal to the component of $\mathbf{d}$ corresponding to the violated constraint.
3. Two or more are negative, i.e. more that one constraint is violated. In this case, the minimum distance, from the state vector to the intersection of violated constraints (vertex of polytope between the violated constraints), should be compared with $\mathbf{d}$, i.e.

$$\begin{aligned}\delta(t) &= \min_{\substack{l,j \\ i\ne j}} \left\{\min(\delta_{lj}, \delta_l, \delta_j)\right\}\\ \delta_{lj} &= \min\left\|\mathbf{v}_{lj} - \mathbf{x}\right\|_2\end{aligned} \tag{2.15}$$

where $(l,j)\in\{\text{index of violated constraints}\}$, δ_l and δ_j are the distances to violated constraints number l and j respectively, δ_{lj} is the distance to the intersection of the

violated constraints l and j, $\mathbf{v}_{lj} \in \mathbf{V}_{ij}=\{\mathbf{x}_{lj} | \ \phi_i(\mathbf{x}_{lj})=0 \vee \phi_j(\mathbf{x}_{lj})=0 \}$ is the intersection between the two boundaries l and j (vertex).

Proof: Figure 2-3 describes the different possible situations of the state vector **x** with respect to the convex safe region.

In Figure 2-3, δ_{u1} and δ_{u2} are the minimum distances from the violated constraints $\phi_1(.)$ and $\phi_2(.)$ respectively.

Note that, $|\delta_a| \geq |\delta_{u1}|$ and $|\delta_a| \geq |\delta_{u2}|$ where δ_a is the actual minimum distance to the safe region (DSM), which is the distance between the current state and the vertex between $\phi_1(.)$ and $\phi_2(.)$ ($\mathbf{v}_{12}$).

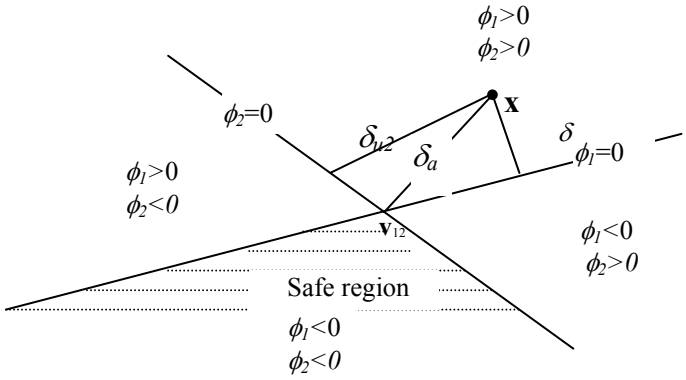


Figure 2-3: The relation between DSM and the minimum distance to the boundaries

For simplicity, (2.14) can also be used to calculate DSM if more than one constraint is violated, which gives an approximate solution. In case of Figure 2-3, assume that $|\delta_{u2}| \geq |\delta_{u1}|$, then the DSM value, calculated using (2.14), is $|\delta_{u2}|$ that is the closest one to the actual value $|\delta_a|$.

Example 2.1

The state space model of a separately exited DC motor (Figure 2-4) is given by

$$\dot{\mathbf{x}} = \begin{bmatrix} \frac{-f}{J} & k_t \\ \frac{-k_t}{L} & \frac{-R}{L} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & \frac{-f}{J} \\ \frac{1}{L} & 0 \end{bmatrix} \mathbf{u}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x} \qquad (2.16)$$

where $\mathbf{x} = [\omega \quad I_a]^T$, $\mathbf{u} = [v_I \quad T_l]^T$, ω is the motor speed (angular velocity), I_a is the armature current, f is the friction coefficient of the motor, J is the moment of inertia, k_t is the torque constant of the motor, L is the motor armature inductance, R is the motor armature resistance, v_I is the input voltage and T_l is uncontrollable input which represent the load torque

At steady state the relation between armature current and motor speed will be

$$I_a = (\omega f + T_l)/k_t$$

Taking into consideration that:

1. The safety variables are all the state variables (speed and current);
2. For simplicity, all motor parameters (R, L, f, k_t, and J) are unity;
3. The maximum speed is 3 rad/s and armature current 3 A;
4. The load torque varies from 0 to 0.5 Nm.

In other words, Φ (the safe operation region) is given by:

$$\begin{aligned} & I_a - (f\omega + 0.5)/k_t \le 0 \\ & I_a - (f\omega + 0.5)/k_t \le 0 \\ & 0 \le I_a \le 4 \\ & 0 \le \omega \le 4 \end{aligned} \tag{2.17}$$

This region is depicted in Figure 2-5. In this example, Φ is defined in the first quadrant only for simplicity.

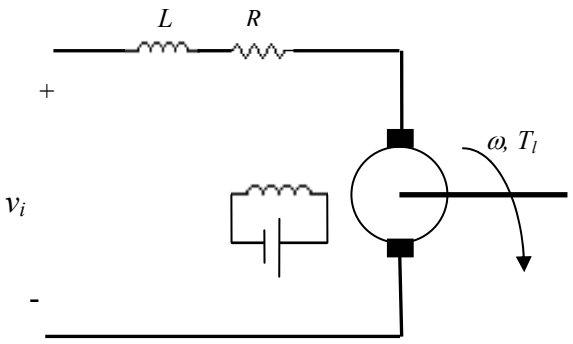


Figure 2-4: Separately excited DC motor

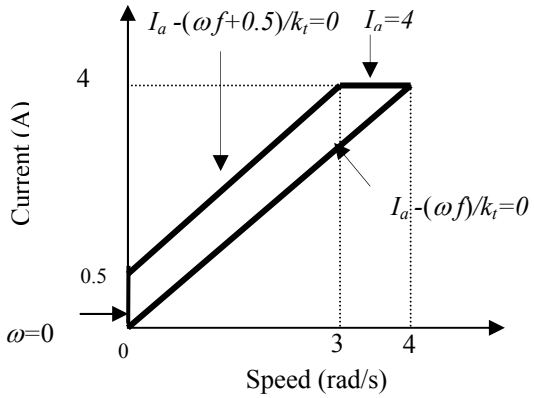


Figure 2-5: Safe region of the DC motor in *Example* 2.1

The open loop response of this motor and DSM variations for a step input of 4v are shown in Figure 2-6. Note that d1, d2, d3, and d4 are the minimum distances between the motor trajectory and the safe region boundaries (b1, b2, b3, and b4). It is clear that DSM at a time *t* is the minimum value of {d1,…,d4}.

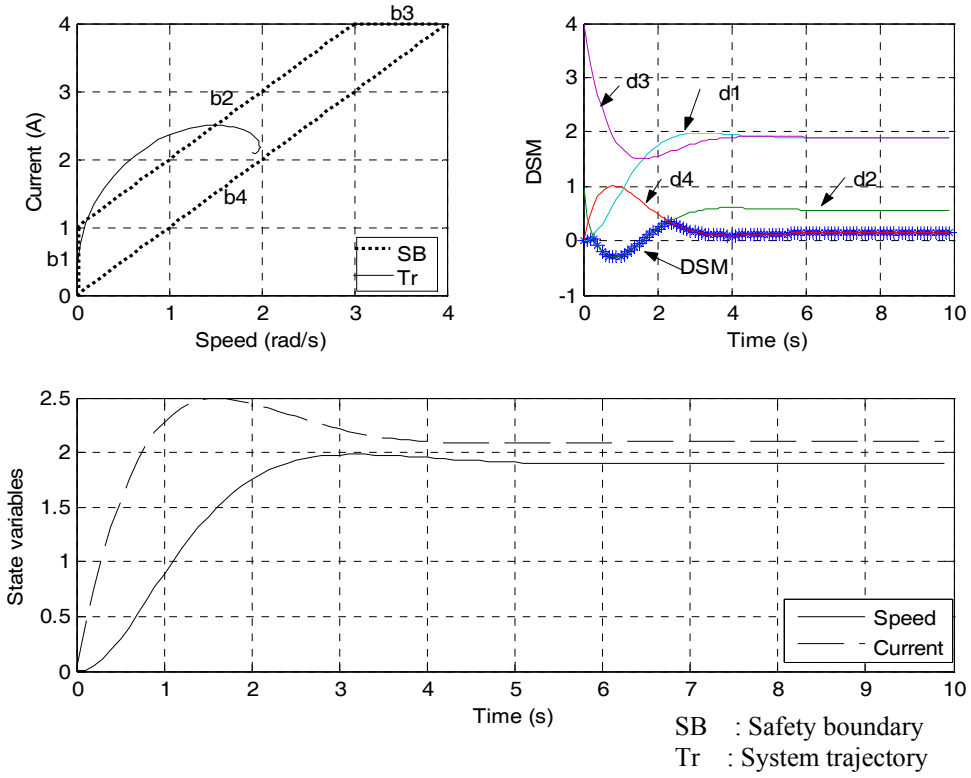


Figure 2-6: Open loop response of the DC motor

To maintain the system state within a predefined margin of safety, the value of DSM must be considered in controller design. Implementing DSM in a controller can be achieved by various methods.

The safe region Φ can be considered, as a controlled invariant set [144], [145] if there is a controller, which assures that DSM is positive for the closed loop system.

Definition 2.2: The set $\Phi \subset \Re^n$ is said (robustly) controlled invariant for the system

$$\Delta \mathbf{x}(t) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)),$$
$$\mathbf{y}(t) = g(\mathbf{x}(t))$$

If for all $\mathbf{x}(0) \in \Phi$ there is a continuous feedback control law

$$\mathbf{u}(t) = \psi(\mathbf{y}(t)) \text{ or } \mathbf{u}(t) = \psi(\mathbf{x}(t))$$

which assures the existence and uniqueness of the solution, $\mathbf{x}(t) \in \Phi$, and then Φ is positively invariant for the closed loop system.

where $\mathbf{x}(t) \in \Re^n$ is the system state, $\mathbf{u}(t) \in \Re^m$ is the control input, $\boldsymbol{y(t)} \in \Re^p$ is the output, $w(t)) \in W \subset \Re^q$ is the external input (disturbance), W is assigned compact set, and Δ is the derivative operator in continuous time and shift operator in discrete time case.

Hence, the invariance condition can be defined as

$$\text{dist}(\mathbf{x}, \Phi) = \inf_{\mathbf{x}_i \in \Phi} \|\mathbf{x} - \mathbf{x}_i\| = 0$$

this means that DSM ≥ 0.

It is not possible in all cases to find a linear controller to a controlled invariant polytope [144], [146], it is often necessary to consider non-linear control laws see for example [146]-[149]. In the following section DSM applications and some ideas about designing controller based on DSM are discussed

2.3 DSM Applications

DSM can be used in different applications, for example:

1. *Controller design*: DSM is an indication to system safety. Hence, controller design based on DSM is important for a safety-critical system to maintain a predefined margin of safety during transient and in the presence of large disturbance or system uncertainties.
2. *Controller evaluation and performance analysis:* DSM can be used as additional performance index to evaluate the controller behaviour and safety performance of the system. Hence, it can be used as a quality measure for the control system.

3. *Fault diagnosis and prognosis*: According to the definition of safe region and DSM, the analysis of DSM can help in fault diagnosis and prognosis.
4. *Fault tolerant control and performance recovery:* DSM index can also be used in designing FTC system, in order to compensate the uncertainties in fault information.

Using DSM in fault diagnosis and fault tolerant control will be addressed in *Chapter 3* and *Chapter 4* respectively. In the following, applying DSM to improve the system performance during transient and in steady state, in the presence of a disturbance or occurerance of "faults" without prior fault information, is explained.

The benefits of employing DSM will be clear in the response of the DC motor (*Example* 2.1). The block diagram of the motor with PID controller, employing analogical gates [151] for anti-reset wind-up, is shown in Figure 2-7. The input voltage to the motor is limited to ±5 v. The strategy for anti-rest wind-up is as follows:

- In linear control range, neither the magnitude nor the sign of the integral-gain (K_I) is changed.
- When commend-saturation occurs, the magnitude of the K_I gain is reduced first.
- As the difference between the saturated (u) and the unsaturated command (u_o) further increases, the sign of KI is made negative together with further decrease of the magnitude.

The strategy of employing analogical gates for anti-reset wind-up is implemented using a single analogical-gate, namely the XOR-gate (see *Appendix* B and [151] for details) as follows:

$$K_i = K_{io}\left[\left(\left(u - u_o\right)/u_o\right) \oplus \left(u/u_o\right)\right] \tag{2.18}$$

where K_I and K_{Io} are the current and the initial integral-gain respectively. The unsaturated and saturated control commands are u_o and u respectively.

Figure 2-8 shows the motor speed response and DSM variation for step reference speed of 2 rad/s and load torque 0.2 N.m, using PID controller with tuned parameter K_p=4, K_I=2 and K_d=2. Note that, the response (transient and steady state) of the motor speed is satisfactory, but the motor state trajectory traverses in the transient period outside the safe operation region (DSM negative). In order to improve the DSM during the transient period, a controller must be redesigned, or the PID controller parameter

must be retuned. The method of tuning PID controller parameter based on DSM is discussed in the *Chapter 4*.

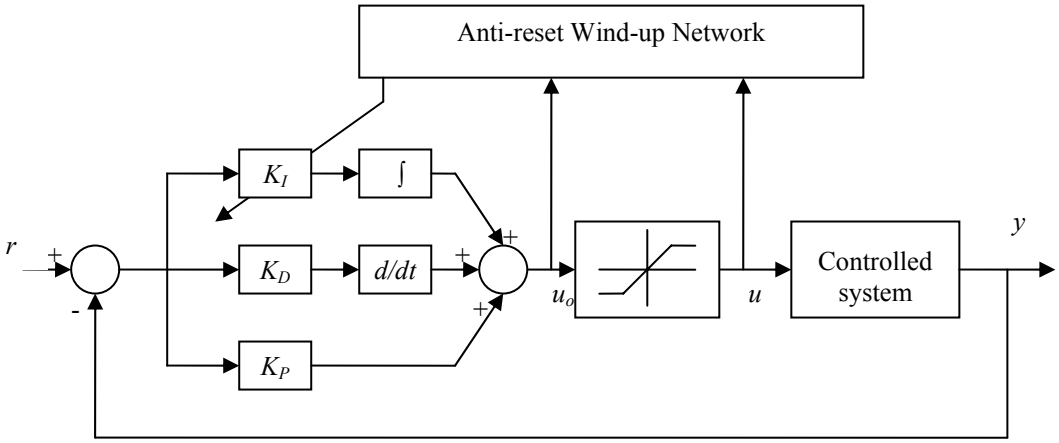


Figure 2-7: PID-controller with saturation employing analogical-gates for anti-reset wind-up

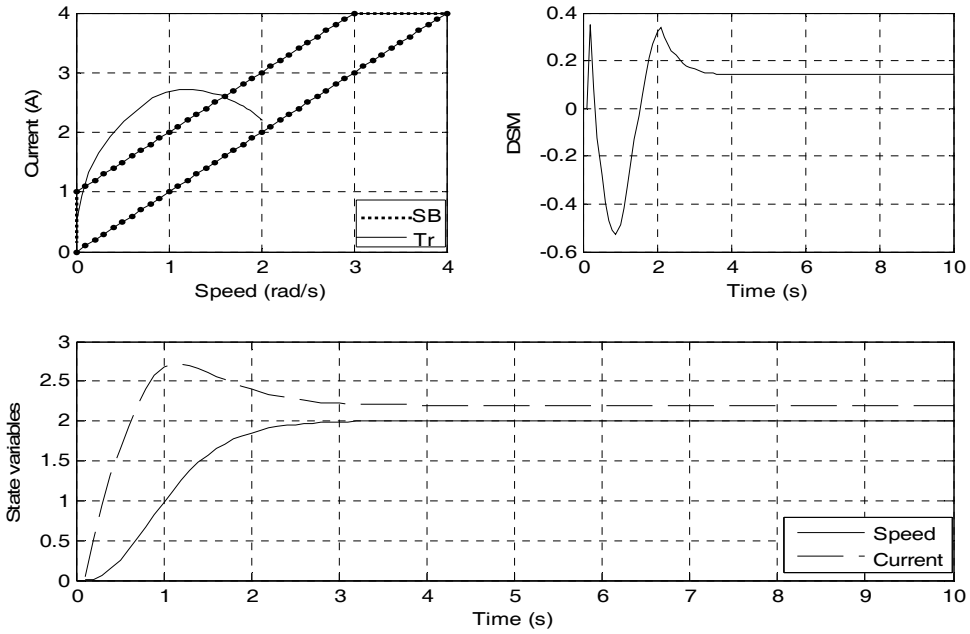


Figure 2-8: DC Motor response and DSM variation using fixed parameter PID-controller

2.3.1 Effect of DSM Design during Transients and in the Presence of Disturbances

Consider that the motor was suddenly exposed to a load torque disturbance from 0.2 to 0.5 Nm after 10s from the motor start. Three controllers including DSM action are tested in this section.

2.3.1.1 Multi-Controllers with Supervisor

DSM can be used as a control signal to switch between two different controllers. The first one operates when DSM is positive (normal operation) and the other when DSM is negative (unsafe operation). The first controller is designed to satisfy the nominal performance, while the priority for designing the second controller is to improve DSM rather than the desired output performance.

Figure 2-9 shows the block diagram of DC motor with two PID controllers (with different parameters) and a supervisory controller to switch between them. If DSM is positive, then the switch moves toward PID_1 and the input to the controller is the error. Otherwise, it moves to PID_2 as shown in the supervisor automata in Figure 2-9b. The input to the second controller (PID_2) is the DSM.

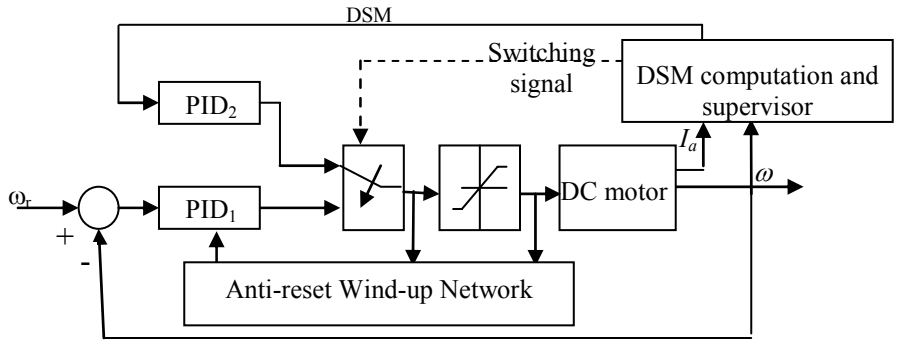
Note that if $\delta(.)<0$, then DSM is the distance between the current state and one of the violated constraints according to the approximate solution (2.14) of *Theorem 2.1*, therefore

$$\begin{aligned}\delta(k) &= \frac{c_i - \mathbf{a}_i^T(k)}{\|\mathbf{a}_i\|} \\ &= \frac{c_i - a_{i1}\omega(k) - a_{i2}I_a(k)}{\|\mathbf{a}_i\|} \\ &= \frac{a_{i1}}{\|\mathbf{a}_i\|}\left[\overline{\omega}_r(k) - (k)\right]\end{aligned} \tag{2.19}$$

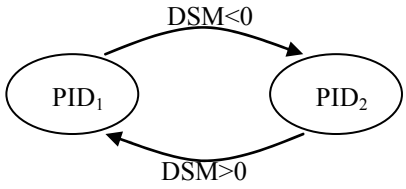
where c_i and $\mathbf{a}_i=[a_{i1}\ a_{i2}]^T$ are the parameters of the violated constraint number i and $\overline{\omega}_r(k) = (c_i - a_{i1}I_a(k)/a_{i1}$

DSM in (2.19) represents the error of the output speed with respect to a new reference $\overline{\omega}_r(k)$, which changes according to the current state and the violated constraint used in (2.19), in order that the state trajectory traverses toward the safe region. Therefore, the input to the second PID controller is DSM instead of the error between the output and nominal reference.

Figure 2-10 shows the DC motor response using switching controller. Note that DSM is improved in the transient period, and the disturbance effect is decreased but the state trajectory is not smooth, and it eventually leaves the safe area. Hence, the state trajectory can be smoothed either by readjusting the two different PID controllers or by changing the switching criteria. Adjusting the two controllers gives a smooth response as shown in Figure 2-11. However, the response is slower than that in Figure 2-10.



a) Block diagram of switching controller



b) Supervisor automata

Figure 2-9: Block diagram of DC motor with two controllers and supervisor

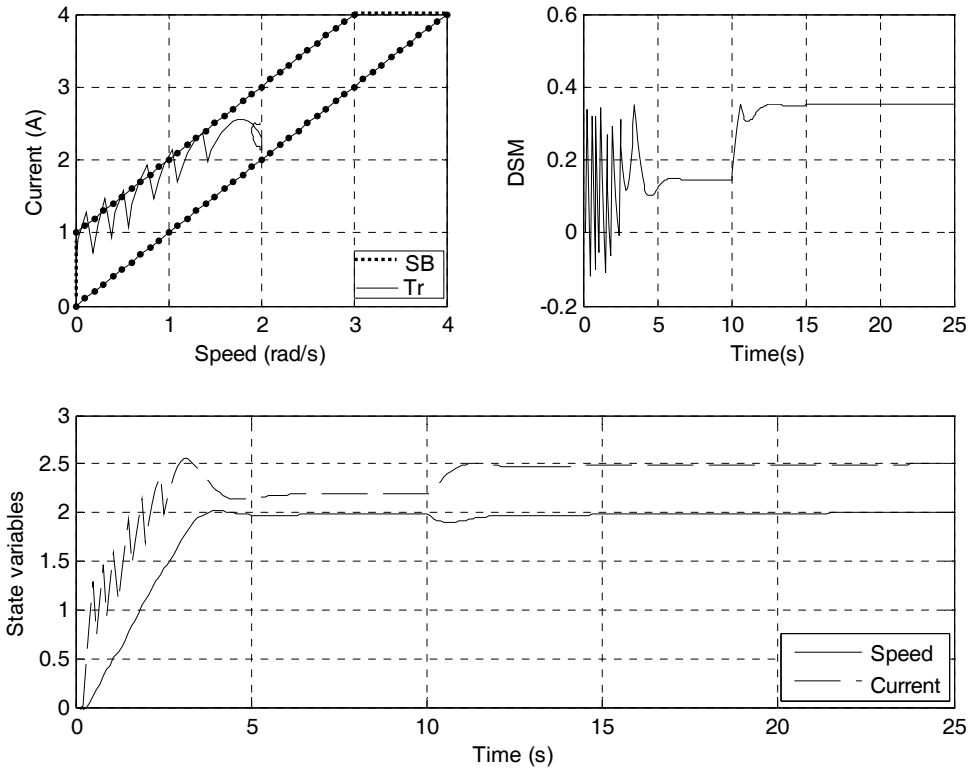


Figure 2-10: DC motor response and DSM using switching controller with

PID1: K_P=4; K_I=1; K_D=1.1
PID2: K_P=2; K_I=2; K_D=2.0

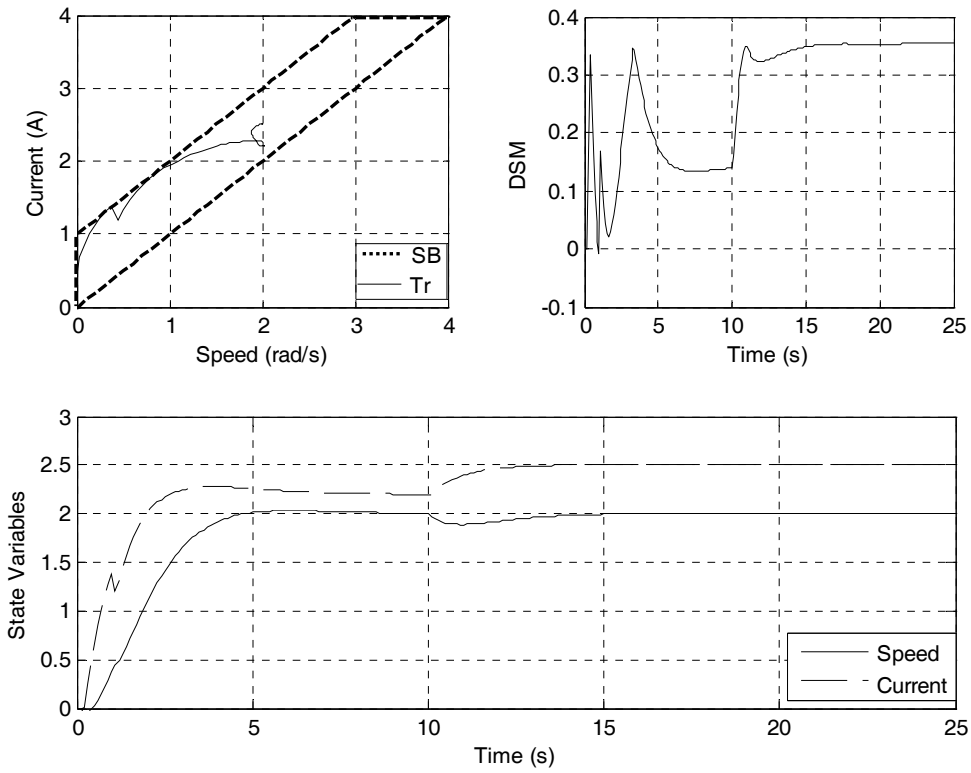


Figure 2-11: DC motor response and DSM using switching controller with

PID1: K_P=1; K_I=1; K_D=1.1
PID2: K_P=2; K_I=2; K_D=2.0

2.3.1.2 Optimal Control

Optimal control can be used to improve system performance and DSM. In this case, the control problem can be solved as a Linear Quadratic Tracking problem (LQT) [152] to find the state feedback gains. The DSM can be added to LQT as inequality constraints in state and the performance index is

$$J = \sum_{k=0}^{\infty} \mathbf{u}(k)^T \mathbf{R}\, \mathbf{u}(k) + \mathbf{e}(k)^T \mathbf{Q}\, \mathbf{e}(k)$$

subject to

$$\mathbf{x}(k+1) = \mathbf{A}\, \mathbf{x}(k) + \mathbf{B}\, \mathbf{u}(k)$$
$$\mathbf{y}(k) = \mathbf{C}\, \mathbf{x}(k)$$
$$\mathbf{x}(k+1) \in \Phi \text{ or}$$
$$\delta(k+1) \geq 0 \text{ i.e. } \mathbf{d}(k+1) \geq 0$$

The optimal control problem with inequality constraints is an infinite horizon optimization problem with infinite number of constraints. The solution of this type of

problem is so difficult therefore MPC is preferred in this case (see *Chapter 4*). If Φ is convex and the safety boundaries are linear, then DSM constraints can be added as additional term in the main objective function, i.e. DSM constraints are considered as soft constraints to simplify the solution.

The new objective function will be, in this case

$$\begin{aligned} J &= \sum_{k=0}^{\infty} \mathbf{u}(k)^{\mathrm{T}}\,\mathbf{R}\,\mathbf{u}(k) + \mathbf{e}(k)^{\mathrm{T}}\,\mathbf{Q}\,\mathbf{e}(k) + \mathbf{d}(k)^{\mathrm{T}}\,\mathbf{P}\,\mathbf{d}(k) \\ &= \sum_{k=0}^{\infty} \mathbf{u}(k)^{\mathrm{T}}\,\mathbf{R}\,u(k) + [\mathbf{e}(k)\;\mathbf{d}(k)]^{\mathrm{T}} \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{e}(k) \\ \mathbf{d}(k) \end{bmatrix} \end{aligned} \tag{2.20}$$

subject to

$$\begin{aligned} &\mathbf{x}(k+1) = \mathbf{A}\,\mathbf{x}(k) + \mathbf{B}\,\mathbf{u}(k) \\ &\mathbf{y}(k) = \mathbf{C}\,\mathbf{x}(k) \\ &\mathbf{d}(k+1) = \mathbf{D}_c - \mathbf{D}_a\mathbf{x}(k+1) \\ &\mathbf{d}(k) = [\delta_1(k)\;\delta_2(k)\cdots\delta q(k)]^{\mathrm{T}} \\ &\mathbf{e}(k) = \mathbf{y}_d(k) - \mathbf{y}(k) \\ &\delta(k) = \min_{1\le i\le q}\left(\delta_i(k)\right) \end{aligned}$$

Here $\delta_i(k)$ is the minimum distance to each boundary of the safe operation region Φ (2.12); $\delta(k)$ is the DSM at instance k; i is number of inequality constrains; **u** is the control signal vector; **e** is the error vector between the actual response and the desired response; **Q**, **P**, and **R** are the weighting matrices; **d**(.) is calculated from (2.13).

The control law will be

$$\mathbf{u}(k) = \mathbf{r}(k) - \mathbf{K}_f\,\mathbf{x}(k) \tag{2.21}$$

where $\mathbf{K}_f$ is the state feedback gain matrix and **r** is the reference inputs.

Figure 2-12 shows the DC motor response and DSM variations using the optimization algorithm with the following parameters:

$$\mathbf{R} = [2],\ \mathbf{Q} = [11.5],\ \mathbf{P} = diag[0.1,1,0.1,0.1],\ \text{and}\ \mathbf{K}_f = [0.457\ \ 0.5739],$$

Note that the DSM values are positive for the whole operation period, which means that the motor operates safely.

2.3.1.3 Adaptive Control

To maintain the system state within a predefined margin of safety, the value of DSM can be taken as an index to adapt controller parameters. The controller parameters should be adapted when the DSM is relatively positive small or negative, otherwise the parameters have to be maintained without change.

2.3.1.3.1 Linear Adaptation

The adapted parameters can be defined as a linear function of the DSM, and calculated from the following equation:

$$k_i(t+\Delta t) = k_i(t) + \alpha_{1i}\delta(t) + \alpha_{2i}\frac{\partial\delta(t)}{\partial t};\ i=1,2,\ldots,N \tag{2.22}$$

where $\delta(t)$ is the DSM at any instance t, k_i is the controller parameter number i, N is the total number of controller parameters, and α_i is the adaptation parameter.

The results of (2.22) may not guarantee that the adapted gains will maintain the state in the safe region in all cases (positive DSM). Therefore, replacing $\delta(.)$ in (2.22) by the term $\frac{\partial\delta(.)}{\partial k_i}$ with appropriate choice of α_i, could guarantee that DSM is positive. The new adaptation equation will be

$$k_i(t+\Delta t) = k_i(t) + \alpha_{1i}\frac{\partial\delta(.)}{\partial k_i} \tag{2.23}$$

However, $\partial\delta(.)/\partial k_i$ in most cases, is nonlinear and not easy to compute. More details about the parameters adaptation of the controller will be discussed in *Chapter 4*.

2.3.1.3.2 Fuzzy Adaptation

A Fuzzy controller [243] based on DSM can be used to calculate the incremental values in the adapted parameters, where the relation between controller gains and DSM, in most cases, is nonlinear and not easy to compute. The input variables of fuzzy controller are function of DSM, e.g. δ, $\frac{\partial\delta(.)}{\partial t}$, etc., and the output is the incremental value in the adapted parameters. The adapted parameter is calculated from the following equation

$$\begin{aligned} k_i(t+\Delta t) &= k_i(t) + F_{yi}(t) \\ F_{yi}(t) &= M_i(\delta(t)) \end{aligned} \tag{2.24}$$

where M_i is the fuzzy function and F_{yi} is the incremental gain, which is equal to the fuzzy output number i.

The fuzzy controller parameters (membership functions, number of variables and their limits,**…**,etc.) are chosen based on the limits of controller gains.

In each method of adaptation, the adapted parameter value should be bounded in the interval, $k_i \in [k_{il}, k_{ih}]$, which satisfies the stability condition of the system.

The complete block diagram of the adapted proportional gain of PID controller based on DSM is shown in Figure 2-13. Figure 2-14 shows the DC motor response using adapted proportional gain. It is clear that the transient and DSM are improved, and the torque disturbance effect is reduced.

The different responses of the DC motor show that the system operates in safe mode at the transient as well as at the steady state, either in a normal operation or in a disturbance case, when DSM is considered in the controller design. Furthermore, DSM can be used as an index to evaluate the different method of control design.

Note that the output response and DSM response of the controlled system change according the priorities in the controller design. For example, if the priority to satisfy DSM is higher than the output, then it is necessary to make DSM more positive even if the output response will degrade, and vice versa.

Adapted PID controller based on DSM is tested in the next section to recover level performance of one-tank system close to the nominal performance due to tank leakage.

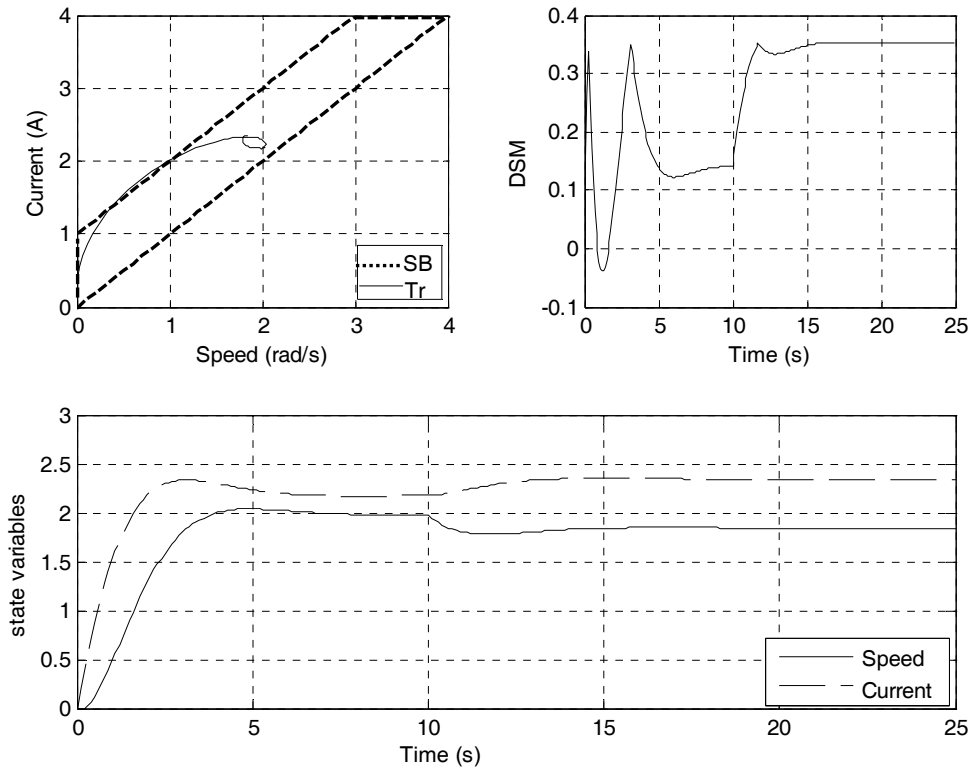


Figure 2-12: DC motor response and DSM using LQT controller

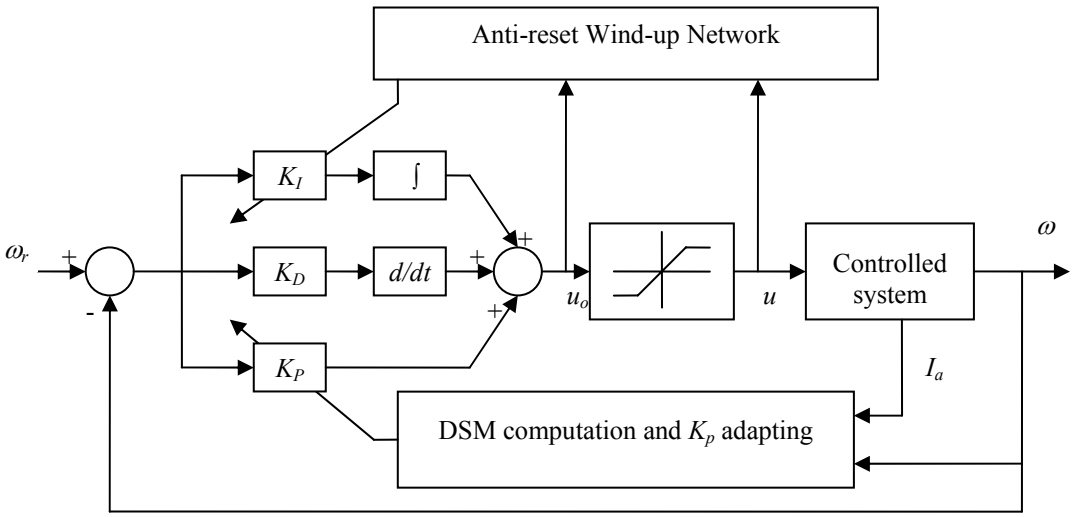


Figure 2-13: PID-controller employing analogical-gates for anti-reset wind-up and adapting proptotinal gain based on DSM

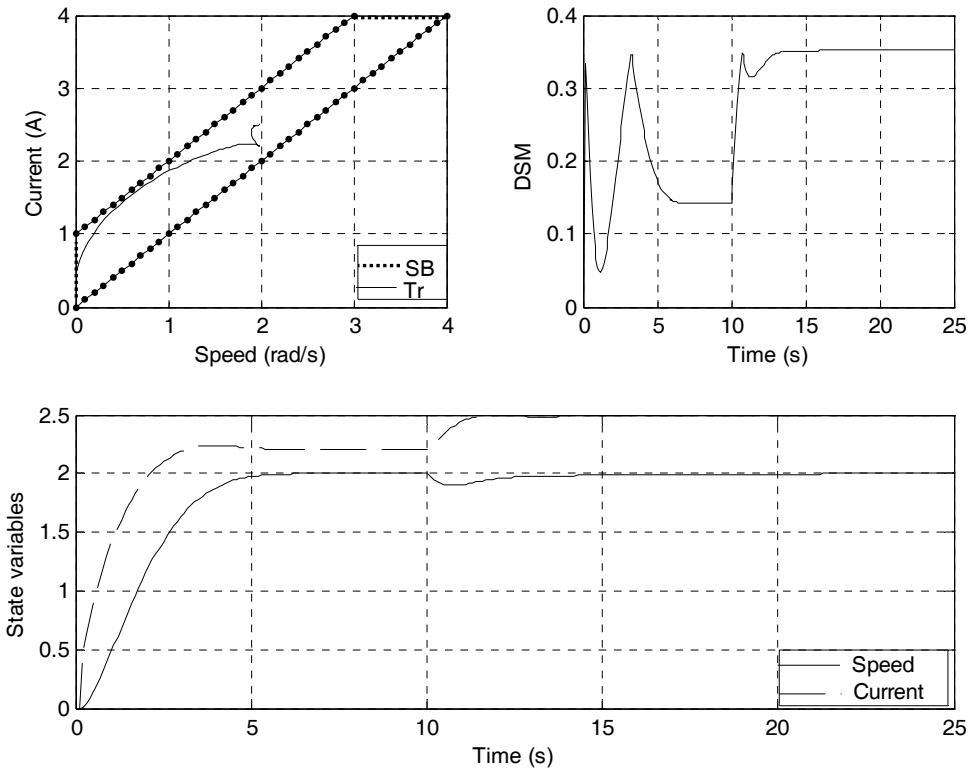


Figure 2-14: DC motor response and DSM using adapted PID controller

2.3.2 Implementation of DSM for System Performance Recovery

The idea of controlling a system that deviates from the nominal operating conditions has been investigated by many researchers. The methods of dealing with this problem usually stem from linear quadratic, adaptive, or robust control [92],[153], [154], and [161]. Most of the methods, used for performance recovery, depend on the diagnosis of the plant and readjust the controller, see *Chapter 1*. Online controller adapting, based on the value of DSM, helps in speeding up the performance recovery close to the nominal performance before the diagnosis of the system has been completed, or the changes in the model parameters are identified. The fault here is considered as unknown disturbance or uncertainties. More investigation about DSM in performance recovery and FTC is addressed in *Chapter 4.* The following example of a level process illustrates the effect of DSM in speeding up the performance recovery.

Example 2.2

An experimental level process, shown in Figure 2-15, which will be explained in *Chapter 5*, consists of one-tank system [134], [135], [155], [156]. The input flow is

adjusted at 1 l/s and the level is controlled using the outflow valve. The discrete linear model of the system at sampling rate 0.1 s is:

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.999741 & -0.000694 \\ 0 & 0.740818 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} -0.00010932 \\ 0.25918177 \end{bmatrix} u(k) \qquad (2.25)$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(k)$$

where $\mathbf{x} = \begin{bmatrix} h & v \end{bmatrix}^{\mathrm{T}}$, h is the level in the tank (m) and v the valve limb movement (m).

Consider that

1. The variables relevant to system safety are the tank level rate (dh/dt) and the control signal (u) which simulate the valve opening (ν);
2. the level rate (dh/dt) is bounded in the interval [-0.4,0.4] and bounded input $v \in \begin{bmatrix} -0.5, 0.5 \end{bmatrix}$ (0.5 means that the valve is completely open and -0.5 completely close);
3. A valve bias of 20% closing may occur during operation.

then the safe operation region (Φ) is given by:

$$\begin{aligned} & dh/dt + 0.8\, v \leq 0 \\ & dh/dt + 0.8\, v - 0.16 \geq 0 \\ & -0.4 \leq dh/dt \leq 0.4 \\ & -0.5 \leq v \leq 0.5 \end{aligned} \qquad (2.26)$$

This region is shown in Figure 2-16

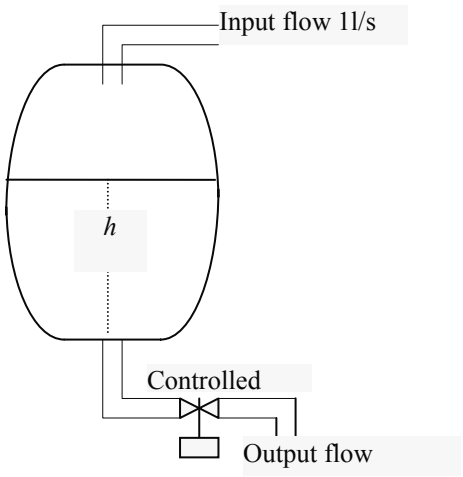


Figure 2-15: Schematic diagram of one-tank level process

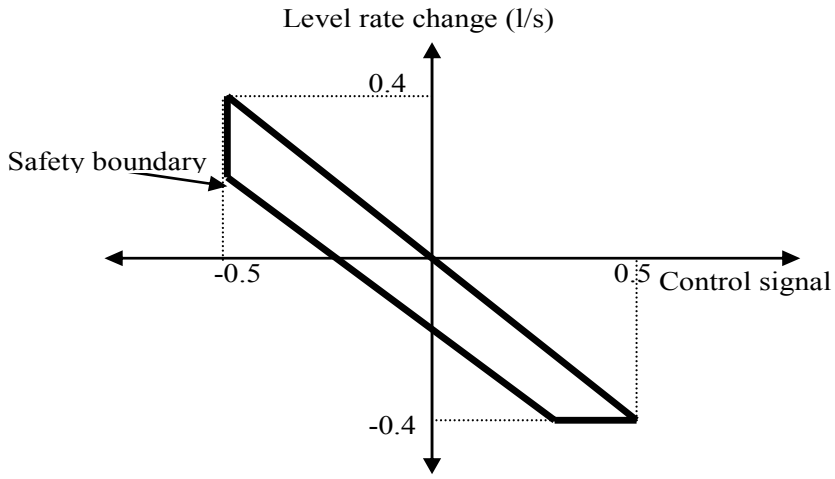


Figure 2-16: Safe region of the level process in *Example 2.2*

Note that firstly, the state vector $\mathbf{x}= [h_1\ v]^T$, secondly, one of the safety variables is $dh/dt = f(x)$ not the state directly, where $f:\Re^n \rightarrow \Re$ is a nonlinear function of **x,**

$$\frac{dh}{dt} = \frac{1}{A(h)}\left(Q_i - kv\sqrt{\rho g h}\right) \tag{2.27}$$

where $A(h)$ is the cross section area of the tank, k valve coefficient, and Q_i is the input flow.

Substituting from (2.27) into the constraint equations (2.26) gives the safety boundary functions. Unfortunately, the boundary functions are nonlinear in this case. Therefore, *dh/dt* is taken as an independent variable that can be easily computed from *h* in order to have linear constraints.

A simulated leakage in the tank about 0.2 l/s has occurred after 200 sec of *Example* 2.1. Figure 2-17 shows the level response using a fixed parameter PID controller. Note that the controller is unable to recover the system performance. Replacing the PID controller with adapted one according to (2.22) yields a response for the tank level as shown in Figure 2-18. It is clear that the level performance is recovered faster and closer to the nominal performance without fault (leakage) diagnosis.

Figure 2-19 shows the level response and DSM variation with adapted proportional gain of PID controller with fuzzy adaptation. The fuzzy supervisor has one crisp input (δ) and one crisp output (incremental propotional gain). The domain of crisp input varible (δ) is divided into five input fuzzy variables: NH, NM, Z, PM, and PH. Whereas, the doman of the output crisp (F_{yi}) variable are devided into three output fuzzy variables: Z, H, and VH. Input/output membership functions are shown in Figure

2-20, and fuzzy allocation matrix is shown in Table 2.1. The normalized input and output signal of the fuzzy controller can help in generalizing the fuzzy supervisor for more than one parameter adaptation. The DSM response can be improved by considering the DSM rate of change in the fuzzy controller design and increasing the number of fuzzy variables.

It is clear, from both examples, that the variables relevant to the safety are not necessary to be the same controlled state (variables). For some proceses, they can be chosen to be mathematical variables related to the controlled state as in *Example* 2.2 in order to simplify the boundary equations. Therefore, the choice of the state variables relevant to the safety is not unique.

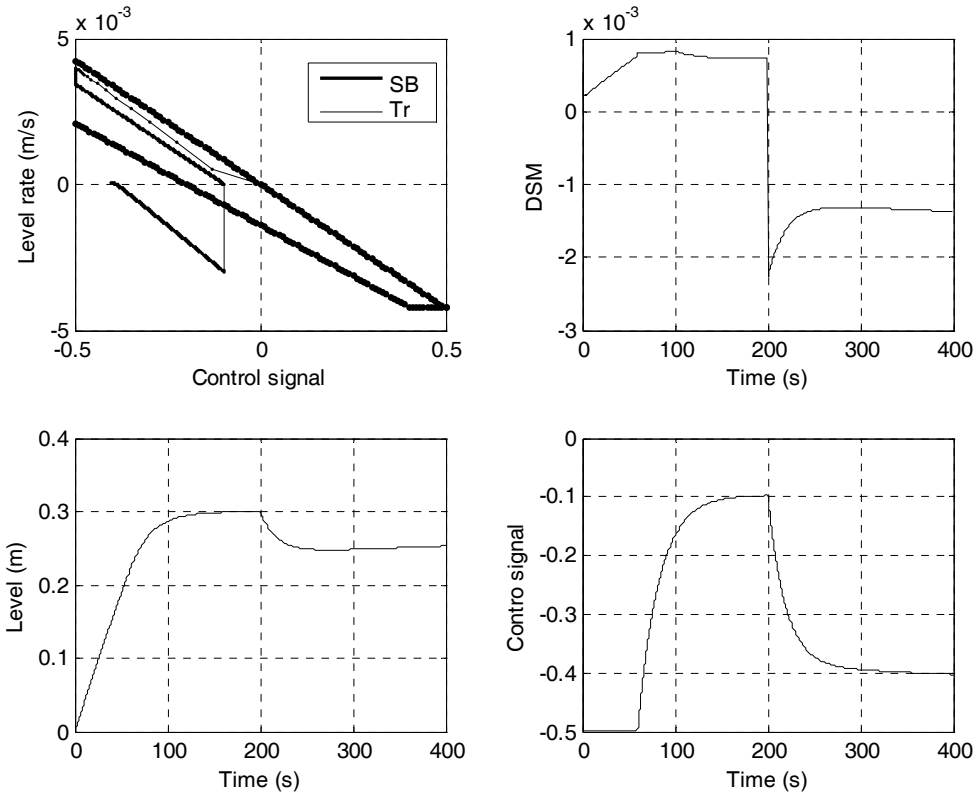


Figure 2-17: Level process response and DSM using PID controller

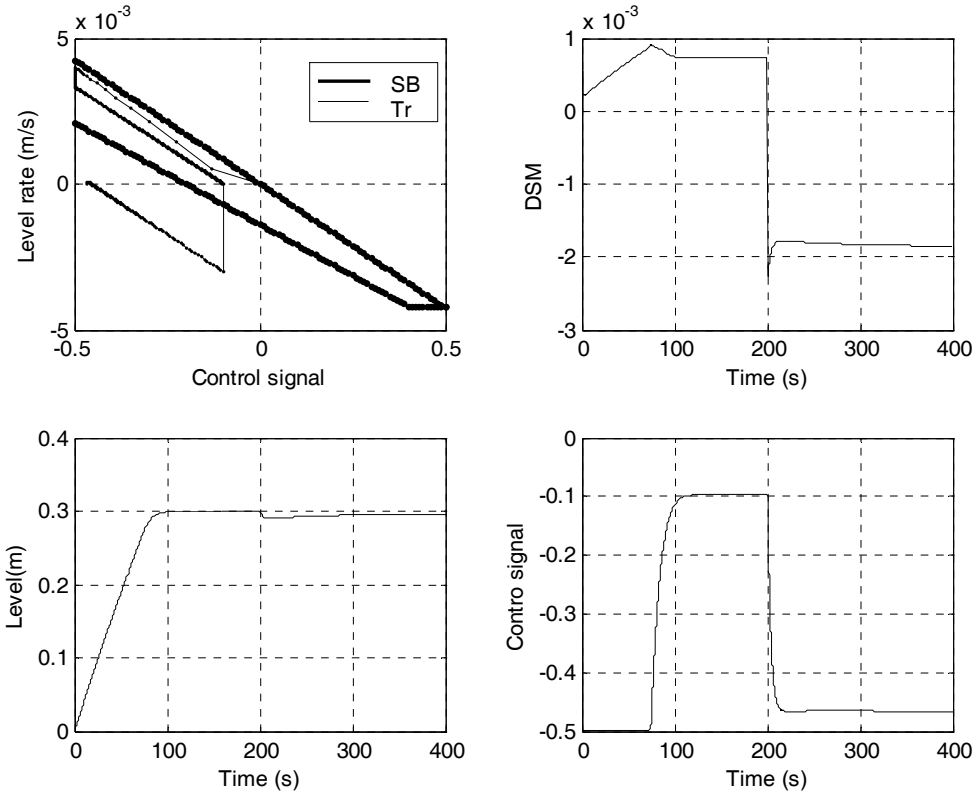


Figure 2-18: Level process response and DSM using adaptive PID controller

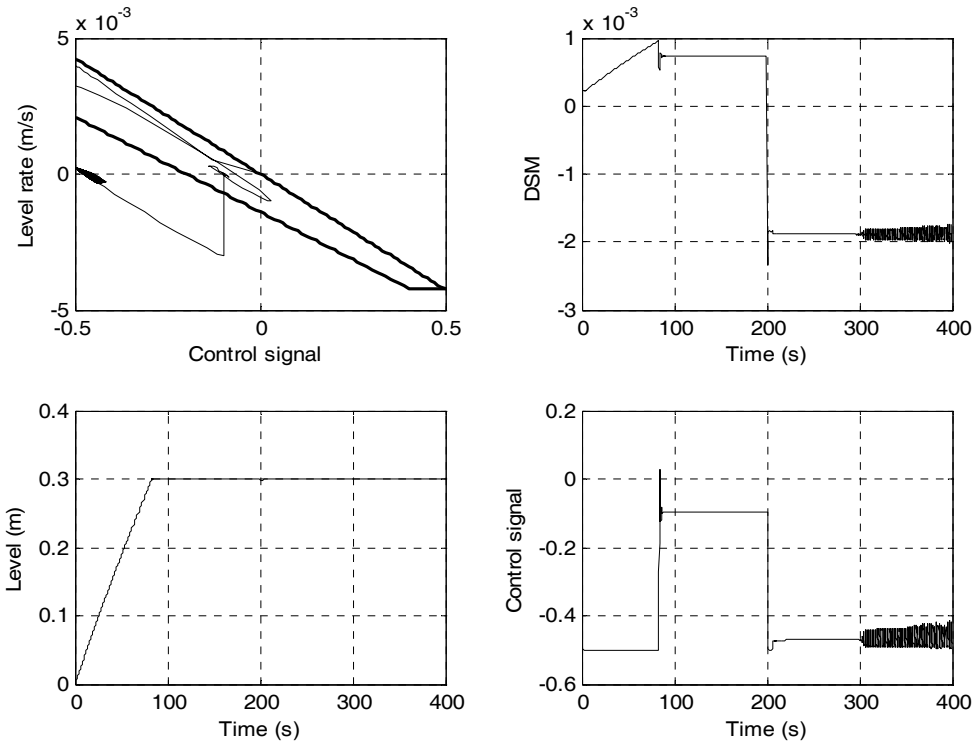


Figure 2-19: Level process response and DSM using adaptive PID controller based on fuzzy adapt

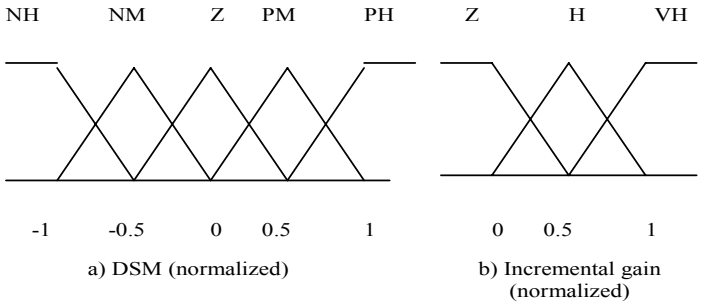


Figure 2-20: Membership function of normalized DSM and incremental gain

Table 2-1: FAM of DSM and incremental gain

δ	NH	NM	Z	PM	PH
F_{yi}	VH	H	H	Z	Z

Note: NH and NM are the abbreviations of negative high and negative medium respectively; PM and PH are the abbreviations of positive medium and positive high respectively. Z, H and VH denote zero, high, and very high respectively.

At this point, the differences between the DSM approach and other related approaches can be discussed:

1. Most of the approaches related to safety state, for instance [157], define the safe values of each state individually, while most of the time, the variables are dependent on one another and none of them adequately defines the system safety by itself. Therefore, DSM represents a general case as shown in Figure 2-2.
2. The distance between current states and safe states is defined as

   $$D = 0 \text{ if } \mathbf{x} \in Xs \text{ and } D>0 \text{ if } \mathbf{x} \notin Xs$$

   where $\mathbf{x}$ is the current state vector, and X_s all safe state vectors [157]. This means that system behaviour inside the safe region is not taken into consideration. On the other hand, DSM has a positive value inside safe region and negative otherwise. Therefore, the value of DSM indicates the safety state more precisely.
3. The safe region can be defined, as a controlled invariant set [144], [158], [202] for the system if there is a controller, which ensures a positive DSM for the closed loop performance.

4. One of the main differences between invariant set and safe region Φ is that the defined safe region for the system is assumed constant if the system structure is changed, while invariant set should be defined for each structure.
5. The problem of finding a controller for a system with state and input constraints has been the subject of study of many authors; see for example [159]-[160], [202], and [185]. Controller design based on DSM can be considered as controller design for a system with state constraints. More investigation about controller design for system with state constraints and DSM is discussed in *Chapter 4*.

However, the main limitation of applying DSM is the determination of the safe region. In some processes, it is not quite easy to determine the safe operation region. Moreover, the mathematical formulation of the DSM is not easy to obtain for some shapes of the safe operation regions. DSM computation and application for large-scale system need excessive study.

2.4 Conclusions

In this chapter, a new definition (DSM) and its computation are presented. Some applications of DSM are stated as well, which cover the applications of DSM in, first, controller design; second, FDI design; third, FTC design. The advantages of controller design based on DSM are discussed. DSM can be used to control the safety of the system during transient and steady state operation, to decrease the disturbance effect, and to help speeding up the performance recovery in case of some system faults. Adaptive PID controller, LQT optimization, and switching controller based on DSM are tested using simulation example. The variables of the system cannot describe safe behaviour of the system individually. Thus, the relation between state boundary and DSM is discussed. It is clear that DSM can be used to evaluate the performance of different control design method as a safety index. Hence, DSM can be considered as a quality measure of the controller performance. Because of occurring most of failures in the transient phase, designing a controller to maintain a margin of safety at transient period of the system is important. The difference between the DSM concept and the related concepts are discussed as well.

The choice of the state variables relevant to the safety and the determination of the associated safe operation region are not unique because they depend mainly on the

operation experience of the process (knowledge based). For some processes, it is not easy to find a mathematical formulation for the DSM due to the complicated shapes of the safe operation regions. In this case, a knowledge-based model (fuzzy, neural,…,etc.) can be used.

CHAPTER 3

FAULT DETECTION AND DIAGNOSIS SYSTEM USING DYNAMIC SAFETY MARGIN

3.1 Introduction

In order to meet the increasing requirements of modern society, industrial processes become large and complex. The complexity makes the systems are vulnerable to faults. Moreover, a fault in a single component may cause a malfunction of the whole system. To achieve the increasing economic demands and safety restriction, high dependability of such processes becomes an essential demand. It is difficult for humans to troubleshoot such systems. Consequently, early fault detection and diagnosis are a vital task. Thus, an extensive research has been done in the field of FDI design. The major FDI methods stated in literature, *see Chapter1*, can be classified into three broad categories; (a) Model based, (b)Knowledge based, and (c) signal based. Most of Model-based FDI systems depend mainly on the analysis of so-called residuals [7]-[9], [36] generated from the input and the output signals and applying dynamic process model. Residual generation is based, e.g., on parameters estimation, parity equation or state observers of the process. The generation of residuals is the first stage in FDI system. Designing a residual generation system, which is insensitive to model parameter variations and external disturbances, is a formidable task and called a robust FDI system. In general, designing a robust FDI system is quite difficult. The robustness is addressed, for linear and nonlinear systems; by different ways, see for example [73], [87], [88], [78] and [153]. In this chapter, a robust FDI problem is explained, and the existing approaches and their limitations are discussed. In addition, a new approach for model-based FDI, which depends on the analysis of the DSM, is introduced. The idea of "Multiple-Model" (MM) system is the basis for the fault diagnosis method used here.

3.2 Robust Fault Detection System

The fault is detected in Model-based FDI by comparing the actual process behavior with the corresponding mathematical model behavior. Since an accurate mathematical model of a physical process is not always available, there is often a mismatch between the

actual process and its mathematical model, even if no fault in the process occurs. This constitutes a source of false alarm, which can corrupt the performance of the fault detection and diagnosis system. To overcome this difficulty, FDI system has to be made robust to such modeling errors or disturbances. A system which is designed to provide both sensitivity to fault and robustness to modeling error or disturbances is called a robust FDI scheme [42], [13]. Fault, disturbance, and uncertainties modeling should be described, to clarify robust FDI problem. Therefore, in this section, the effect of disturbances and model uncertainties on the residual generation is introduced, moreover the main limitations in the existing FDI methods.

3.2.1 Fault Modeling

A fault is defined as an unpermitted deviation of at least one characteristic property or parameter of the system from the accepted behavior [7]. The fault is the state that may lead to malfunction or a failure in the system. Faults can be classified based on several criteria, such as the time characteristics of faults, physical locations in the system and the effect of faults on the system performance [13]. The time dependency of faults can be distinguished as abrupt fault (stepwise), incipient fault (drift-like) or intermitted fault. When faults are classified according to their physical locations, three main faults can be defined: actuator faults, sensor faults, and plant component faults. Faults in an actuator range from loss off partial control effectiveness (stuck at a fixed value) to a complete loss of control. Since an actuator is often considered as the entrance to the system, actuator faults have severe consequences on the system performance. Sensor faults include incorrect readings due to malfunction in sensor circuit elements or transducers. Three types of sensor faults can be identified: dynamic changes in transducer, gain reduction, and unknown bias. Plant component faults cause changes in the dynamical relationship among the system variables. These faults are caused by physical parameters changes in the system, such as resistance, inductance, amplifier gain, etc. If faults are to be classified according to their induced effects on the system performance, they can be classified into two types: additive and multiplicative [8], [7]. Additive faults result in changes only in the mean value of the system output signal, which include sensor bias fault (input and output) and actuators faults. Whereas, multiplicative faults results in changes in variance, correlations of the system output signal, as well as changes in the spectral characteristics and dynamics of the system

which include process components (system parameters) faults and dynamic change in the transducer or gain reduction of the sensor [13].

The system model with faults for a discrete linear time invariant (LTI) system, shown in Figure 3-1, can be represented in the following form:

$$\begin{aligned}\mathbf{x}(k+1) &= \left(\mathbf{A}+\mathbf{A}_f(k)\right)\mathbf{x}(k)+\left(\mathbf{B}+\mathbf{B}_f(k)\right)\mathbf{u}(k)+\mathbf{R}_a\mathbf{f}_a(k)\\ \mathbf{y}(k) &= \left(\mathbf{C}+\mathbf{C}_f(k)\right)\mathbf{x}(k)+\mathbf{R}_y\mathbf{f}_y\end{aligned} \tag{3.1}$$

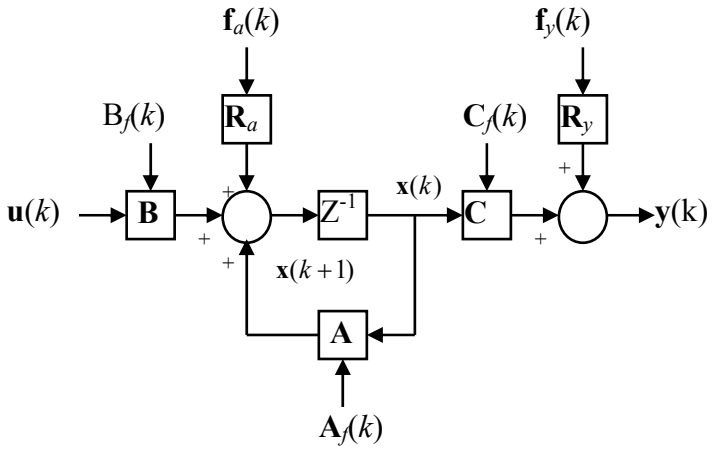


Figure 3-1: State space and fault modeling

Equation (3.1) can be written as

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k)+\mathbf{B}\mathbf{u}(k)+\mathbf{A}_f(k)\mathbf{x}(k)+\mathbf{B}_f(k)\mathbf{u}(k)+\mathbf{R}_a\mathbf{f}_a(k)\\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)+\mathbf{C}_f(k)\mathbf{x}(k)+\mathbf{R}_y\mathbf{f}_y\end{aligned} \tag{3.2}$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the input vector, $\mathbf{y}$ is the output vector, $\mathbf{f}_a$ is the input or state variable fault and $\mathbf{f}_y$ the output faults, which represent the additive faults; $\mathbf{A}_f$, $\mathbf{B}_f$, and $\mathbf{C}_f$ are fault parameters, which represent the multiplicative faults; **A, B**, and **C** are the nominal system parameters; $\mathbf{R}_a$ and $\mathbf{R}_y$ are distribution fault matrix with appropriate dimensions.

Different approaches for fault detection using mathematical model have been developed in the last 30 years, see, e.g., [7], [10], [53], [63]-[67],[78], and [162]. The task consists of the detection of fault in the processes, actuators, and sensors by using the dependencies between different measurable signals. Mathematical process models express these dependencies.

The basic idea of model-based FDI, as mentioned in *Chapter 1*, is to generate analytical redundancy with the help of a mathematical model of the diagnosed system. The fault-indicating signal, called usually "residual", is generated by comparing the measured output or the state with the estimated ones.

3.2.2 Residual Generation Methods

The generation of symptoms (residual) is the main issue in model-based fault detection. Varieties of methods are available in literature for residual generation. Observer-based approaches, parity space approaches and parameter estimation approaches are the most popular approaches to produce residuals [7], [10], [8], [78]. The use of these approaches differs according to the fault types and the system model.

3.2.2.1 Observer-Based Approaches

The basic idea behind the observer or filter-based techniques is to estimate the output or state of the system from the measured using, Luenberger observers in a deterministic system or Kalman filters in a noisy environment. The output or state estimate error (or its weighted value) is therefore used as a residual. The advantage of the using observer is its flexibility in the selection of its gains, a matter that leads to a rich variety of FDI schemes [40],[42], [54], [41]. Fault modeling is then performed with additive faults for the input (additive actuator or process faults) and the output (sensor faults).

Consider a discrete LTI model for the process under consideration

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \tag{3.3}$$

where $\mathbf{u}(k) \in \Re^r$ is the input vector, $\mathbf{x}(k) \in \Re^n$ is the state vector and $\mathbf{y}(k) \in \Re^m$ is the output vector and assume that all matrices **A**, **B** and **C** of the system are perfectly known.

According to Figure 3-2, the following equations hold if there are no disturbances, noises, and parameters changes.

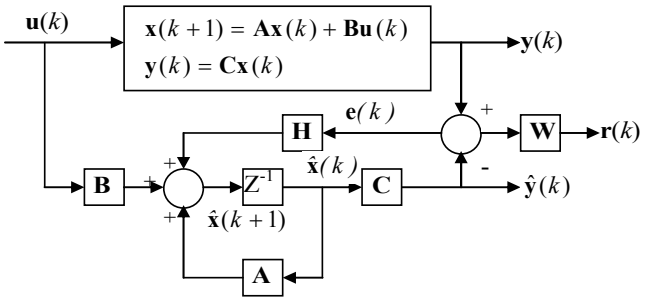


Figure 3-2: Process and state observer

$$\mathbf{e}_x(k+1) = (\mathbf{A} - \mathbf{HC})\mathbf{e}_x(k) + \mathbf{R}_a\mathbf{f}_a(k) - \mathbf{HR}_y\mathbf{f}_y(k) \tag{3.4}$$

and the output error e(k) becomes

$$\mathbf{e}(k) = \mathbf{Ce}_x(k) + \mathbf{R}_y\mathbf{f}_y(k) \tag{3.5}$$

and the residual

$$\mathbf{r}(k) = \mathbf{We}(k) \tag{3.6}$$

When a sudden and permanent fault $\mathbf{f}(k)$ occurs, the state estimation error will deviate from zero.

$\mathbf{e}_x(k)$ and $\mathbf{e}(k)$ show dynamic behaviours, which are different for $\mathbf{R}_a\mathbf{f}_a$ and $\mathbf{R}_y\mathbf{f}_y$. Both $\mathbf{e}_x(k)$ or $\mathbf{e}(k)$ can be taken as residuals.

For the generation of residual with special properties, the design of the observer feedback matrix **H** is of interest [78], [37]-[39].

Limiting conditions are the stability and sensitivity against disturbance. If the signals are affected by noise, the Kalman filter must be used instead of classical observers, assuming the noises are Gaussian white noise [13], [8].

If the faults appear as changes $\mathbf{A}_f$, $\mathbf{B}_f$, or $\mathbf{C}_f$ of the parameters, the process behavior becomes

$$\begin{aligned} \mathbf{x}(k+1) &= (\mathbf{A} + \mathbf{A}_f(k))\mathbf{x}(k) + (\mathbf{B} + \mathbf{B}_f(k))\mathbf{u}(k) \\ \mathbf{y}(k) &= (\mathbf{C} + \mathbf{C}_f)\mathbf{x}(k) \end{aligned} \tag{3.7}$$

while the state $\mathbf{e}_x(k)$ and the output estimate $\mathbf{e}(k)$ errors are

$$\begin{aligned} \mathbf{e}_x(k+1) &= (\mathbf{A} - \mathbf{HC})\mathbf{e}_x(k) + \mathbf{A}_f\mathbf{x}(k) + \mathbf{B}_f\mathbf{u}(k) - \mathbf{HC}_f\mathbf{x}(k) \\ \mathbf{e}(k) &= \mathbf{Ce}_x(k) + \mathbf{C}_f\mathbf{x}(k) \end{aligned} \tag{3.8}$$

The changes $\mathbf{A}_f$, $\mathbf{B}_f$, and $\mathbf{C}_f$ are then multiplicative faults [7], [13]. In this case, the changes in the residuals depend on the parameter changes, as well as input and state variable changes. Hence, the influence of the parameter changes on the residuals is not straightforward as in the case of additive faults.

Special observers were designed and summarized in [162], [35], [78], [13], and [7].

1. *Dedicated observers for MIMO process*

- Observer, excited by one output: one observer is driven by one sensor output. The outputs **y** are reconstructed and compared with measured outputs **y**. This allows the detection of single sensor faults [8], [152];
- Bank of observer, excited by all outputs: Several state observers are designed for a finite fault signal, and detected by hypothesis test [8];
- Bank of observers, excited by single outputs: Several observers for single sensor outputs are used. The estimated output **y** is compared with the measured output y. This allows the detection of multiple sensor faults [8], [163] (Dedicated observer scheme);
- Bank of observers, excited by all outputs except one: As before, but each observer is excited by all outputs, except one sensor output which is supervised [53].

2. *Fault Detection filter (fault sensitive filter) for MIMO processes*

The feedback **H** of the state observer is chosen, so that particular fault signal $\mathbf{f}_a$ changes in a definite direction and signal $\mathbf{f}_y$ in a definite plane. With directional residual vectors, the fault isolation problem consists of determining which of the known fault signature directions the residual vector lies the closest one. More work in this area is found in [41], [43], and [65].

Another possibility is the use of output observer (or unknown input observer), Figure 3-3, if the reconstruction of $\mathbf{x}(k)$ is not of interest. A linear transformation then leads to a new state variable $\mathbf{z}(k)$.

$$\mathbf{z}(k) = \mathbf{T}\mathbf{x}(k) \tag{3.9}$$

The state-space representation of the observer becomes

$$\hat{\mathbf{z}}(k+1) = \mathbf{F}\,\hat{\mathbf{z}}(k) + \mathbf{J}\mathbf{u}(k) + \mathbf{G}\mathbf{y}(k) \tag{3.10}$$

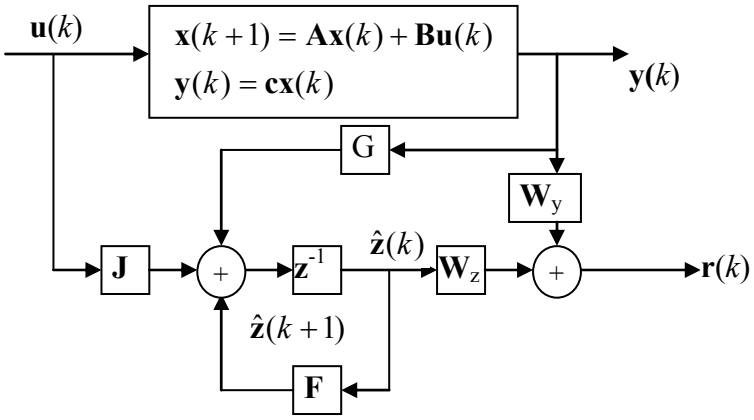


Figure 3-3: Process and output observer

The residual can be designed such that they are independent of the unknown input, for example disturbance, and of the state $\mathbf{x}(k)$ and $\mathbf{u}$ by special selection of $\mathbf{W}_z$ and $\mathbf{W}_y$.

$$\mathbf{r}(k) = \mathbf{W}_z\hat{\mathbf{z}}(k) + \mathbf{W}_y\mathbf{y}(k) \tag{3.11}$$

subject to structural conditions:

$$\begin{aligned} &\mathbf{TA} - \mathbf{FT} = \mathbf{GC} \\ &\mathbf{W}_z\mathbf{T} - \mathbf{W}_y\mathbf{C} = 0 \\ &\mathbf{J} = \mathbf{TB} \\ &\mathbf{W}_z \neq 0 \\ &\mathbf{F} \text{ be stable} \end{aligned} \tag{3.12}$$

In this way, the residual is dependent only on fault signals $\mathbf{f}_a$ and $\mathbf{f}_y$ [7], [10], [41]. However, all process model matrices must be known precisely.

3.2.2.2 Fault Detection with Parity Equations

The basic idea of parity relations approach is to provide a proper check of the parity (consistency) of the measurements acquired from the monitored system.

A straightforward model-based method of fault detection is to take a model $\mathbf{G}_m(z) = \dfrac{\hat{\mathbf{A}}(z)}{\hat{\mathbf{B}}(z)}$ and to run it in parallel to the process described by $\mathbf{G}_p(z) = \dfrac{\mathbf{A}(z)}{\mathbf{B}(z)}$, thereby forming an error vector $\mathbf{r}(z)$

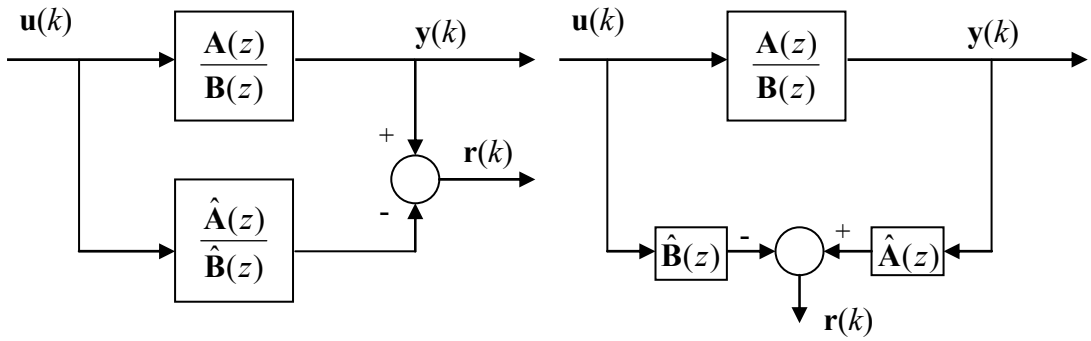
$$\mathbf{r}(z) = \left(\frac{\mathbf{A}(z)}{\mathbf{B}(z)} - \frac{\hat{\mathbf{A}}(z)}{\hat{\mathbf{B}}(z)} \right) \mathbf{u}(z) \tag{3.13}$$

If $\mathbf{G}_p$=$\mathbf{G}_m$, then the output error for additive input and output faults becomes

$$\mathbf{r}(z) = \mathbf{G}_p \mathbf{f}_a(z) + \mathbf{f}_y(z) \tag{3.14}$$

Another possibility to generate a polynomial error or equation error is as shown in Figure 3-5 [13].

$$\begin{aligned} \mathbf{r}(z) &= \hat{\mathbf{A}}(z)\mathbf{y}(z) - \hat{\mathbf{B}}(z)\mathbf{u}(z) \\ &= \mathbf{B}(z)\mathbf{f}_a(z) + \mathbf{A}(z)\mathbf{f}_y(z). \end{aligned} \tag{3.15}$$



(a) Output error (b) Equation error

Figure 3-4: Parity equation methods [13]

The residuals then depend only on the additive input faults $\mathbf{f}_a$ and output faults $\mathbf{f}_y$. Moreover, for the generation of specific characteristics of the parity vector **r**(z), and for obtaining fault detection and isolation properties, the residual can be filtered according to matrix $\mathbf{G}_f$(z) to compute the vector $\mathbf{r}_f(z)$ [7], [10], [8]:

$$\mathbf{r}_f(z) = \mathbf{G}_f \mathbf{r}(z) \tag{3.16}$$

The same procedure can be applied for multivariable processes by using the state space model, as shown in [8] for discrete time system.

$$\begin{aligned} \mathbf{x}(\mathrm{k}+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) \quad &= \mathbf{C}\mathbf{x}(k) \end{aligned} \tag{3.17}$$

By substituting the second of (3.17) in the first one and delaying several times, the following system is obtained:

$$\begin{bmatrix} \mathbf{y}(k) \\ \mathbf{y}(k+1) \\ \mathbf{y}(k+1) \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 & 0 & 0 & \cdots \\ \mathbf{CB} & 0 & 0 & \cdots \\ \mathbf{CAB} & \mathbf{CB} & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \mathbf{u}(k+2) \\ \vdots \end{bmatrix} \tag{3.18}$$

$$\mathbf{Y}_f(k) = \mathbf{T}\mathbf{x}(k) + \mathbf{Q}\mathbf{U}_f(k)$$

In order to remove the non-measurable states $\mathbf{x}(k)$, and to obtain a parity vector useful for FDI, a weighting matrix $\mathbf{W}$ is used, such that

$$\mathbf{WT}=0. \tag{3.19}$$

This lead to the residuals

$$\mathbf{r}(k) = \mathbf{W}\mathbf{Y}_f - \mathbf{W}\mathbf{Q}\mathbf{U}_f(k) \tag{3.20}$$

The design of the matrix $\mathbf{W}$ gives some freedom to generate a structured set of residuals in order to obtain a good isolation pattern. The parity space approach is suitable for the detection of additive faults. In addition, it is simpler to design and to implement than output observer-based approaches and lead approximately to the same results.

A comparison between observer-based and parity space techniques is gevin in [164].

3.2.2.3 Fault Detection with Parameter Estimation

In most practical cases, the process parameters are not known at all, or they are not known exactly enough. Therefore, they can be determined with parameter estimation methods, by measuring the input and output signals, $\mathbf{u}(k)$ and $\mathbf{y}(k)$, if the basic structure of the model is known [67], [7], [13], [8].

This approach is based on the assumption that the faults are reflected in the physical system parameters, and the basic idea is that the parameters of the actual process are estimated on-line using well-known parameter estimation methods. Two approaches for modelling the input-output behaviour of the system are used: minimization of equation error and output error.

The discrete-time model of order n for an SISO process is written in the vector form

$$y(t) = \mathbf{\Psi}^T\boldsymbol{\theta} \tag{3.21}$$

where $\boldsymbol{\theta} = [a_1 \ldots a_n, b_1 \ldots b_n]^T$ is the parameter vector of the transfer function $\frac{B(z)}{A(z)}$ and $\Psi = \left[y(t-1)\ldots y(t-n)\, u(t-1)\ldots u(t-n)\right]^T$ is the discrete-time data vector.

The equation error $e(t)$ is introduced as

$$e(t) = y(t) - \mathbf{\Psi}^T\boldsymbol{\theta} \tag{3.22}$$

The least-squares (LS) estimate of the parameters ($\hat{\boldsymbol{\theta}}$) is obtained from the minimization of the sum of squared error and

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\left(\min_{\hat{\boldsymbol{\theta}}} \sum_{l=0}^{N-1} \left\| y(t) - \boldsymbol{\Psi}^T \boldsymbol{\theta} \right\|_2^2 \right) \\ &= \left[\boldsymbol{\Psi}^T \boldsymbol{\Psi} \right]^{-1} \boldsymbol{\Psi}^T y\end{aligned} \tag{3.23}$$

As described in e.g., [7] and [66], the least-squares estimate can be also expressed in recursive form (RLS) with respect to the estimates at the instant *t*, with *t*=0,1,2,**...**

$$\hat{\boldsymbol{\theta}}(t+1) = \hat{\boldsymbol{\theta}}(t) + \gamma(t)[y(t+1) - \boldsymbol{\Psi}^T(t+1)\hat{\boldsymbol{\theta}}(t)] \tag{3.24}$$

where

$$\begin{aligned}\gamma(t) &= \frac{1}{\boldsymbol{\Psi}^T(t+1)\mathbf{P}(t)\boldsymbol{\Psi}(t+1)+1}\mathbf{P}(t)\boldsymbol{\Psi}(t+1) \\ \mathbf{P}(t+1) &= \left[I - \gamma(t)\boldsymbol{\Psi}^T(t+1) \right]\mathbf{P}(t)\end{aligned} \tag{3.25}$$

The results are thus compared with the parameters of the reference model; obtained initially under fault free assumptions. Any discrepancy can indicate that a fault may have occurred. The symptoms are the deviation of the process parameter, $\Delta\theta$:

$$\Delta\boldsymbol{\theta} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_o \tag{3.26}$$

where $\boldsymbol{\theta}_o$ is the nominal parameter, and $\hat{\boldsymbol{\theta}}$ is the estimated parameters.

As the process parameters $\boldsymbol{\theta} = f(\mathbf{p})$ depend on physically defined process coefficients $\mathbf{p}$ (like stiffness, resistance, etc.), the determination of the changes $\Delta\mathbf{p}$ allows usually a deeper insight and makes fault diagnosis easier [8], [7]. Parameter estimation methods usually need a process input excitation, and they are especially suitable for the detection of the multiplicative faults.

3.2.3 Disturbance, Noise and Uncertainties Modeling

The disturbance and noise of the system can be represented in the system model, in most cases, as additional unknown inputs with a specific distribution, while the uncertainties can be represented as unknown parameters. Therefore, the system model with disturbance, noise, and parameter uncertainties can be represent in a discrete linear model as:

$$\begin{aligned}\mathbf{x}(k+1) &= \big(\mathbf{A}+\Delta\mathbf{A}(k)\big)\mathbf{x}(k)+\big(\mathbf{B}+\Delta\mathbf{B}(k)\big)\mathbf{u}(k)+\mathbf{E}_a\mathbf{d}_a(k)+\boldsymbol{\nu}_x\\ \mathbf{y}(k) &= \big(\mathbf{C}+\Delta\mathbf{C}(k)\big)\mathbf{x}(k)+\mathbf{E}_y\mathbf{d}_y(k)+\boldsymbol{\nu}_y\end{aligned} \tag{3.27}$$

where ΔA, ΔB, and ΔC are the parameters uncertainties, ν_x and ν_y are the state and output noise respectively, $\mathbf{d}_a$ and $\mathbf{d}_y$ are the disturbance component in the state and output respectively, and $\mathbf{E}_a$ and $\mathbf{E}_y$ are the distribution matrices of disturbance with the appropriate dimension, which is assumed to be known.

3.2.4 Problem Formulation

In order to summarize the robustness problem, the complete state space model with faults, disturbances, uncertainties, and noise is represented by combining (3.1) and (3.27) as

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k)+\mathbf{B}\mathbf{u}(k)+\big(\mathbf{A}_f(k)+\Delta\mathbf{A}(k)\big)\mathbf{x}(k)\\ &\quad+\big(\mathbf{B}_f(k)+\Delta\mathbf{B}(k)\big)\mathbf{u}(k)+\mathbf{R}_a\mathbf{f}_a(k)+\mathbf{E}_a\mathbf{d}_a(k)+\boldsymbol{\nu}_x\\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)+\big(\mathbf{C}_f(k)+\Delta\mathbf{C}(k)\big)\mathbf{x}(k)+\mathbf{R}_y\mathbf{f}_y+\mathbf{E}_y\mathbf{d}_y(k)+\boldsymbol{\nu}_y\end{aligned} \tag{3.28}$$

Equation (3.28) can be written in general form for different types of systems, weither linear or nonlinear, as

$$\begin{aligned}\mathbf{x}(k+1) &= g(\theta,\mathbf{x},\mathbf{u},\mathbf{f},\mathbf{d},\boldsymbol{\nu})\\ \mathbf{y} &= h(\theta,\mathbf{x},\mathbf{u},\mathbf{f},\mathbf{d},\boldsymbol{\nu})\end{aligned} \tag{3.29}$$

where $\theta=\theta_0 \cup \theta_f \cup \Delta\theta$, θ_0 is the nominal parameter space, θ_f is the faulty parameter space, and $\Delta\theta$ is the uncertainty parameter space of the system; $\mathbf{f}=[\mathbf{f}_a\ \mathbf{f}_y]^T \in \Re^f$ is the total additive fault vector, $\mathbf{d}=[\mathbf{d}_a\ \mathbf{d}_y]^T \in \Re^d$ is the total disturbance vector, and $\boldsymbol{\nu}=[\boldsymbol{\nu}_x\ \boldsymbol{\nu}_y]^T \in \Re^{n+m}$ is the total noise vector; g: $\Re^n\times\Re^m\times\Re^f\times\Re^d\times\Re^{n+m}\rightarrow\Re^n$ and h: $\Re^n\times\Re^m\times\Re^f\times\Re^d\times\Re^{n+m}\rightarrow\Re^m$. For the system (3.28),

$\theta_0=\{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{E}_a,\mathbf{E}_y\}$, $\theta_f=\{\mathbf{A}_f,\mathbf{B}_f,\mathbf{C}_f,\mathbf{R}_a,\mathbf{R}_f\}$, and $\Delta\theta=\{\Delta\mathbf{A}, \Delta\mathbf{B}, \Delta\mathbf{C}\}$.

It is required to design a FDI system, which is sensitive to system faults and less sensitive to system disturbances, uncertainties, and noise with respect to the detected features.

It is clear that the uncertainties in the model parameters seem to be as nonlinear terms in the system model (3.28), and that is one of the difficulties of the robust FDI system.

By neglecting the noise in (3.28), the discrete transfer matrix description between the output $\mathbf{y}(k)$ and the input $\mathbf{u}(k)$ of the system (3.28) is then

$$\mathbf{y}(z) = \left(\mathbf{G}_u(z) + \Delta\mathbf{G}_u(z)\right)\mathbf{u}(z) + \mathbf{G}_d(z)\mathbf{d}(z) + \mathbf{G}_f(z)\mathbf{f}(z) \tag{3.30}$$

where $\mathbf{f}(z)=[\mathbf{f}_a(z)\ \ \mathbf{f}_y(z)]^T$ is the total additive fault vector, $\mathbf{d}(z)=[\mathbf{d}_a(z)\ \ \mathbf{d}_y(z)]^T$ is the total disturbance vector, $\Delta\mathbf{G}_u(z)$ is used to describe modelling errors, whilst both $\Delta\mathbf{G}_u$ and $\Delta\mathbf{G}_d$ represent modelling uncertainty.

According to residual generator, general structure described in [8], [37] and [108],

$$\mathbf{r}(z)=\mathbf{H}_u(z)\ \mathbf{u}(z)+\mathbf{H}_y(z)\ \mathbf{y}(z) \tag{3.31}$$

In case of no fault and uncertainties, the design of H_u and H_y must satisfy the constraints condition

$$\mathbf{H}_u(z)+\ \mathbf{H}_y(z)\ \mathbf{G}_u=0 \tag{3.32}$$

The residual vector in case of fault and uncertainties has to be written as

$$\mathbf{r}(z) = \mathbf{H}_y(z)\mathbf{G}_f(z)\mathbf{f}(z) + \mathbf{H}_y(z)\mathbf{G}_d(z)d(z) + \mathbf{H}_y(z)\Delta\mathbf{G}_u(z)\mathbf{u}(z) \tag{3.33}$$

Both faults and modeling uncertainties (disturbance and modeling error) affect the residual, and hence discrimination between these two effects is difficult. The principle of disturbance de-coupling for robust residual generation requires that the residual generator satisfy

$$\mathbf{H}_y(z)\mathbf{G}_d(z) = 0 \tag{3.34}$$

in order to achieve total de-coupling between residual $\mathbf{r}(z)$ and disturbance $\mathbf{d}(z)$.

During the last decades, many FDI researches have focused on robust fault diagnosis of an uncertain system. Adaptive threshold can be used to increase the robustness to modeling uncertainties [79], [8]. Surveys of adaptive threshold technique are provided in [37]. This method represents a passive approach since no effort is made to design robust residual. One of the most successful robust FDI approaches is the use of disturbance decoupling principle. This can be done by using unknown input observers [7], [165], [41], [167], optimal (robust) parity relations [58], [7], [166] or alternatively EA approach [7], [34], [37], [35]. However, the complete elimination of disturbance effect may be not possible due to the lack of degree of freedom [13]. In addition, in some cases, such as unstructured uncertainties or structured uncertainties, which do not enter the system as an additive disturbance, perfect decoupling is not possible [80].

Moreover, it may be problematic, in some cases, because the fault effect may also be eliminated. Hence, an appropriate criterion for robust residual design should take into accounts both modeling error and faults. There is a trade-off between sensitivity to faults and robustness to modeling uncertainty, and hence robust residual generation can be considered as a multi-objective optimization problem [78]. It consists of the maximization of fault effects and the minimization of uncertainty effects. Despite the extensively study of decoupling method in robust FDI system, their effectiveness regarding real problems has not been fully demonstrated. The described method of disturbance decoupling methods cannot be directly applied to the system with other uncertainties such as modeling error [13]. Different robust FDI techniques are scattered in the literature, see for example [81]-[88].

Although the analytical redundancy method for residual generation has been recognised as an effective technique for detecting and isolating faults, the critical problem of unavoidable modelling uncertainty has not been fully solved [13].

3.3 Multi-Model Fault Detection and Isolation System

Since failures in systems may cause structure change, the system cannot be modeled well by a single model. Moreover, accurate fault identification in favor of complete isolation cannot usually be achieved using a single model. One of the most effective approaches for such problems is based on the use of Multiple Models (MMs). It runs a bank of filters in parallel, each based on a model matching to a particular mode (i.e. structure or behavior pattern) of the system. Since a system subject to failures is a typical hybrid system [168], MM algorithms for FDI have been developed for different names, such as multiple hypothesis test detector [162], structure hypothesis test [34] and Multiple Model Adaptive Estimator (MMAE) algorithm [169], [170], [49], [50]. In addition, a so-called dedicated observer scheme, which uses a bank of observers for FDI of deterministic system, was devised in [163] and a generalized dedicated observer to enhance the robustness of FDI was given in [53]. A neural network bank-based FDI approach was developed in [171]. Only filter-based approaches are considered in the above approaches to estimate system state. The above filter-based approaches are based on the "non-interacting" MM method originally proposed by Magill [172]: the single-model based filters are running in parallel with out mutual interacting (i.e. each filter operates independently at all time). An Interacting Multiple-Model (IMM) estimator for

FDI was introduced in [108]. The IMM differs from non-interacting MM algorithm in that the single-model based filters interact with each other in highly cost-effective fashion, and thus leads to significantly improved performance.

Figure 3-5 shows the general diagram for MM-based FDI approaches. The system is described by a set of model $M=\{M_0, M_{1...} M_z\}$, and each model is designed to distinguish one fault mode of the fault mode set.

All fault detection and estimators (FDE) are driven by the system input **u** and the measurements **y,** and they operate in parallel to generate an individual residual for each one. All residuals and possibly all measurements are treated in the residual evaluation logic. The resulting faults are reflected in the alarm signal in the decision statements S= $\{a_1, a_2...a_z\}$; $a_i \in \{0,1\}$.

The task of the diagnosis system is to generate a diagnosis statement S, which contains information about which fault models that can explain the behavior of the process.

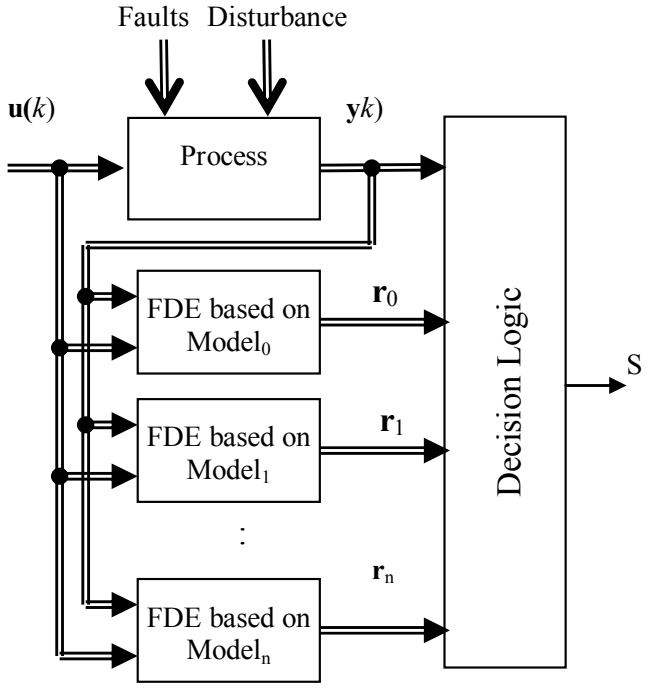


Figure 3-5: General block diagram of the MM fault detection scheme

For each actuator fault, an FDE unit can be an UIO as in the case of Generalized Observer Scheme (GOS) [7], [35]. The number of observers is equal to the number of the control inputs. The *i*-th observer is sensitive to all faults instead of *i*-th fault.

For a linear system with an additive actuator faults, the model can be defined as

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}[\mathbf{u}(k) + \mathbf{f}_a(k)] = \mathbf{A}\mathbf{x}(k) + \sum_{i=1}^{m} \mathbf{b}_i[u_i(k) + f_{ai}(k)] \\ \mathbf{y}(k) &= \mathbf{c}\mathbf{x}(k)\end{aligned} \tag{3.35}$$

The *i*-th model of actuator fault number *i* (*i*=1,2...*m*) has the form

$$\begin{aligned}\mathbf{x}^i(k+1) &= \mathbf{A}\mathbf{x}^i(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{b}_i f_{ai}(k) \\ \mathbf{y}^i(k) &= \mathbf{c}\mathbf{x}^i(k)\end{aligned} \tag{3.36}$$

and the corresponding observer

$$\begin{aligned}\mathbf{z}^i(k+1) &= \mathbf{F}^i\mathbf{z}^i(k) + \mathbf{J}^i\mathbf{u}(k) + \mathbf{G}^i\mathbf{y}(k) \\ \mathbf{r}^i(k) &= \mathbf{W}_z^i\mathbf{z}(k) + \mathbf{W}_y^i\mathbf{y}(k)\end{aligned} \tag{3.37}$$

where $\mathbf{x}^i$(k) is the state of the model *i*, the triple (**A**, **B**, **C**) is the system matrix, $\mathbf{b}_i$ is the *i*-th column of **B**, $\mathbf{z}^i(k) \in \Re^n$ denotes the observer state vector, $\mathbf{r}^i$ (k) $\in \Re^m$ is the residual vector and $\mathbf{F}^i$, $\mathbf{J}^i$, $\mathbf{G}^i$, $\mathbf{W}_z^i$ and $\mathbf{W}_y^i$ are matrices to be designed with appropriate dimensions which satisfy

$$\begin{aligned}&\mathbf{T}^i\mathbf{A} - \mathbf{F}^i\mathbf{T}^i = \mathbf{G}^i\mathbf{C} \\ &\mathbf{W}_z^i\mathbf{T}^i - \mathbf{W}_y^i\mathbf{C} = 0 \\ &\mathbf{J}^i = \mathbf{T}^i\mathbf{B}^i \\ &\mathbf{W}_z^i \neq 0 \\ &\mathbf{F}^i \text{ be stable}\end{aligned} \tag{3.38}$$

If the linear transformation $\mathbf{T}^i$ is chosen as [214]

$$\mathbf{T}^i = \mathbf{I}_n - \mathbf{b}_i(\mathbf{C}\mathbf{b}_i)^+\mathbf{C}$$

then the solution of (3.38) will be

$$\begin{aligned}&\mathbf{F}^i = \mathbf{T}^i\mathbf{A} - \mathbf{K}^i\mathbf{C}, \\ &\mathbf{G}^i = \mathbf{K}^i + \mathbf{F}^i\mathbf{b}_i(\mathbf{C}\mathbf{b}_i)^+, \\ &\mathbf{J}^i = \mathbf{T}^i\mathbf{B}^i \\ &\mathbf{W}_z^i = -\mathbf{C}, \\ &\mathbf{W}_y^i = \left[\mathbf{I}_m - (\mathbf{C}\mathbf{b}_i)(\mathbf{C}\mathbf{b}_i)^+\right]\end{aligned} \tag{3.39}$$

where $\mathbf{K}^i$ is selected such that $\mathbf{F}^i$ is asymptotically stable and $(\mathbf{C}\mathbf{b}_i)^+$is the pseudo inverse of matrix $\mathbf{C}\mathbf{b}_i$.

This selection makes the estimation error, as well as then the residual of the *i*-th UIO become independent of the *i*-th system input. However, the *i*-th input fault changes all the other UIO residuals.

For sensor faults, a set of dynamic observers as in the case of Dedicated Observer Scheme (DOS) [163] is designed. The *i*-th observer is designed where the estimated output error and then the residual of the *i*-th observer is dependent on the *i*-th sensor fault only.

3.4 Dynamic Safety Margin in Fault Diagnosis System

According to the previous discussion about robust FDI systems in section 3.2, we can conclude that:

1. The complete elimination of disturbances effect may not be possible;
2. Modelling uncertainties is difficult and has not been fully solved;
3. Most of the methods in literature try to reduce the effect of either disturbance or uncertainties, but not both;
4. The effectiveness of the existing robust FDI regarding real problems has not been fully demonstrated. In general, the robust FDI problem has not been fully solved.

This section explains how DSM can be helpful in designing a robust FDI system.

Assumption 3.1: Based on the definition of DSM in *Chapter 2*, the state variables of the diagnosed system ($\mathbf{x}$) and the associated estimated state from the nominal model ($\hat{\mathbf{x}}$) must satisfy $\{\mathbf{x}, \hat{\mathbf{x}}\} \in \Phi$ in normal operation, even if there exist bounded uncertainties, disturbances and/or noise, i.e. fault-free case.

where $\Phi \subseteq \Re^n = \{\phi_i(\mathbf{x}) \leq 0 \,|\, i=1...q\}$ is a compact set which contains the entire safe state variable.

According to the definition of DSM, it is positive in normal operation with probable parameters variation (uncertainties in system model) and/or disturbance. Otherwise, DSM is negative if the system suffers from a large variation in the parameters or a large disturbance, which simulate different types of faults (additive and/or multiplicative). The following example explains that.

Example 3.1

Consider a system having two state variables ($\mathbf{x}$=[x_1, x_2]), and the nominal relation between the two variables is

$$x_1(t)\text{-}a\,x_2(t)=b \tag{3.40}$$

where a and b are the nominal values of the system parameters.

If the probable change in the system parameter a is $\Delta a\in[-\alpha_1, \alpha_2]$ and the parameter b is constant, then the safe operation region can be defined as

$$\begin{aligned} &-x_1(t)+(\alpha-\alpha_1)\,x_2(t)\le \text{-}b \\ &x_1(t)+(\alpha+\alpha_2)\,x_2(t)\le b \end{aligned} \tag{3.41}$$

and δ(DSM) satisfies $\delta\ge 0 \leftrightarrow \Delta a\in[-\alpha_1, \alpha_2]$;

$$\Delta a\notin[-\alpha_1, \alpha_2] \vee \exists \text{fault}\rightarrow\delta<0$$

Any probable parameter variations or disturbances can be handled by the same way i.e. additional constrains can be added to increase the sensitivity of DSM to faults. Meanwhile, for larger systems the safe region is not readily obtained by this way.

For linear system with bounded disturbance and uncertainties the safe operation region can be considered as an invariant set, as stated in *Chapter 2*, and the methods of determine the invariant set , see for example [144], [148], [150], [158][148], [202], can be applied to construct the safe operation region.

If it is difficult to construct the safe operation region due to the less information about the system operation, using fuzzy or neural clustering is a helpful tool to determine the boundary of the safe region.

Theorem 3.1: Based on the state space model of (3.28), if there is no fault and $\mathbf{x}(k)\in\Phi$, then for any $\mathbf{x}_p\in\partial\Phi=\left\{\mathbf{x}_p \middle| \phi_i(\mathbf{x}_p)=0, i=1,2,...q\right\}$ the following conditions are satisfied

$$\begin{aligned} &\left(\delta(k)\ge 0\right)\wedge\left(\delta(k+1)\ge 0\right) \text{ and} \\ &\inf_{\mathbf{x}_p}\left\|\mathbf{x}_p-\mathbf{x}(k)\right\| \ge \sup_{\Delta\mathbf{A},\Delta\mathbf{B},\mathbf{d}}\left\|\left(\mathbf{A}+\Delta\mathbf{A}(k)-\mathbf{I}\right)\mathbf{x}(k)+\left(\mathbf{B}+\Delta\mathbf{B}(k)\right)\mathbf{u}(k)+\mathbf{E}\mathbf{d}(k)\right\| \end{aligned} \tag{3.42}$$

On the other hand, if the fault exists, then there are two recursive instants k and k+1such that

$$\inf_{\mathbf{x}_p}\left\|\mathbf{x}_p-\mathbf{x}(k)\right\| < \left\|\mathbf{x}(k+1)-\mathbf{x}(k)\right\| \tag{3.43}$$

where $\partial\Phi \subset \Phi$ is the boundary state set of Φ, and $\mathbf{x}(k)$ is the current state vector.

Proof: Based on the *Assumption* 3.1 and taking into consideration that all state variables are measurable, then the maximum effect of modelling uncertainties and disturbances together without faults should maintain $\mathbf{x}(k+1) \in \Phi$, which implies

$$\big(\delta(k) \geq 0\big) \wedge \big(\delta(k+1) \geq 0\big) \text{ (first condition of (3.42))}$$

and

$$\mathbf{x}(k+1) = (\mathbf{A} + \Delta\mathbf{A}(k))\mathbf{x}(k) + (\mathbf{B} + \Delta\mathbf{B}(k))\mathbf{u}(k) + \mathbf{E}\mathbf{d}(k) \in \Phi \tag{3.44}$$

The difference between the current state and the next state is

$$\begin{aligned} \mathbf{x}(k+1) - \mathbf{x}(k) &= \big(\mathbf{A} + \Delta\mathbf{A}(k)\big)\mathbf{x}(k) + \big(\mathbf{B} + \Delta\mathbf{B}(k)\big)\mathbf{u}(k) + \mathbf{E}\mathbf{d}(k) - \mathbf{x}(k) \\ &= \big(\mathbf{A} + \Delta\mathbf{A}(k) - \mathbf{I}\big)\mathbf{x}(k) + \big(\mathbf{B} + \Delta\mathbf{B}(k)\big)\mathbf{u}(k) + \mathbf{E}\mathbf{d}(k) \end{aligned} \tag{3.45}$$

The value of this difference varies due to the variation of the system parameters and current disturbance. Therefore, the maximum distance between the current and next state is

$$\big\|\mathbf{x}(k+1) - \mathbf{x}(k)\big\|_{max} = \sup_{\Delta\mathbf{A},\Delta\mathbf{B},\mathbf{d}} \big\|\big(\mathbf{A} + \Delta\mathbf{A}(k) - \mathbf{I}\big)\mathbf{x}(k) + \big(\mathbf{B} + \Delta\mathbf{B}(k)\big)\mathbf{u}(k) + \mathbf{E}\mathbf{d}(k)\big\| \tag{3.46}$$

Consequently, the maximum effect of the combined disturbance and modeling uncertainties makes $\mathbf{x}(k+1)$ tends to $\partial\Phi$ i.e.

$$\inf_{\mathbf{x}_p} \big\|\mathbf{x}_p - \mathbf{x}(k)\big\| \geq \big\|\mathbf{x}(k+1) - \mathbf{x}(k)\big\|_{\max} \tag{3.47}$$

Hence, from (3.46) and (3.47) the second condition of (3.42) is satisfied, which means that the distance between the state vectors from instant k to k+1 should be smaller than the minimum distance between the current state and $\partial\Phi$.

If there is a fault, then the state trajectory traverses outside the safety boundary. Therefore, there are two recursive instances k and k+1 where $\mathbf{x}(k) \in \Phi$ and $\mathbf{x}(k+1) \notin \Phi$ i.e. there is an $\mathbf{x}_p \in X_p$ that lies on the line between $\mathbf{x}(k)$ and $\mathbf{x}(k+1)$. This implies that the distance between $\mathbf{x}(k)$ and $\mathbf{x}_p$ is smaller than the distance between $\mathbf{x}(k)$ and $\mathbf{x}(k+1)$ (second part of *Theorem 3.1* (3.43)).

Thus, the sign of DSM is sensitive to faults. Moreover, the value of DSM itself is an indication of the hazards of faults. Hazards mean how much the fault can lead to a component failure or the damage of the process.

3.4.1 Fault Isolation

It is not always sufficient to indicate that a fault occurred, but it is more important to know which fault or faults have occurred (fault isolation), fault size, location, etc...

The suggested method for fault isolation depends on the generation of DSM from a set of models $M = \{M_0, M_1, M_z\}$, based on the idea of MM-FDI method discussed before, for the system under consideration (analytical redundancy of DSM) and the comparison of the generated DSM with the actual value calculated from the measured state. Each of these models simulates one fault of the faults set, which should be isolated in addition to the nominal fault free model as shown in Figure 3-6.

The discrete model of each faulty model is described in general as

$$M_i : \begin{cases} \mathbf{x}_i(k+1) = g_i\left(\theta_i(k), \mathbf{x}_i(k), \mathbf{u}(k), \mathbf{f}_i(k)\right) \\ \mathrm{y_i} = h_\mathrm{i}\left(\theta_i(\mathrm{k}), \mathbf{x}_i(\mathrm{k}), \mathbf{u}(\mathrm{k}), \mathbf{f}_\mathrm{i}(k)\right) \end{cases} \tag{3.48}$$

where $\mathbf{x}_i \in \Re^n$ is the state vector of the system model i, $\mathbf{u} \in \Re^m$ is the input vector, $\mathbf{y}_i \in \Re^p$ is output vector; $\mathbf{f}_i \subseteq \mathbf{F} \in \Re^l$ is unknown additive fault signal vector, g_i: $\Re^n \times \Re^m \times \Re^l \to \Re^n$, h_i: $\Re^n \times \Re^m \times \Re^l \to \Re^p$, $\theta_i \subset \theta$ is the system parameters for faulty model i, $i \in \{0,1,...,z\}$, and z is the number of anticipated faults in addition to fault free case, i=0, nominal model.

In case of LTI system

$$\begin{aligned} g_i &= \mathbf{A}_i \mathbf{x}_i(k) + \mathbf{B}_i \mathbf{u}(k) + \mathbf{R}_{ai} \mathbf{f}_i(k) \\ h_i &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}_i \mathbf{u}(k) + \mathbf{R}_{si} \mathbf{f}_i(k) \end{aligned} \tag{3.49}$$

and $\theta_i = \{\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i, \mathbf{D}_i, \mathbf{R}_{ai}, \mathbf{R}_{si},\}$

The fault isolation system is activated when $\delta(\mathrm{t}) < 0$ and/or $d\delta(\mathrm{t})/dt < 0$. $d\delta(\mathrm{t})/dt < 0$ means that the state trajectory moves in the direction of unsafe operation.

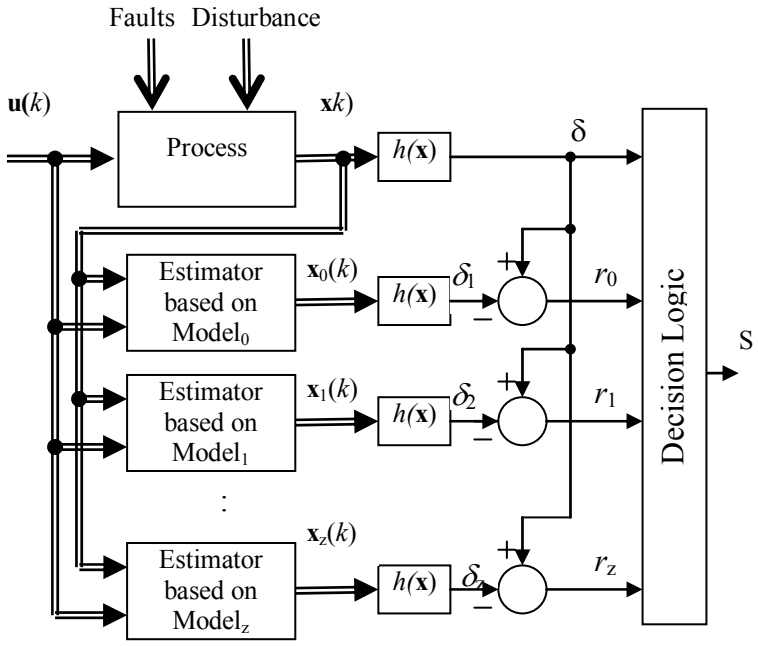


Figure 3-6: MM fault detection based on DSM

3.4.1.1 Fault Modes

Different faults can be classified into different fault modes. For example, consider a system containing a water tank and leakages exist in the bottom of this tank. All such leakages, regardless of their area, belong to the same fault mode “water tank bottom leakage" [34].

The classification of different faults into fault modes corresponds to a partition of the fault-parameter space θ and additive fault space F. This means that each fault mode i is associated with a subset $\theta_i \subseteq \theta$ and $\mathbf{f}_i \subseteq F$. One of the fault modes corresponds to the fault-free case. This fault mode will be denoted “no fault" or M_o. Further, all sets θ_i and f_i are pair wise disjoint and

$$\theta = \bigcup_{i \in \Omega} \theta_i \text{ and } F = \bigcup_{i \in \Omega} \mathbf{f}_i \tag{3.50}$$

where Ω is used to denote the set of all fault modes.

Let $\Sigma = \theta \bigcup F$ is the total fault mode data, $\Sigma_i = \theta_i \bigcup F_i$ is the fault mode i data, and $\Sigma = \bigcup_{i \in \Omega} \Sigma_i$

If fault mode i exists in the system, then $\Sigma_i \in \Sigma$. The fact that all sets Σ_i are pair wise disjoint means that only one fault mode can exist at the same time.

For notational convenience to each fault mode, an abbreviation is associated, e.g. “no fault" is abbreviated M_0. All these are is illustrated in Figure 3-7, which shows how the whole set Σ has been divided into five subsets corresponding to fault modes M_0, M_1, M_2, M_3, and M_4.

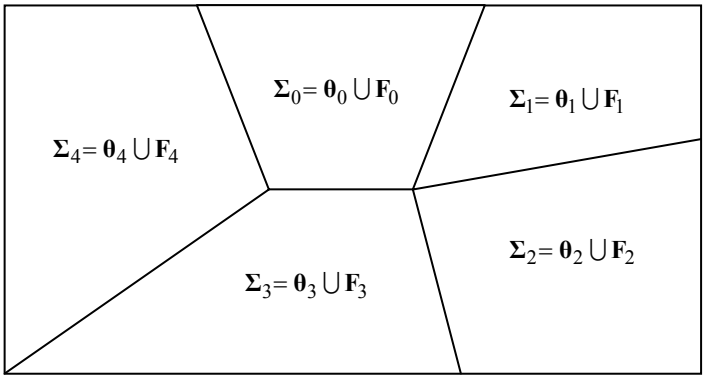


Figure 3-7: Parameter space and fault space divided into convex subspace

Example 3.2

Consider a SISO system represented in state space form (3.3), and assume that there are three different faults, which have to be detected and isolated; actuator fault, internal (leakage) fault and level sensor fault. Each of the actuator and the sensor fault has two types of faults, either bias or draft. Therefore, there are five modes of faults in addition to the fault-free case (3.3).

1. The actuator bias fault model is defined as

$$\begin{aligned} \mathbf{x}_1(k+1) &= \mathbf{A}_1\ \mathbf{x}_1(k) + \mathbf{b}_1\ (\ u(k) + f_a(k)) \\ y &= \mathbf{c}_1\ \mathbf{x}_1(k) \end{aligned} \tag{3.51}$$

$$\theta_1=\{\mathbf{A}_1, \mathbf{b}_1, \mathbf{c}_1\},\ \mathbf{f}_1=[f_a\ 0\ 0]^{\mathrm{T}}$$

where $\mathbf{A}_1=\mathbf{A}$, $\mathbf{b}_1=\mathbf{b}$, and $\mathbf{c}_1=\mathbf{c}$, and f_a is the additive actuator bias.

2. The actuator draft fault model is defined as

$$\begin{aligned} \mathbf{x}_2(k+1) &= \mathbf{A}_2\ \mathbf{x}_2 u(k) + \mathbf{b}_2\ (\alpha\ u(k)) \\ y &= \mathbf{c}_2\ \mathbf{x}_2 u(k) \end{aligned} \tag{3.52}$$

$$\theta_2=\{\mathbf{A}_2, \mathbf{b}_2, \mathbf{c}_2, \alpha\},\ \mathbf{f}_2=[0\ 0\ 0]^{\mathrm{T}}$$

where $\mathbf{A}_2=\mathbf{A}$, $\mathbf{b}_2=\mathbf{b}$, and $\mathbf{c}_2=\mathbf{c}$, and α is the actuator draft.

3. The leakage fault model can be represented by two different methods; one of them can be represented as unknown parameters in **A** matrix (multiplicative fault). The other method, which is described here, represents the leakage fault as an additional signal in the state model.

$$\begin{aligned} \mathbf{x}_3(k+1) &= \mathbf{A}_3\mathbf{x}_3(k) + \mathbf{b}_3\, u(k) + b_a f_i(k) \\ y &= \mathbf{c}_3\, \mathbf{x}_3(k) \end{aligned} \tag{3.53}$$

$$\theta_3=\{\mathbf{A}_3, \mathbf{b}_3, \mathbf{c}_3, \mathbf{b}_a\},\ \mathbf{f}_3=[0\ f_i\ 0]^{\mathrm{T}}$$

where $\mathbf{A}_3=\mathbf{A}$, $\mathbf{b}_3=\mathbf{b}$, and $\mathbf{c}_3=\mathbf{c}$, $\mathbf{b}_a$ is leakage fault distribution matrix with appropriate dimension, f_i is the additive signal represent internal leakage.

4. The sensor bias fault model is

$$\begin{aligned} \mathbf{x}_4(k+1) &= \mathbf{A}_4\, \mathbf{x}_1(k) + \mathbf{b}_4\, u(k) \\ y &= \mathbf{c}_4\, \mathbf{x}_4(k) + f_{\mathrm{s}} \end{aligned} \tag{3.54}$$

$$\theta_4=\{\mathbf{A}_4, \mathbf{b}_4, \mathbf{c}_4),\ \mathbf{f}_3=[0\ 0\ f_{\mathrm{s}}]^{\mathrm{T}}$$

where $\mathbf{A}_4=\mathbf{A}$, $\mathbf{b}_4=\mathbf{b}$, and $\mathbf{c}_4=\mathbf{c}$, f_{s} is the additive signal represent sensor fault.

5. The sensor draft fault model is

$$\begin{aligned} \mathbf{x}_5(k+1) &= \mathbf{A}_5\, \mathbf{x}_1(k) + \mathbf{b}_5\, u(k) \\ y &= \alpha_{\mathrm{s}}\mathbf{c}_5\, \mathbf{x}_5(k)_{\mathrm{s}} \end{aligned} \tag{3.55}$$

$$\theta_5=\{\mathbf{A}_5, \mathbf{b}_5, \mathbf{c}_5, \alpha_{\mathrm{s}}),\ \mathbf{f}_5=[0\ 0\ 0]^{\mathrm{T}}$$

where $\mathbf{A}_5=\mathbf{A}$, $\mathbf{b}_5=\mathbf{b}$, and $\mathbf{c}_5=\mathbf{c}$, α is the sensor draft.

3.4.1.2 Fault Estimation

One possibility to simulate the fault value by generating a set of models for the same fault type, each of which for a certain partial value of fault [48]. For example, the fault in a sensor *i* can be modelled as

$$y_i(k)= y_{oi}(k)+a \tag{3.56}$$

where $a\in\{0,a_1,a_2\ldots\ y_{i\max}\}\in\Re^q$ represents a partial value sensor bias, $y_{i\max}$ is the max input limit of sensor *i*, and *q* is the number of partial values of sensor fault; y_{oi} is the measured output signal and y_i is the actual output signal of the sensor number *i.*

For each value of *a* there is a corresponding fault model. The number of models for bias fault mode is equal to *q*. The magnitude (size) of fault can be determined by the

probability-weighted sum of the fault magnitudes of the corresponding partial fault models [48]. However, the main disadvantage of this approach is that the number of model increases proportionally with the resolution of fault discrimination.

Another method of fault isolation is by estimating the fault magnitude from the measured data; see for example [34].

The suggested method is based on the estimation of the fault values from the input-output data, and using these estimation parameters to determine the estimated state of the faulty model. For each faulty model the unknown fault parameter $\boldsymbol{\gamma}_i \in \boldsymbol{\Sigma}_I$, which simulates unknown multiplicative or unknown additive faults. $\boldsymbol{\gamma}_i$ can be estimated from

$$\hat{\boldsymbol{\gamma}}_i = \arg\left(\min_{\boldsymbol{\gamma}_i \in \Sigma_i} \sum_{l=0}^{N-1} \left\| (\mathbf{y}(k-l) - \hat{\mathbf{y}}_i(k-l) \right\|_2^2 \right) \tag{3.57}$$

where N is the estimation time.

The main disadvantage of this method is that the minimization problem (3.47) is not easy to solve, especially in case of multiplicative parameters.

Estimating the fault value from available redundant equations of the system model is considered a special case of estimation principle.

The following example describes how the unknown parameters can be estimated for a linear model with actuator fault.

Example 3.3

A state space model of a linear system with actuator faults can be defined as

$$\begin{aligned} &\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}[\mathbf{u}(k) + \mathbf{f}_a(k)] = \mathbf{A}\mathbf{x}(k) + \sum_{i=1}^{m} b_i [u_i(k) + f_{ai}(k)] \\ &y(k) = \mathbf{c}x(k) \end{aligned} \tag{3.58}$$

For a single actuator fault mode, the fault model will be

$$\begin{aligned} &\mathbf{x}_i(k+1) = \mathbf{A}\mathbf{x}_i(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{b}_i f_{ai}(k) \\ &\mathbf{y}_i(k) = \mathbf{c}\mathbf{x}_i(k) \end{aligned} \tag{3.59}$$

and the estimate actuator fault, $\hat{f}_{ai}$, according to (3.47) is given by

$$\hat{f}_{ai} = \arg(\min_{f_{ai}} \left(\left\| \mathbf{Y} - \mathbf{C}_a \mathbf{x}(k-N) - \mathbf{C}_b \mathbf{U} - \mathbf{C}_b f_{ai} \right\|_2^2 \right) \tag{3.60}$$

where $\mathbf{Y}=[\mathbf{y}(k)\ \mathbf{y}(k+1)\ \ldots\ \mathbf{y}(k+N)]^T$, $\mathbf{U}=[\mathbf{u}(k)\ \mathbf{u}(k+1)\ \ldots\ \mathbf{u}(k+N-1)]^T$, $\mathbf{C}_a$ and $\mathbf{C}_b$ matrices with appropriate dimension obtained from (3.48)

Special case

If the additive fault dynamics can be represented in state space form as

$$f_{ai}(k+1) = \alpha_i\ f_{ai}(k) + \beta_i \tag{3.61}$$

then the fault can be considered as an additional state variable, and it can be estimated using a state estimator as follows:

The new state vector $\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ f_{ai} \end{bmatrix}$, the input vector $\mathbf{u}_{fi} = \begin{bmatrix} \mathbf{u} \\ \beta_i \end{bmatrix}$, and the model

$$\begin{aligned} \mathbf{z}_i(k+1) &= \begin{bmatrix} \mathbf{A} & \mathbf{b}_i \\ 0 & \alpha_i \end{bmatrix} \mathbf{z}(k) + \begin{bmatrix} \mathbf{B} & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_{fi}(k) \\ y &= \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix} \mathbf{z}_i \end{aligned} \tag{3.62}$$

where α_i and β_i are assumed to be constant, for instance, α_i =1 and β_i=0 in case of bias fault. The new state space model has to be observable in order to estimate the state and fault value.

The estimated faulty model state, $\hat{\mathbf{x}}_i$, and the fault, $\hat{f}_{ai}$, are obtained using the state observer

$$\begin{aligned} \hat{\mathbf{z}}_i(k+1) &= \begin{bmatrix} \mathbf{A} & \mathbf{b}_i \\ 0 & \alpha_i \end{bmatrix} \hat{\mathbf{z}}(k) + \begin{bmatrix} \mathbf{B} & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_{fi}(k) + \mathbf{L}(\mathbf{y} - \hat{\mathbf{y}}) \\ \hat{\mathbf{y}} &= \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix} \hat{\mathbf{z}}_i \end{aligned} \tag{3.63}$$

where $\hat{\mathbf{z}}_i = \begin{bmatrix} \hat{\mathbf{x}}_i \\ \hat{f}_{ai} \end{bmatrix}$, $\hat{\mathbf{y}}$ is the estimated output, and $\mathbf{L}$ is the observer gain matrix.

This procedure was implemented using EKF for constant sensor and actuator fault, i.e. α_i =1 and β_i=0, in [49], [50]. The fault can also be reconstructed using the idea in [56].

3.4.1.3 Fault Isolation Algorithm

The estimated fault parameters for each fault model are used to estimate the DSM of each model. The estimated value of DSM is compared with the DSM of the actual system, as shown in Figure 3-6.

It is clear, as in Figure 3-6, that the difference signal, r_i, between the actual DSM and the estimated one can be considered as a residual. The fault isolation logic is obtained from the analysis of the residual vector $\mathbf{r}\in\Re^z$. The following algorithm is one of the simplest methods to isolate fault number 'i' from the other faults.

1. Compute DSM for the actual system and the different faulty model $\delta_i(k)=h(\mathbf{x}_i(\mathrm{k}))$; i=0,1,2,...,z
2. Compute

$$r_i(k)= \delta(k)- \delta_i(k);\ i=1,2,...,z \tag{3.64}$$

$$T_i=\frac{1}{N}\sum_{k=k_0}^{k_0+N}\left|r_i(k)\right|;\ i=1,2,...,z \tag{3.65}$$

3. Construct the decision logic statement

$$\mathrm{S}= \{a_1, a_2, ...,a_z\};\ a_i\in\{0,1\} \tag{3.66}$$

where $a_i= 1 \leftrightarrow T_i = \min_j T_j$, i.e. f_i (fault number i) exists; otherwise a_i=0, h(.) is the DSM computation function, δ(.) is the DSM of the actual system, δ_i(.) is the DSM of fault model number i, z is the number of faults in addition to the fault free case, and N is the number of diagnostic samples (diagnostic time and isolation).

The logic statement contains at most one element equal "1" which is corresponding to the existing fault "i", and the others are "0", i.e. fault "i" occurs if the average value of '$\left|r_i\right|$' has the minimum value among the others.

There are some limitations in applying this algorithm:

1. A false signal may be obtained if the diagnostic time is small with respect to the fault reaction time, or if there are more than one fault occurring at the same time. Using threshold can help solving these problems, and the decision statement elements can be calculated as

$$a_i= 1 \leftrightarrow T_i\le T_{th}(i),\ \text{i.e. } f_i \text{ exists; otherwise } a_i=0$$

$$\text{and } \mathbf{T}_{th}(i) \le \delta_{\max} \tag{3.67}$$

where $\mathbf{T}_{th} \in\Re^z$ is the threshold vector, $\mathbf{T}_{th}(i)$ is the threshold value corresponding to the fault number "i", and $\delta_{\max}$ is the maximum positive value of DSM in the safe operation mode. In general, $\mathbf{T}_{th}(i)$ can be constant or time varying. For simplicity, in most cases all $\mathbf{T}_{th}(i)$'s are constant and equal for all fault models.

2. If a particular fault exists and more than one fault mode can achieve the fault isolation condition ($T_i \leq T_{th}(i)$) i.e. $\sum_{i=1}^{z} a_i > 1$, then this problem can be reduced if the magnitudes of the estimated fault parameters are considered in the isolation logic.

Thus, some definitions and related conditions of fault delectability and isolability are defined in the following section.

3.4.2 Detectability and Isolability

Definition 3.2 (*Fault Detectability*): The fault is detectable if and only if the effect of the fault on the system state causes ($\delta(t) < 0$) $\vee$ ($\delta_0(t) < 0$), other wise the fault cannot be detected.

where $\delta(t)$ is the DSM computed from the measured state and $\delta_0(t)$ is the DSM computed from the estimated state of nominal model (fault-free model).

Definition 3.3 (*Fault Isolability*): the fault i is isolable from fault j if and only if $\lim_{t\to\infty}\left(\delta(t) - \delta_i(t)\right) \leq \mathbf{T}_{th}(i) \wedge \lim_{t\to\infty}\left(\delta(t) - \delta_j(t)\right) \geq \mathbf{T}_{th}(j)$

where $\delta(t)$ is the actual DSM and $\delta_i(t)$ and $\delta_j(t)$ are the estimated DSM from fault model i and j respectively.

Definition 3.4 (*Missed Detection*): Assume that a fault i is exist, then the fault detection represents a missed detection if $\delta(t) > 0 \wedge \delta_0(t) > 0$.

Definition 3.4 (*Missed Isolation*): Assume that a fault i is present. Then the diagnosis statement S represents a missed isolation if $S(i) \neq 1$.

Definition 3.5 (*False Alarm*)**:** Assume that no faults exist, i.e. $\boldsymbol{\theta}_0 = \theta_0 \wedge \mathbf{f}_0 = 0$. Then the diagnosis statement S represents a false alarm if $S(0) \neq 1$.

Definition 3.6 (*Complete Isolable*): A Fault i is completely isolable from the fault set z if and only if $\lim_{t\to\infty}\left(\delta(t) - \delta_i(t)\right) \leq T_{th} \wedge \left\{\hat{\boldsymbol{\theta}}, \hat{f}_i\right\} \in \boldsymbol{\Sigma}_i$.

where $\hat{\boldsymbol{\theta}}_i$ is the estimated parameter of fault mode i, and $\hat{f}_i$ is the estimated additive fault.

3.4.3 Robustness of Detection and Isolation System

The robust fault detection, as discussed before, means that the detection system should give an alarm signal in case of fault and avoid false alarm in case of system disturbance

and parameter variation. Using DSM as an indication to fault satisfies a good robustness of the monitored system because of the value of DSM is sensitive to fault based on the assumption that the system should operate in the safe region in fault-free case, in spite of the existence of disturbance or uncertainties in the system parameters. Moreover, most of the control systems try to maintain the desired system performance in normal, disturbed and/or uncertain case (robust controller) which means that the controller in fault-free case tries to maintain DSM positive. Adaptive threshold for FDI method is one of the robust fault detection methods. The DSM value can be considered as an adaptive threshold where the DSM is the distance between the system state and the nearest state lying on the safety boundary, which is the maximum system state in fault free case. The maximum state value that the system can reach is not fixed, but it varies according to the boundary function and current state position. This approach has a high robustness where it can give positive alarm, i.e. fault indication in most fault cases, while it gives a false alarm in limited cases e.g. a) if the fault effect is less than the effect of probable model uncertainties and/or disturbance; b) If there are simultaneous faults having an opposite effects.

Robust fault isolation means that the fault isolation data should represent the actual system fault. MM fault isolation method is one of the most appropriate robust fault isolation methods. The main advantage of the suggested method is that the fault type and its estimated value are obtained in one-step. Moreover, it is not restricted to a special type of faults i.e. the fault can be modeled by any way as an additional signal or parameter variation.

The advantage of using DSM in FDI instead of the outputs are:

- The number of variables used in diagnosis using DSM is less (i.e. the measured output date are reduced to a single variable).
- The value of $\delta(t)$ and $d\delta(t)/dt$ is more sensitive to the system variation.
- Slow faults (e.g. equipment weakling) are very hard to detect from the output and DSM helps in the diagnosis and prognosis of such types of faults.

3.4.4 Simulation Example

Consider the tank system in *Example* 2.2, the linearized nominal model is

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\,\mathbf{x}(k) + \mathbf{b}\,u(k) \\ y &= \mathbf{c}\,\mathbf{x}(k) \end{aligned} \tag{3.68}$$

and the safe region is defined as

$$\begin{aligned} dh/dt + 0.8\,v_i &\le 0 \\ dh/dt + 0.8\,v_i - 0.16 &\ge 0 \\ -0.4 \le\ dh/\mathrm{dt} &\le 0.4 \\ -0.5 \le\ v_i &\le 0.5 \\ 2.75 \le h &\le 3.25 \end{aligned} \tag{3.69}$$

where $\mathbf{A}=\begin{bmatrix} 0.999741 & -0.000694 \\ 0 & 0.740818 \end{bmatrix}$, $\mathbf{b}=\begin{bmatrix} -0.00010932 \\ 0.25918177 \end{bmatrix}$, $\mathbf{c}=\begin{bmatrix} 1 & 0 \end{bmatrix}$, $\mathbf{x}=\begin{bmatrix} h \\ v_i \end{bmatrix}$, h is the level in the tank (m) and v_i the valve limb movement (m).

Assume there are three different faults that should be detected and isolated; actuator bias fault $\in$ [0,1], internal (leakage) fault in (l/s) $\in$ [-0.5,0.5] and level sensor bias fault $\in$ [-0.35,0.35]. Therefore, there are three modes of faults in addition to the faulty free case (3.65). The parameters of each fault mode are obtained as in *Example* 3.2. In case of leakage fault, the system model as in (3.53) where $\mathbf{b}_a=[1\ 0]^T$.

The estimated state of the fault-free model is obtained using a state observer as shown in Figure 3-2, where $\mathbf{H}=\begin{bmatrix} 1.4 & -200 \end{bmatrix}^T$ and $\mathbf{W}$ is not considered here because the purpose is to estimate the state.

Figure 3-8 shows the level response and normalized DSM variation in case of actuator bias fault of 30% of the actuator limit after 200s. The DSM value is positive before the fault and negative after fault. Figure 3-9 shows the estimated fault value for each fault model. Note that the DSM value generated from the actuator fault model, sensor fault model, and actual DSM are coincident, but the value of estimated sensor fault is out of limits and increases with the time. Therefore, according to *Definition* 3.6 the actuator fault is completely isolable from the other faults. Figure 3-10 and Figure 3-12 show the system response and DSM variation for leakage fault of 0.25 l/s and sensor fault 0.05m respectively. Figure 3-11 and Figure 3-13 show the estimated fault values in leakage and sensor fault respectively. Table 3-1 summaries the results of the three fault scenarios.

Figure 3-14 shows the response in case of actuator bias fault 30% after 200s in addition to additive disturbance with frequency 0.1Hz and amplitude 0.0001. Figure 3-15 shows the estimated fault values. It is clear that DSM value is positive in case of disturbance while it is negative in case of faults occur. Moreover, the effect of disturbance *i* is reflected on the estimated fault value.

Table 3-1: Summary of fault results

	$\hat{f}_a$	$\hat{f}_I$	$\hat{f}_s$	T_0	T_1	T_2	T_3
$f_a = 0.3$	0.3	0.16	ramp	$> T_{th}$	$\leq T_{th}$	$> T_{th}$	$\leq T_{th}$
$f_I = -0.25$	0.37	-0.25	ramp	$> T_{th}$	$> T_{th}$	$\leq T_{th}$	$\leq T_{th}$
$f_s = -0.05$	0.02	0	-0.05	$> T_{th}$	$> T_{th}$	$> T_{th}$	$\leq T_{th}$

Note: f_a, f_l, and f_s are the actuator, leakage and sensor faults respectively; $\hat{f}_a$, $\hat{f}_a$ and $\hat{f}_a$ are the estimated ones. T_i is the integral error between actual DSM and the computed one from each faulty model (3.37), $i \in \{0,1,2,3\}$ is the fault mode, and T_{th}=0.1 is a threshold error with integration step N=10.

3.5 Conclusions

In this chapter, the robust FDI problem is defined, and the existing techniques to design a robust FDI system and their limitations are discussed. Design FDI system based on DSM is introduced. The main advantage of the proposed approach of using DSM in FDI is the reduction of the number of diagnostic variables. In addition, DSM and its derivative are more sensitive to system parameter variation, i.e. DSM in FDI introduces robust fault-detection schemes. The simulation results demonstrate the advantage of this approach.

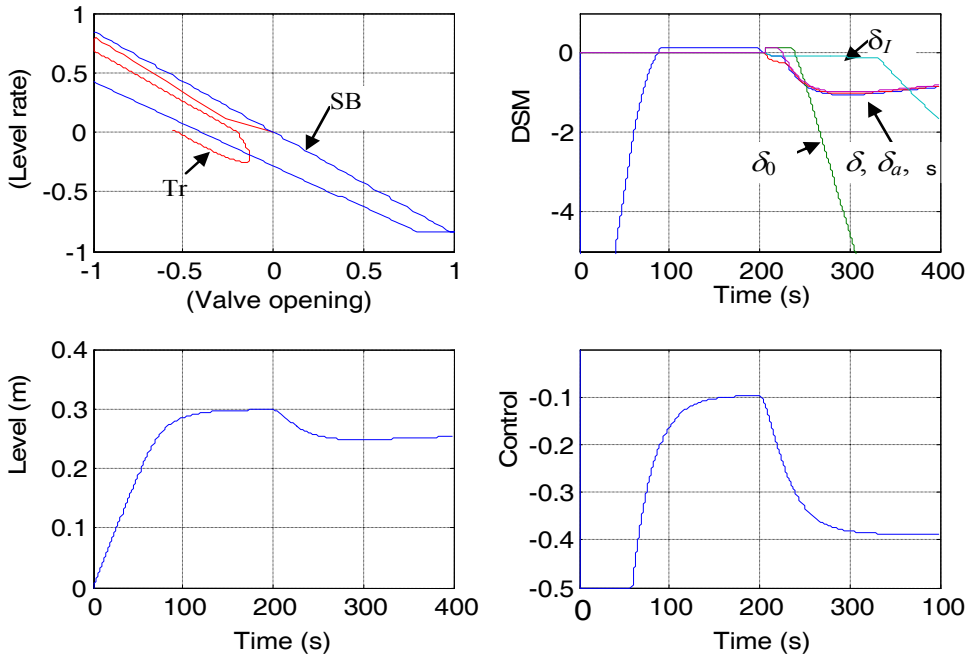


Figure 3-8: Actuator fault response

Tr: state trajectory; SB: Safety boundary; δ, δ_0, δ_I, δ_a , and δ_s are the DSM's of the actual system, nominal model, internal fault model, actuator fault model, and sensor fault model respectively.

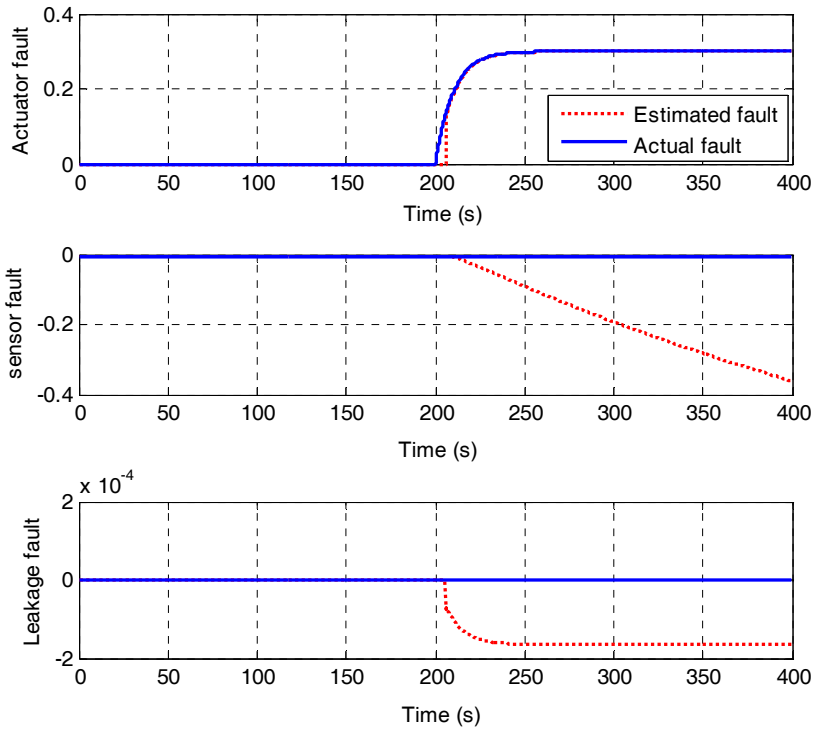


Figure 3-9: Estimation of faults in case of the actuator fault

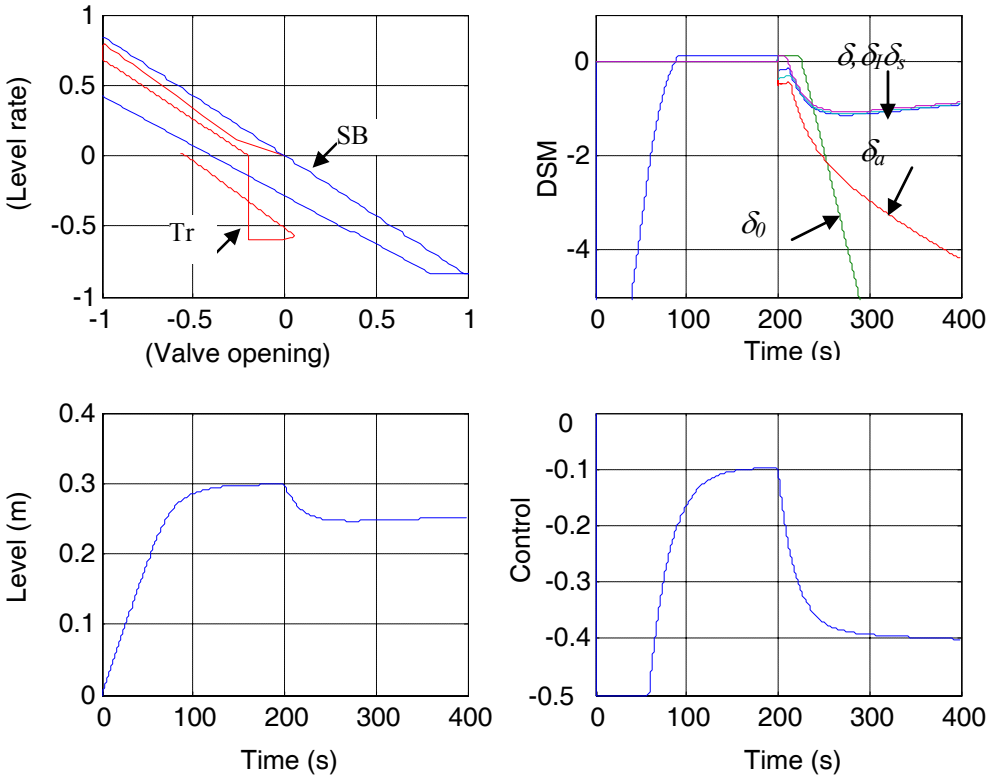


Figure 3-10: Leakage fault response

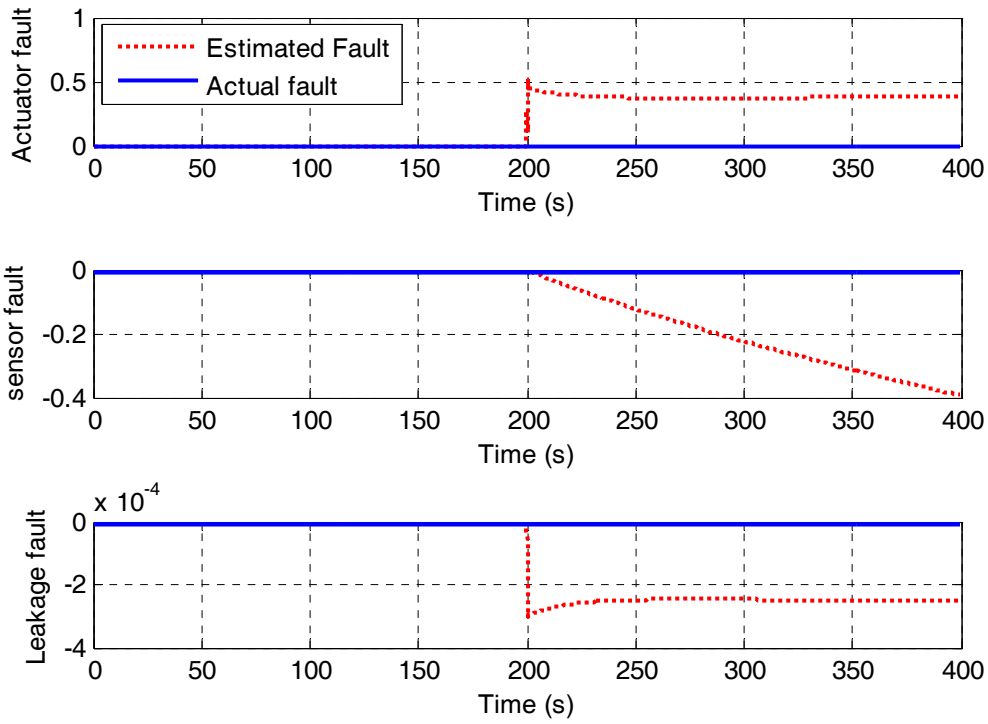


Figure 3-11: Estimation of faults in case of the leakage fault

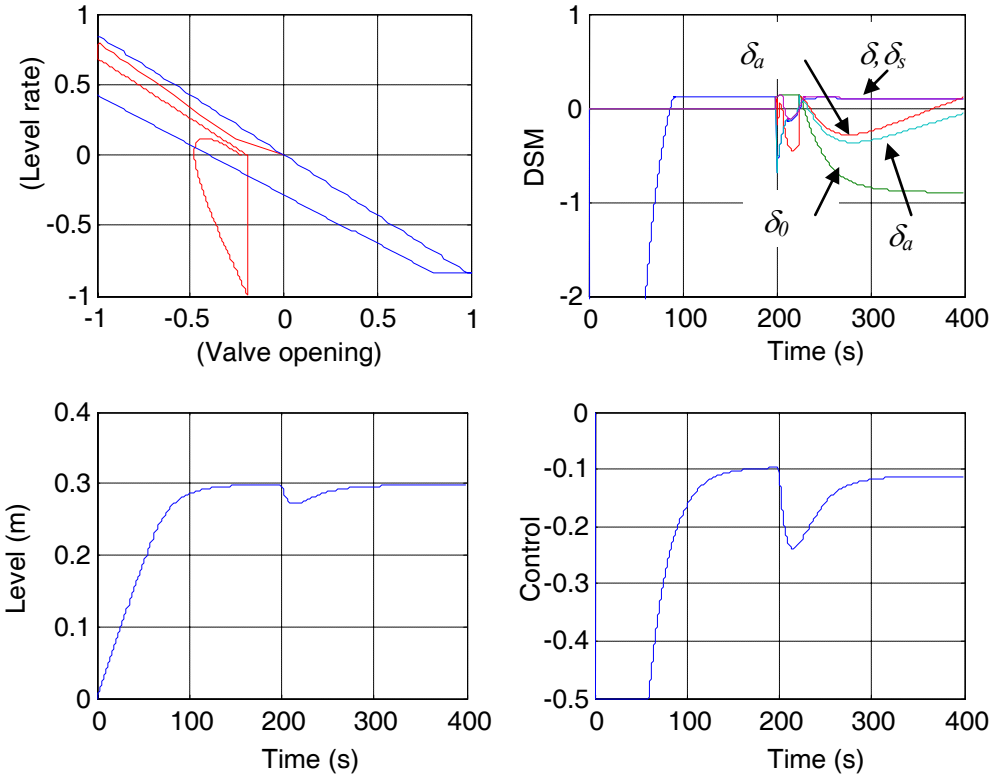


Figure 3-12: Sensor fault response

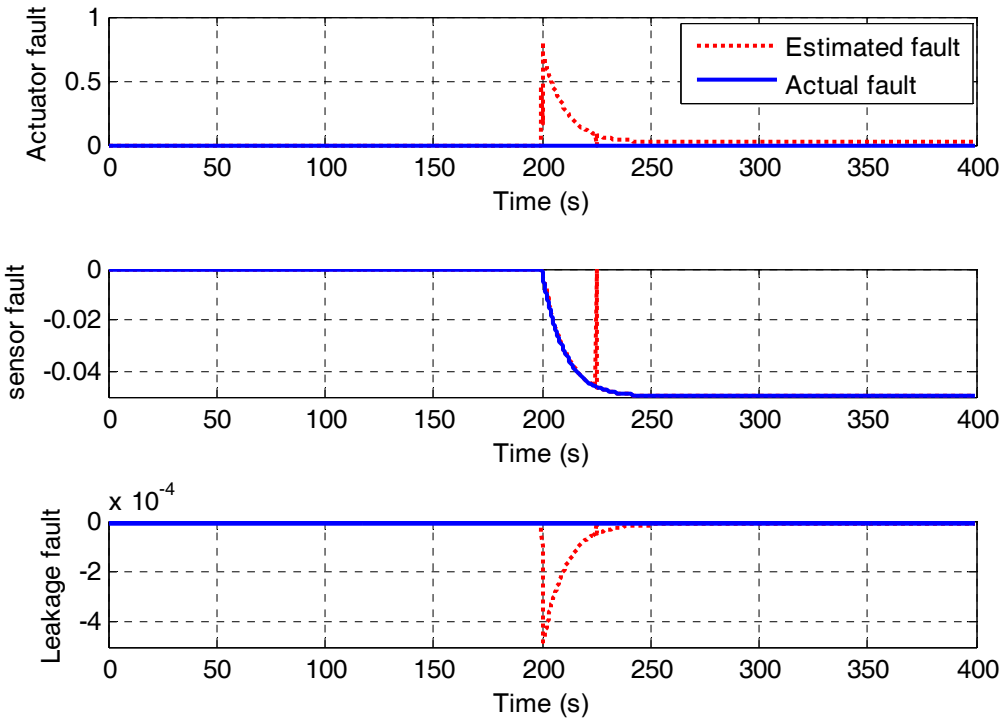


Figure 3-13: Estimation of faults in case of the sensor fault

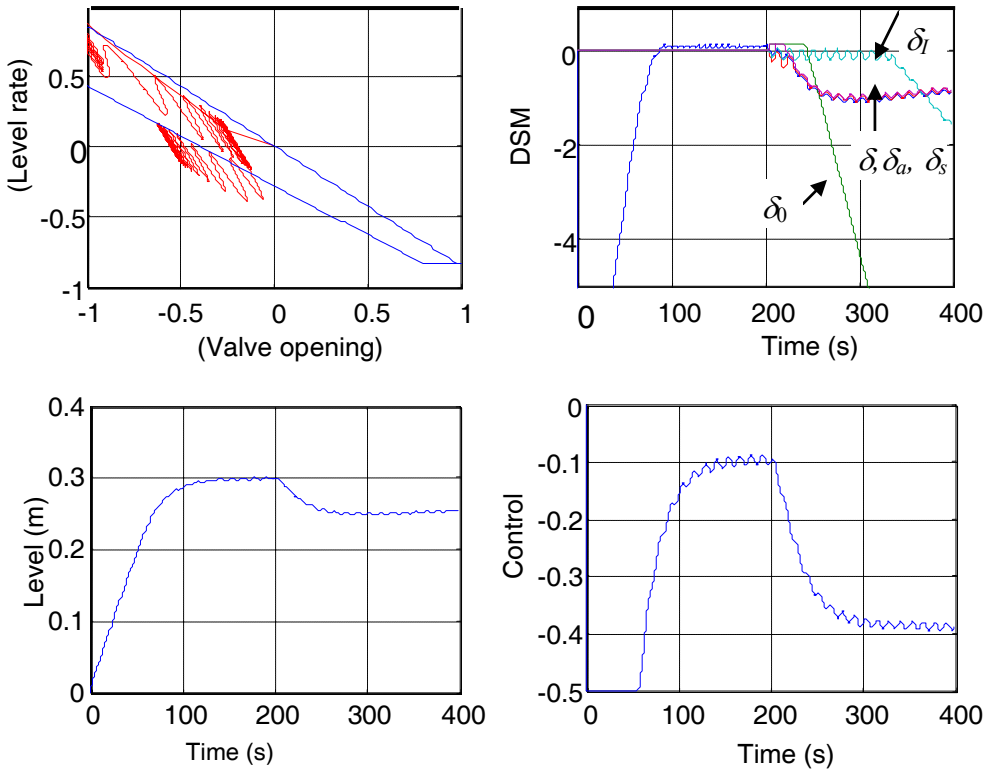


Figure 3-14: Actuator fault response with disturbance

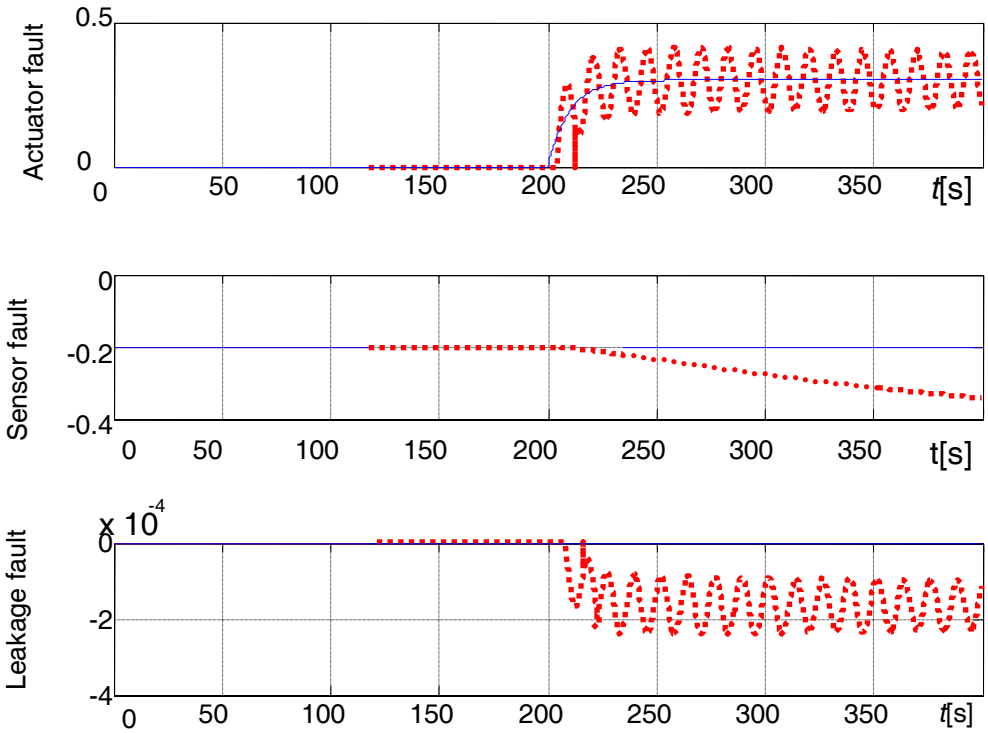


Figure 3-15: Estimation of faults for actuator fault with disturbance

CHAPTER 4

PERFORMANCE RECOVERY USING DYNAMIC SAFETY MARGIN

4.1 Introduction

The system performance deviates from the desired one due to different reasons, for instance, faults, disturbance, etc. Therefore, the performance recovery of a controlled system is an important task in order to enhance system dependability. It is addressed by different techniques, in particular, adaptive control and robust control. Designing a controller based on DSM is important to maintain a predefined margin of safety during transient and steady state in normal or due to disturbance actions. Moreover, it can help speeding up performance recovery in some cases of system faults. Hence, the controller design based on DSM is the main focus of this chapter. PID controller is one of the most popular controllers, particularly, for SISO systems. Hence, adapting PID controller parameters based on DSM is highlighted in *Section 4.2* as an example of controller design based on DSM. For MIMO systems, MPC is successfully used in process control due to its ability to handle explicitly hard constraints on control and states. Therefore, it has been widely applied in petrochemical and related industries. MPC design based on DSM is addressed as another example for controller design based on DSM for SISO and MIMO systems as well.

An FTC system is a performance recovery system due to faults. As stated in [2], [104], and *Chapter 2*, it is a control system that can accommodate components faults, and it is to maintain stability and acceptable degree of performance not only when the system is fault-free but also when component malfunctions are present. FTC prevents faults, which occur in a subsystem, from developing into failures at the system level. Hence, the application of MPC in FTC can be very useful, particularly because most of the processes have control and state constraints, which specify the actuator limits and safety requirements of the components. A frame work of FTC system using MPC based on DSM is introduced in *Section 4.4*.

4.2 Controller Design Based on Dynamic Safety Margin

To maintain the system states within a predefined margin of safety, the value of DSM has to be considered in controller design. The controller design based on DSM has the advantage that the system will be maintained within the safe region not only during the normal operation (transient and steady state) but also in case of fault or disturbance. The inclusion of DSM into the controller design can be achieved by various methods; some of them are introduced in C*hapter* 2. In this section, the inclusion of DSM in controller design, especially FTC to recover the system performance in faulty system, is addressed. The DSM value can be used as a performance index to adapt the parameters of a certain controller, select a controller among different pre-designed controllers, or to combine both methods of controller selection and tuning.

4.2.1 Single Controller Tuning

Adaptive control is one of the control techniques used to improve the system performance by adapting the controller parameters based on the deviation of the system performance from the desired one in case of disturbance or modeling error in the system. An adaptive controller, as stated in [173], is a controller with adjustable parameters and mechanism for adjusting the parameters (see Figure 4-1). Model Reference Adaptive System (MRAS) (direct adaptive control) and Self-tuning Regulators (STR) (indirect adaptive control) are the most common approaches for parameter adjustments [173]. Adaptive controller seems to be the most natural approach to accommodate faults; the faults effects appear as model parameter changes, and they are identified online, and the control law is reconfigured automatically based on new parameters [97]-[98]. The controller acts as PFTC where no information about the fault is introduced.

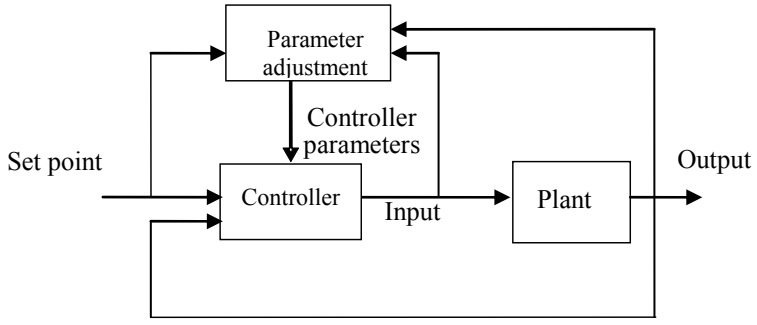


Figure 4-1: Block diagram of the adaptive system [173]

DSM can be used as a performance index, instead of output error in MRAS technique, to tune the controller parameters in order to maintain the safety requirements in addition to the output performance. A brief description about controller adapting based on DSM is introduced in *Chapter 2*, more details are discussed here.

Based on the MIT rule [173], the controller parameters can be updated by using the following equation:

$$k_i(k+1) = k_i(k) + \alpha_i \frac{\partial}{\partial k_i}\delta(k+1) \tag{4.1}$$

where k_I is the controller parameter number i, α_i is the adaptation parameter, and $\delta(k)$ is the discrete-time form of DSM. The performance index δ^2 can also be used instead of δ in (4.1)

This equation can be applied either for $\delta \geq 0$ or $\delta < 0$. In case of $\delta \geq 0$, applying (4.1) moves the system state far away from the safety boundary. Contrarily, the system state is led to the safety boundary when $\delta < 0$. Since the adaptaition paratmter, α_I, varies based on the sign of δ, using the absolute value of δ in (4.1) is not necessary.

Note that, $\delta(.)$ is a nonlinear and non-differentiable function (2.14) and it depends on **d** (.), which is the distance vector from safe region boundaries. Therefore, $\partial\delta(k+1)/\partial k_i$ can be replaced with a function $f_i(\partial\delta(k+1)/\partial k_i)$ i.e.

$$k_i(k+1) = k_i(k) + \alpha_i f_i(\frac{\partial}{\partial k_i}\mathbf{d}(k+1)) \tag{4.2}$$

If **d** has only one element negative, i.e. only one constraint of Φ is violated, then

$$\frac{\partial}{\partial k_i}\delta(k+1) = \frac{\partial}{\partial k_i}\delta_m(k+1) \tag{4.3}$$

where δ_m is the distance between the current state and the violated constraint $m \in \{1,2...q\}$, and q is the total number of constraints.

If more than one violated constraint are violated, then the infinity norm

$$\frac{\partial}{\partial k_i}\delta(k+1) = \left\| \frac{\partial}{\partial k_i}\mathbf{d}_v(k+1) \right\|_\infty \tag{4.4}$$

is used. It corresponds to the maximum effect of k_i on violated constraints. $\mathbf{d}_v \subseteq \mathbf{d}$ is the distances vector between the current state and violated constraints $v \leq q$.

Adaptation Algorithm: The parameters can be adapted according to the following procedure:

1. Calculate $\mathbf{d}(k)$
2. If $\mathbf{d}(k) \geq 0$ then fix the parameters at the nominal values which satisfy the output performance.
3. If $\mathbf{d}(k)<0$ then adapt the parameters according to (4.4).
4. If $k_i(k+1)$ within the range which satisfy system stability then update the parameters.
5. If one of the adapted parameters is out of the stability range, then it shall be fixed at the closest allowable gain to the calculated one.

4.2.2 Multi-Controller Selection

The system parameters variation and disturbances are not identified in the previous method (single controller tuning based on DSM). Thus, a single controller may not recover the system performance and safety requirements, especially in case of system fault or large disturbance. Recover the system performance can be achieved by reconfigure the controller. Reconfiguration mechanisms can be classified as on-line controller selection and on-line controller calculation methods based on DSM. Controller selection methods, assumed fault conditions are computed a priori in the design phase and initiated on-line, based on the real-time information from the diagnosis or supervisor system. On-line controller design methods are synthesized on-line after the abnormal behaviors are diagnosed. The real-time information, which is obtained from the diagnosis system, and DSM are used to design a new controller. The priority to select a certain pre-computed control law depends on the estimation of the system impairment status and DSM. This approach is highly dependent on prompt and correct operation of the diagnosis system. Any false, missed, or error in diagnosis may lead to a degraded performance or even a complete loss of the stability of the closed loop system.

4.2.3 Multi-Controller Selection and Tuning

In most cases, the information of the diagnosis system may not be sufficiently accurate. Therefore, the selected controller may need to be adapted on-line to compensate the

missed information. A combination between both of the above methods can be used to compensate inaccurate information about the diagnosed system i.e. after a fault has been detected a new controller is selected or redesigned on-line. Moreover, this controller is tuned based on the value of DSM in order to achieve the required output and safety performance.

4.3 Examples of Controller Design Based on DSM

4.3.1 PID Controller Tuning for SISO Systems

The PID controller is one of the popular controllers used in more than 80% of industrial SISO process. It has dominated industrial control for half a century, and there has been a great deal of research interest into the implementation of the advanced controllers. The reason is that the PID control has a simple structure, which is easy to be understood by field engineers, and it is robust to disturbance and system uncertainty [129]]. A tutorial given by Hang et al. [174] outlined the recent development in PID parameters adjustment based on relay feedback test. Some other techniques have also been used in developing auto-tuning PID controllers, such as the gain and phase margin based method [175]; the stable auto-tuning PID method designed using the Lyapiunov method [176], etc.

Methods based on online parameter estimation have also been proposed for the automatic tuning of PID regulators. Some authors proposed auto-tuning regulators based on pole placement or Linear Quadratic Gaussian (LQG) design methods. Auto-tuning of PID using adaptive parameter estimation method is proposed in [177]. Another method for auto-tuning is to use expert (neural network, fuzzy, etc.) system to tune the controller see for example [129], [178].

In this section, a mathematical formula to adapt PID controller parameters based on DSM for a system defined by state space model is deduced, in order to satisfy the output performance and safety requirements.

The control signal $u \in \Re$ at any instant k, using a discrete PID controller, is defined as

$$u(k) = K_P e(k) + K_I \sum_{j=1}^{k} e(j) + K_D \frac{e(k) - e(k-1)}{T} \tag{4.5}$$

where K_P, K_I, and K_D are the controller proportional, integral, and derivative gains respectively.

It is required to adapt the PID controller parameters (K_P, K_I, and K_D) to achieve the safety requirements ($\mathbf{d}(k+1) \geq 0$) in addition to the output performance. Hence, the incremental values of the parameters should depend on DSM.

Consider that the safe operation region is a polytope, then the distance vector $\mathbf{d}(.)$ based on (2.13) is defined as

$$\mathbf{d}(k+1) = \mathbf{d}_c - \mathbf{D}_a \mathbf{x}(k+1) \tag{4.6}$$

Substituting by the state space model and control input equation of PID controller then

$$\mathbf{d}(k+1) = \mathbf{d}_c - \mathbf{D}_a(\mathbf{A}\mathbf{x}(k) + \mathbf{b}(K_p e(k) + K_I \sum_{j=1}^{k} e(j) + K_d \frac{e(k)-e(k-1)}{T})) \tag{4.7}$$

Refereeing to the single controller tuning method described in the previous section, then

$$\mathbf{d}_v(k+1) = \mathbf{d}_c^v - \mathbf{D}_a^v \mathbf{x}(k+1)$$

where $\mathbf{D}_a^v \in \Re^{vxn} \subseteq \mathbf{D}_a \in \Re^{qxn}$ and $\mathbf{d}_c^v \in \Re^{vx1} \subseteq \mathbf{d}_c \in \Re^{qx1}$. The variation of $\mathbf{d}(k+1)$ with respect to PID parameters are given by:

$$\begin{aligned}
&\frac{\partial}{\partial k_P}\mathbf{d}_v(k+1) = -\mathbf{D}_a^v(\mathbf{b}e(k)) \\
&\frac{\partial}{\partial k_I}\mathbf{d}_v(k+1) = -\mathbf{D}_a^v(\mathbf{b}\sum_{j=1}^{k} e(j)) \\
&\frac{\partial}{\partial k_D}\mathbf{d}_v(k+1) = -\mathbf{D}_a^v(\mathbf{b}\frac{e(k)-e(k-1)}{T})
\end{aligned} \tag{4.8}$$

and the updated parameters are

$$\begin{aligned}
&k_P(k+1) = k_P(k) + \alpha_P \left\| (-\mathbf{D}_a^v \mathbf{b}e(k)) \right\|_\infty \\
&k_I(k+1) = k_I(k) + \alpha_I \left\| (-\mathbf{D}_a^v(\mathbf{b}\sum_{j=1}^{k} e(j)) \right\|_\infty \\
&k_D(k+1) = k_D(k) + \alpha_D \left\| (-\mathbf{D}_a^v(\mathbf{b}\frac{e(k)-e(k-1)}{T})) \right\|_\infty
\end{aligned} \tag{4.9}$$

The initial values of the controller parameters are designed in order to satisfy the output performance in normal operation.

Example 4.1

Consider the DC motor model in *Example* 2.1, and assume that there exists a sudden load torque disturbance 0.5 N.m after 15 s. Figure 4-2 shows the system response and state trajectory using fixed PID controller parameters where K_P, K_I, and K_D are 4, 2, and 2 respectively. The output response of the fixed PID controller in Figure 4-2c is accepted with respect to the transient and the steady state because the settling time is less 5 s, and the steady state error is almost zero. However, the state trajectory lies outside the safe boundaries at the transient (Figure 4-2a) i.e. DSM <0. By decreasing the PID controller gain, the DSM response may be improved. On the other side, the reaction time (rise and settling time) will increase. Figure 4-3 shows the motor response and DSM variation using PID controller where K_P, K_I, and K_D are 1.15, 1, and 1 respectively. Note that the DSM is positive in transient and steady state response; contrarily, the motor reached the steady state after 10 s, while the motor reached steady state after 5 s in Figure 4-2. Therefore, to obtain a fast response in addition to maintaining positive DSM in transient and steady-state period, the PID controller parameters have to be adapted based on DSM.

Figure 4-4 shows the motor response using adapted PID controller parameters based on (4.9) where the nominal controller parameters K_P, K_I, and K_D are 4, 2, and 2 respectively (parameters of Figure 4-2); the adaptation parameters α_P, α_I and α_D are 0.9, 0.001, and 0.05 respectively. Comparing the response of Figure 4-2 with Figure 4-4, it is clear that the controller in Figure 4-4 tries to pull the state trajectory in the direction of the safe region and the output response is almost similar to Figure 4-2. Changing the nominal controller parameters or adaptation factors could enhance DSM response. Figure 4-5 shows the motor response using adapted PID controller based on DSM where nominal parameters K_P, K_I, and K_D are 1.15, 1, and 1 respectively (parameters of Figure 4-3) and the same adaptation parameters. It is clear that the response in Figure 4-5 is the best among the previous responses, because the settling time is less than 5 s and the DSM is almost positive in the transient period. Despite using the same parameters in both responses of Figure 4-5 and Figure 4-3, the system response in Figure 4-5 is faster with acceptable DSM.

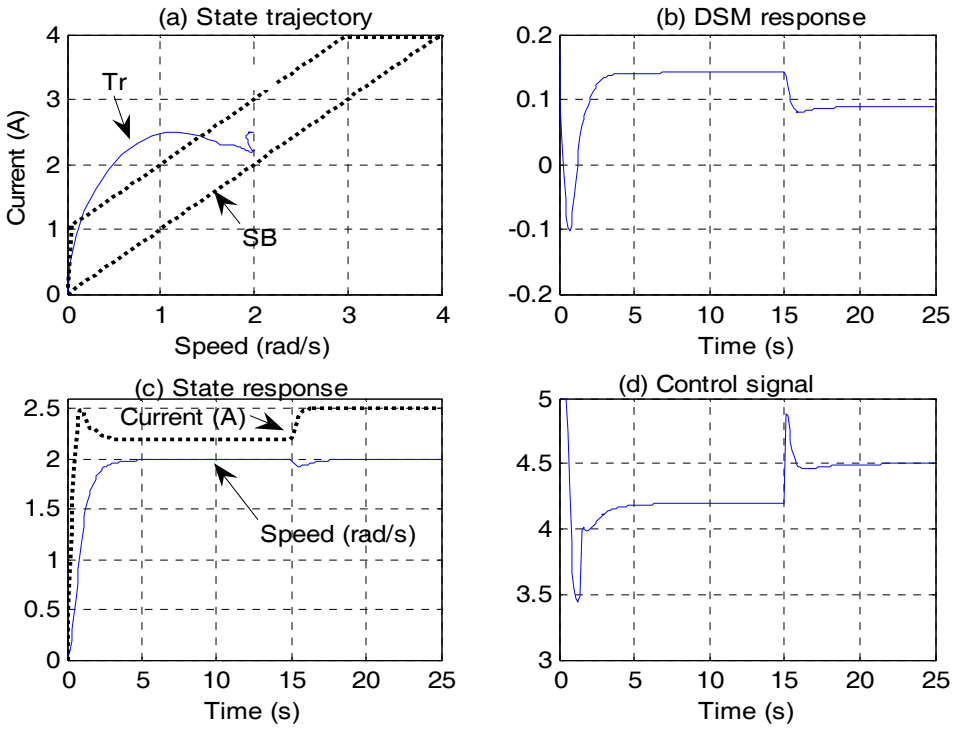


Figure 4-2: DC motor response using fixed parameter PID; KP=4, KI=2 and KD=2

(SB: Boundary and Tr: Trajectory)

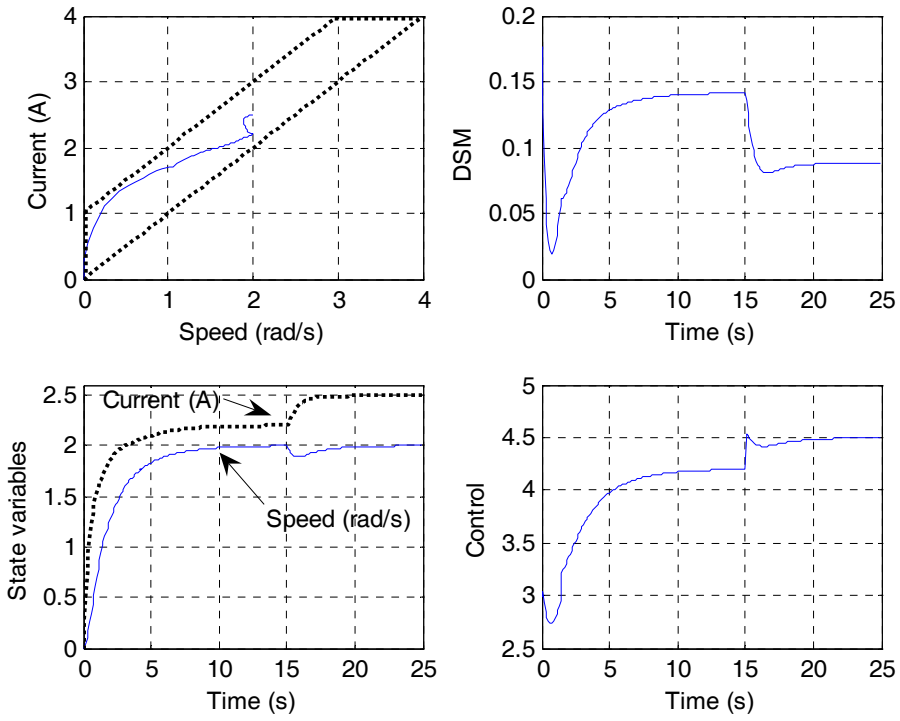


Figure 4-3 DC motor response using fixed parameter PID; K_P=1.15, K_I=1 and K_D=1

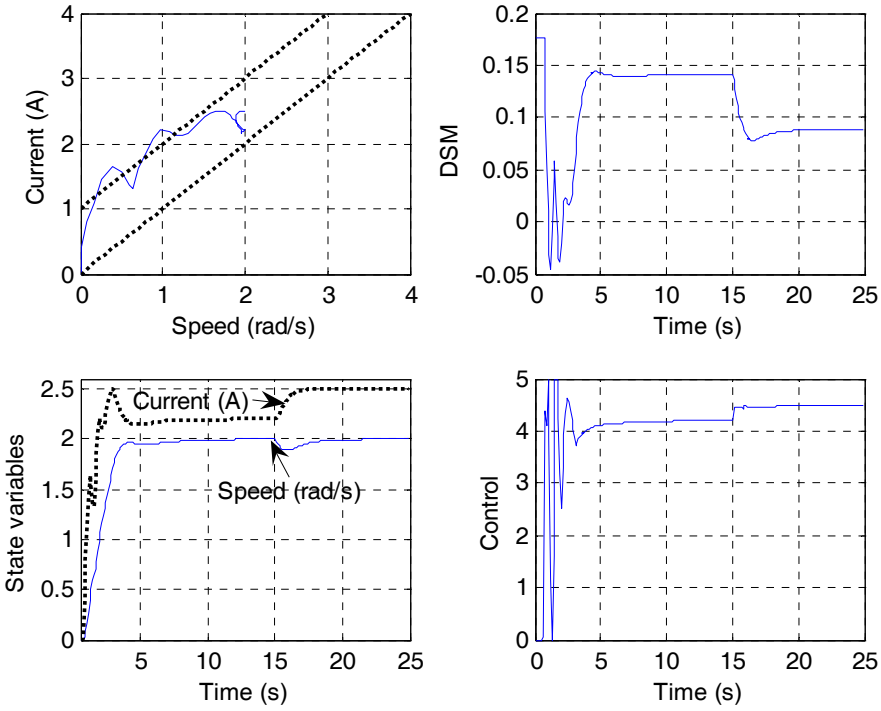


Figure 4-4: DC motor response using adapted PID controller;
K_P=4, K_I=2 and K_D=2

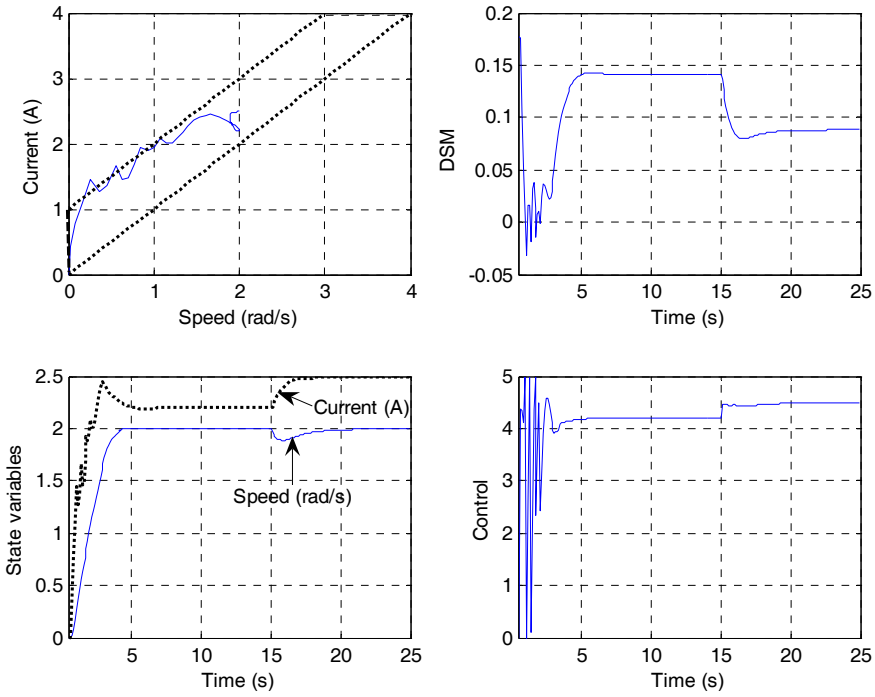


Figure 4-5: DC motor response using adapted PID controller;
K_P=1.15, K_I=1 and K_D=1

4.3.2 Predictive Controller Design Based on DSM for SISO and MIMO Systems

Model predictive Control (MPC) or receding horizon control (RHC) is a form of control in which the current control action is obtained by solving on-line, at each sampling instance, a finite horizon optimal control problem, using the current state of the plant as the initial state. The internal model is used to obtain prediction of system behavior over the finite horizon [179]-[182]. The optimization yields an optimal control sequence, but only the first control of the sequence is applied to the plant and in the next sampling time, the complete calculation is repeated (*receding horizon principle*). This is the main difference from conventional control, which uses a pre-computed control law.

LQR (Linear Quadratic Regulator) and EA (Eigen Assessment) are among the most popular controller design techniques for MIMO systems. Each one has its own advantages and disadvantages [102], [89]. Most of the process operates at control and state constraints. It is not easy to handle control and state constraints using EA controller design. LQR is an infinite horizon optimization problem and therefore LQR design with control and state constraints is hard.

Since MPC is formulated as an optimization problem, inequality constraints can naturally be added to the controller [182]. It naturally handles the control of multivariable plant and takes into account the information on constraints arising from equipment limitations, safety requirements, etc. In its usual form, it does this by combining linear dynamic models with linear inequalities, which seems to be a very powerful combination, since the linear model keeps the dynamic simple, while the inequalities can be used to represent important nonlinearities, as well as constraints. The usual formulation of MPC using a quadratic or linear cost function combined with a linear model and linear inequalities leads to a quadratic programming (QP) or linear programming (LP) optimization problem [181], [183]. The ability to handle explicitly hard constraints on control and states may be viewed as one of the major factors of the success of MPC in process control. Therefore, it has been widely applied in process industries. Although constraints improve the appeal of MPC as an advanced control strategy, they make the controller implementation difficult.

The control law of a predictive controller, for a system defined by the state-space model, is obtained by minimizing the 2-norm measure of predicted performance [119], [181] given by

$$J = \sum_{i=N_1}^{N} \left\| \hat{\mathbf{e}}(i+k|k) \right\|_{\mathbf{Q}_i}^2 + \sum_{i=0}^{N_u-1} \left\| \mathbf{u}(i+k|k) \right\|_{\mathbf{R}_i}^2 \tag{4.10}$$

subject to

$$\begin{aligned} &\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ &\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \\ &\mathbf{x}(k+i|k) \in \mathbf{X}_s \\ &\mathbf{u}_{min} \le \mathbf{u}(k+i) \le \mathbf{u}_{max} \end{aligned} \tag{4.11}$$

with respect to the control sequence $\underset{\rightarrow}{\mathbf{u}}$

where

$$\underset{\rightarrow}{\mathbf{u}} = \left[\mathbf{u}(k)\, \mathbf{u}(k+1) \ldots \mathbf{u}(k+N_u-1)\right]^{\mathrm{T}} \in \Re^{r.N_u};$$

$$\hat{\mathbf{e}}(k+i|k) = \mathbf{y}_d(k+i) - \hat{\mathbf{y}}(k+i|k)$$

$\left\|\mathbf{e}\right\|_{\mathbf{Q}}^2 = \mathbf{e}^{\mathrm{T}}\mathbf{Q}\,\mathbf{e}$, $\hat{\mathbf{e}}(k+i|k) \in \Re^m$ is the predicted error between the desired and predicted response. $\mathbf{x} \in \Re^n$ is the system state vector; $\boldsymbol{y}_{\mathrm{d}} \in \Re^m$ is the reference output vector. $\hat{\mathbf{x}}(k+1)|k$ is the prediction of $\mathbf{x}(k+i)$ made at instance k, $\mathbf{X}_s \subseteq \mathbf{X}$ is the set of state vectors which satisfy all state constraints. $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ system parameter matrices of adequate dimensions, $\mathbf{Q}_i$ are the error weighting matrices, $\mathbf{R}_i$ are the input weighting matrices. N, N_1 and N_u are the maximum, minimum, and control horizons, respectively. Notice that $\mathbf{Q}$, $\mathbf{R}$, N, N_1, and N_u are free design parameters.

The MPC control law is based on the following idea: At time k, compute the optimal solution $\underset{\rightarrow}{\mathbf{u}}^* = \left\{\mathbf{u}_k^*, \ldots, \mathbf{u}_{k+N_u-1}^*\right\}$ to problem (4), apply $\mathbf{u}(k) = \mathbf{u}_k^*$ as input to the system, and repeat the optimization at time $k+1$ based on the new state $\mathbf{x}(k+1)$.

In most cases, $\mathbf{X}_s$ is a polytope defined by a set of linear inequalities in the form $\mathbf{a}_i^T\mathbf{x} \le c_i$, $i=1,\ldots,q$, where $\mathbf{a}_i \in \Re^n$, $c_i \in \Re$, and q is the number of constraints. Therefore the state constraints can be written as

$$D_c\mathbf{x}(k+i|k) \le \mathbf{c}_c \tag{4.12}$$

where $\mathbf{D}_c = [\mathbf{a}_1 \ \ldots \ \mathbf{a}_q]^{\mathrm{T}}$ and $\mathbf{c}_c = [c_1 \ldots c_q]^{\mathrm{T}}$

4.3.2.1 Model Predictive Control with DSM constraints

The design of MPC based on DSM can be handled by replacing $\mathbf{X}_s$ with the safe operation region$\mathbf{\Phi}$. The state constraints can be written in the form

$$\hat{\delta}(k+i|k) \geq 0 \text{ or } \hat{\mathbf{d}}(k+i|k) \geq 0 \tag{4.13}$$

where $\hat{\delta}(k+i|k) \in \Re$ and $\hat{\mathbf{d}}(k+i|k) \in \Re^q$ are the prediction of DSM and the distance vector between the predicted state and the boundaries of $\mathbf{\Phi}$, made at instance k, respectively.

Assuming that Φ is a polytope, then the distance vector $\mathbf{d}(.)$ is obtained from (2.3), which is deduced in *Chapter 2*

Consider the system model with input constraints and DSM constraints (4.13) then the objective function according to (4.10) is

$$J = \underset{\rightarrow}{\mathbf{u}}^T \mathbf{M} \underset{\rightarrow}{\mathbf{u}} + 2\mathbf{H}\underset{\rightarrow}{\mathbf{u}} + c_r \tag{4.14}$$

subject to

$$\begin{aligned} &\mathbf{d}_t - \mathbf{D}_A \mathbf{x}(k) \geq \mathbf{D}_b \underset{\rightarrow}{\mathbf{u}} \\ &\underset{\rightarrow}{\mathbf{u}}_{min} \leq \underset{\rightarrow}{\mathbf{u}}(k+i) \leq \underset{\rightarrow}{\mathbf{u}}_{max} \end{aligned} \tag{4.15}$$

where

$$\mathbf{M} = \mathbf{C}_B{}^T \mathbf{Q}_t \mathbf{C}_B + \mathbf{R}_t;$$

$$\mathbf{H} = (\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a \mathbf{x}(k))^{\mathrm{T}} \mathbf{Q}_t \mathbf{C}_B;$$

$$\mathbf{c}_r = (\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a \mathbf{x}(k))^{\mathrm{T}} \mathbf{Q}_t (\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a \mathbf{x}(k));$$

$$\underset{\rightarrow}{\mathbf{y}} = \begin{bmatrix} \mathbf{y}_d(k+N_1) \\ \mathbf{y}_d(k+N_1+1) \\ \vdots \\ \mathbf{y}_d(k+N) \end{bmatrix} \in \Re^{m.(N-N_1+1)};$$

$$\underset{\rightarrow}{\mathbf{u}}_{min} = \begin{bmatrix} \mathbf{u}_{min} \\ \mathbf{u}_{min} \\ \vdots \\ \mathbf{u}_{min} \end{bmatrix} \in \Re^{r.N_u} \; ; \; \underset{\rightarrow}{\mathbf{u}}_{max} = \begin{bmatrix} \mathbf{u}_{max} \\ \mathbf{u}_{max} \\ \vdots \\ \mathbf{u}_{max} \end{bmatrix} \in \Re^{r.N_u}$$

$$\mathbf{R}_t = \begin{bmatrix} \mathbf{R}_1 0 \cdots 0 \\ 0\, \mathbf{R}_1 \cdots \\ \vdots \quad \ddots \\ 0\, 0 \cdots \mathbf{R}_1 \end{bmatrix} \in \Re^{r.N_u \times r.N_u};$$

$$\mathbf{Q}_t = \begin{bmatrix} \mathbf{Q}_1 0 \cdots 0 \\ 0\, \mathbf{Q}_1 \cdots \\ \vdots \quad \ddots \\ 0\, 0 \cdots \mathbf{S}_1 \end{bmatrix} \in \Re^{m.(N-N_1+1) \times m.(N-N_1+1)}$$

$$\mathbf{D}_A = \begin{bmatrix} D_c A^{N_1} \\ D_c A^{N_1+1} \\ \vdots \\ D_c A^{N} \end{bmatrix} \in \Re^{q(N-N_1+1) \times n};\ \mathbf{d}_t = \begin{bmatrix} d_c \\ d_c \\ \vdots \\ d_c \end{bmatrix} \in \Re^{q(N-N_1+1)};$$

$$\mathbf{C}_a = \begin{bmatrix} CA^{N_1} \\ CA^{N_1+1} \\ \vdots \\ CA^{N} \end{bmatrix} \in \Re^{m.(N-N_1+1) \times n}$$

and $\mathbf{C}_B = \mathbf{C}_b + \mathbf{D}_u$;

$$\mathbf{C}_b = \begin{bmatrix} \mathbf{CA}^{N_1-1}\mathbf{B} & \cdots & \mathbf{CB} & 0 & 0 \\ \mathbf{CA}^{N_1}\mathbf{B} & \cdots & & \ddots & 0 \\ \vdots & & \ddots & & \vdots \\ \mathbf{CA}^{N-1}\mathbf{B} & \cdots & \cdots & \cdots & \mathbf{CA}^{N-N_u}\mathbf{B} \end{bmatrix} \in \Re^{m(N-N_1+1) \times r.N_u};$$

$$\mathbf{D}_b = \begin{bmatrix} \mathbf{D}_a\mathbf{A}^{N_1-1}\mathbf{B} \cdots \mathbf{D}_a\mathbf{B} & 0 & \cdots\cdots 0 \\ \mathbf{D}_a\mathbf{A}^{N_1}\mathbf{B} & \cdots\cdots \mathbf{D}_a\mathbf{B} & \cdots\cdots & 0 \\ \vdots & & & \\ \mathbf{D}_a\mathbf{A}^{N-1}\mathbf{B} \cdots\cdots\cdots\cdots & \mathbf{D}_a\mathbf{A}^{N-N_u}\mathbf{B} & & \end{bmatrix} \in \Re^{q(N-N_1+1) \times r.N_u};$$

$$\mathbf{D}_u = \begin{bmatrix} \overbrace{\begin{matrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{matrix}}^{r(N_1)} & \overbrace{\begin{matrix} \mathbf{D} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{D} \end{matrix}}^{r(N_u - N_1)} \end{bmatrix} \left.\vphantom{\begin{matrix} a \\ b \\ c \end{matrix}}\right\} m(N_u - N_1) \\ \begin{bmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} \left.\vphantom{\begin{matrix} a \\ b \\ c \end{matrix}}\right\} m(N - N_u + 1)$$

The minimization of (4.14) is known as a Quadratic Programming (QP) problem. Since the problem depends on the current state $\mathbf{x}(k)$, the implementation of MPC requires the on-line solution of a QP at each time step. Although efficient QP solvers ([180]-[182], [184]) based on *active set* methods and *interior point* methods [124], [123], [181] are available, computing the input $\mathbf{u}(k)$ demands significant on-line computation effort. For this reason, the application of MPC has been limited to "slow" and/or "small" processes. If all the constraints are inactive, the solution of the predictive controller is exactly the same as in the unconstrained case (see *Appendix* C). But if the constraints become active then the controller becomes nonlinear. The constrained predictive control law is a linear time invariant control law, in case that the set of active constraints is fixed. In practice the set of active constraints changes, so the control law seems as consisting of a number of linear controllers with the same structures and switching between them [181].

A new idea to design a piecewise-linear controller for MPC with constraints is introduced in [185] and [186]. The idea is that for small problems, in which the state-space is divided up into a manageably small number of (convex) pieces, one could pre-compute (off-line) the control law that should be applied in each piece, and then the MPC algorithm would consist simply of reading the appropriate gain matrix from a look-up table, depending on the current state estimate. This idea is not feasible for application in which the number of constraints is large. The approach taken by [185] and [186] is based on the observation that in the MPC problem, the inequality constraints (4.15) depends on the current state $\mathbf{x}(k)$, which can be thought as a set of parameters of the QP problem. Therefore, the problem in (4.14) is defined as a multi-parametric quadratic program (mp-QP) [185], [181].

The solution of mp-QP described in [185] is a Piecewise Affine (PWA). The complexity of the polyhedral partition tends to increase rapidly with the number of

constraints, and the dimension of state vector. This has led to approximate algorithms for solving mp-QP problems being investigated in [187], and [188], with significant reduction in complexity. Moreover, it has led to the investigation of efficient implementation of piecewise linear function evaluation [189]. Several properties of the geometry of the polyhedral partition and its relation to the combinations of active constraints at the optimum of the quadratic program of the approach in [185] are analyzed in [190].

A major problem, which can occur with constrained MPC, is that the optimization problem may be infeasible. Standard QP solvers just stop in such case [181], [182]. This can happen because an unexpected large disturbance or fault has occurred. Therefore, there is really no way in which the plant can be kept within specified constraints. Some times also it can happen because the real plant behaves differently from the internal model. The predictive controller may then attribute differences between the plant and the model behaviors to large disturbances or fault. If these keep growing, then it can eventually decide, erroneously, that it does not have enough control authority to keep the plant within constraints. There are many ways in which the predictive control problem can become infeasible, and most of them are difficult to anticipate [181].

Thus, it is essential to have a strategy for dealing with the possibility of infeasibility. Various possibilities exist, ranging from *ah hoc* measures such as outputting the same control signal as in the previous case, or (better) the control signal computed $\mathbf{u}(k+2|k)$ in the previous step, to sophisticated strategies of 'constraints managements', in which one tries to relax the least-important constraints in an attempt to regain feasibility [181]. Typically, some of the constraints, such as physical limitations, must be enforced at all times, while other constraints can be relaxed in order to transform the optimization problem into a feasible one in the case of infeasibility.

There exist techniques which transform an infeasible MPC into a feasible one without being able to *explicitly* differentiate between the relative importance's among the constraints, see e.g. [191] and [192]. However, the constraints are often not equally important, e.g. a safety constraint is usually more important than a product quality constraint. One way to explicitly express this difference in importance is to give the constraints different priorities. When the on-line optimization problem becomes infeasible, the lowest prioritized constraints are dropped [191]. In the research literature, there are some algorithms and methods to solve infeasible MPC based on the relaxation

of constraints with different priorities, see for example [193]-[198], [184], [185], and [199].

The work in [193] discusses issues related to the problems of infeasibility in constrained predictive control, and proposes several strategies to solve such problems, including strategies that involve priority levels. The most rigorous approach it proposes for infeasibility handling is to satisfy as many of the highest prioritized constraints as possible, and then compute a feasible relaxation of the other constraints by treating them as soft constraints, that is, a term is added to the cost function in the original MPC optimization problem that penalizes the violations of these constraints.

An approach is presented in [194] and [195] for solving infeasible MPC problems considering that the constraints have different priorities,. In this approach, integer variables are introduced in order to handle the priorities in an optimal fashion. The minimization of the size of the violation of the constraints is performed according to their prioritization by solving a sequence of mixed integer optimization problems. In [196], another algorithm to minimize a sequence of LP (or QP) problems in addition to the original MPC optimization is presented. An important difference between the algorithms presented in [193]-[196] and the other approaches mentioned above which also take prioritization into account, is that the algorithms in [193]-[196] minimize the violations of those constraints which can not be fulfilled.

A modification of the [196] approach is presented in [197], [198], and [185]. In the case when all constraints have different priorities, it reduces the sequence of LP problems to a single LP problem by selecting the weights (or, cost vector) in this LP problem. It solves only a single LP in addition to the standard QP problem on-line in order to find the feasible solution of MPC with constraints.

The approach, which is introduced [198], divides the problem into a multi-objective framework that can handle a large class of prioritized objectives and constraints in an optimal fashion. The internal model, objectives and their relative can be changed on-line without the need for redesigning the controller off-line. However, this increase in flexibility also demands an increase in the amount of on-line computation power that is required.

If there is an uncertain input in the system, which can be considered as the additive fault information in this case, Min-Max predictive controller design can be used [200]-[202].

Example 4.2

Consider an MIMO system of magnetic tape drive system explained in [203] is given by

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 10 \\ 3.315 & -3.315 & -0.5882 & -0.5882 \\ 3.315 & -3.315 & -0.5882 & -0.5882 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 8.533 & 0 \\ 0 & 8.533 \end{bmatrix}\mathbf{u}$$

$$\mathbf{y} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ -2.113 & 2.113 & 0.372 & 0.375 \end{bmatrix}\mathbf{x} \tag{4.16}$$

A brief description of the system is introduced in *Appendix* D.

Consider that, the system has the following linear boundary constrains

1. $\phi_1 = [-2.11\ 2.113\ 0.372\ 0.375]\mathbf{x} \le 2.5$
2. $\phi_2 = [-2.11\ 2.113\ 0.372\ 0.375]\mathbf{x} \ge 1.75$ (4.17)
3. $\phi_3 = [0.5\ 0.5\ 0\ 0]\mathbf{x} \le 1.5$
4. $\phi_4 = [0.5\ 0.5\ 0\ 0]\mathbf{x} \ge 0.5$

The distance vector from these boundaries is given from

$$\mathbf{d}(k) = \mathbf{d}_c - \mathbf{D}_a\,\mathbf{x}(k) \ge 0$$

and the control input constraints are

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \le \mathbf{u}(k) \le \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

where $\mathbf{D}_a = \begin{bmatrix} -0.697 & 0.697 & 0.12276 & 0.12276 \\ 0.697 & -0.697 & -0.12276 & -0.12276 \\ -0.707 & 0.707 & 0 & 0 \\ 0.707 & -0.707 & 0 & 0 \end{bmatrix}$, $\mathbf{d}_c = \begin{bmatrix} 0.825 \\ -0.5775 \\ 1.7121 \\ -0.5707 \end{bmatrix}$

$\delta(k) = \min_{1 \le i \le q} \mathbf{d}_i(k)$, $\delta(k)$ is the DSM at sampling instance k, q=4 (number of constraints) and $\mathbf{d}_i(k)$ is the variable number i in $\mathbf{d}(k)$.

Figure 4-6 shows the system response and DSM variation using MPC without DSM constraints for a command input vector $\mathbf{r} = [1\ 2]^T$ and the system is affected by

a constant input disturbances ([-0.3 -0.2]^{T}) after 1.5 s. The controller has the following parameters:

$\mathbf{Q}_i = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}$ and $\mathbf{R}_i = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$; N_1=1 and N_u=N_y=5; the desired reference outputs are chosen where

$\mathbf{y}_r(k) = \begin{bmatrix} 1 - e^{-Tk/0.15} \\ 2.\left(1 - e^{-Tk/0.1}\right) \end{bmatrix}$, where the sampling time T=0.05.

In this case, the control law is (see *Appendix* C)

$$\mathbf{u}(k) = \left[\mathbf{I}_r : 0 : \cdots 0\right]\left[\mathbf{K}_y \underset{\rightarrow}{\mathbf{y}} - \mathbf{K}_x \mathbf{x}(k)\right] \tag{4.18}$$

$$\mathbf{K}_y = \left[\mathbf{R}_t + \mathbf{C}_B{}^{\mathrm{T}} \mathbf{Q}_t \mathbf{C}_B\right]^{-1} \mathbf{C}_B{}^{\mathrm{T}} \mathbf{Q}_t$$

$$\mathbf{K}_x = \left[\mathbf{R}_t + \mathbf{C}_B{}^{\mathrm{T}} \mathbf{Q}_t \mathbf{C}_B\right]^{-1} \left[\mathbf{C}_B{}^{\mathrm{T}} \mathbf{Q}_t \mathbf{C}_a\right]$$

The system response is acceptable in the transient and steady state, since that the 1% settling time is less than 1 sec for each output, and steady state errors are very small. However, the system behavior is not accepted in case of disturbance, DSM is negative and the steady state errors are 10% and 25% for the first and the second output respectively. Figure 4-7 shows the system response using MPC with DSM constraints; q*uadprog* function in *Matlab* toolbox is used in the simulation in order to solve the quadratic optimization with hard constraints (4.14). The controller parameters are taken as in the previous response (Figure 4-6). The response is very bad since the steady state errors are 50% and 25% for the first and the second output respectively. The DSM is negative as well, not only due to disturbance but also in the normal case. The reason for that is the infeasibility of the solution of MPC with DSM constraints. Thus, to improve the responses the infeasibility problem should be solved.

Based on the previous discussion about the infeasibility solution, it is clear that changing the parameters of the controller can solve the infeasibility of MPC with constraints. Figure 4-8 shows the MPC with DSM constraints with new controller parameters as follow:

$\mathbf{Q}_i = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}$ and $\mathbf{R}_i = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$; N_1=1 and N_u=N_y= 3.

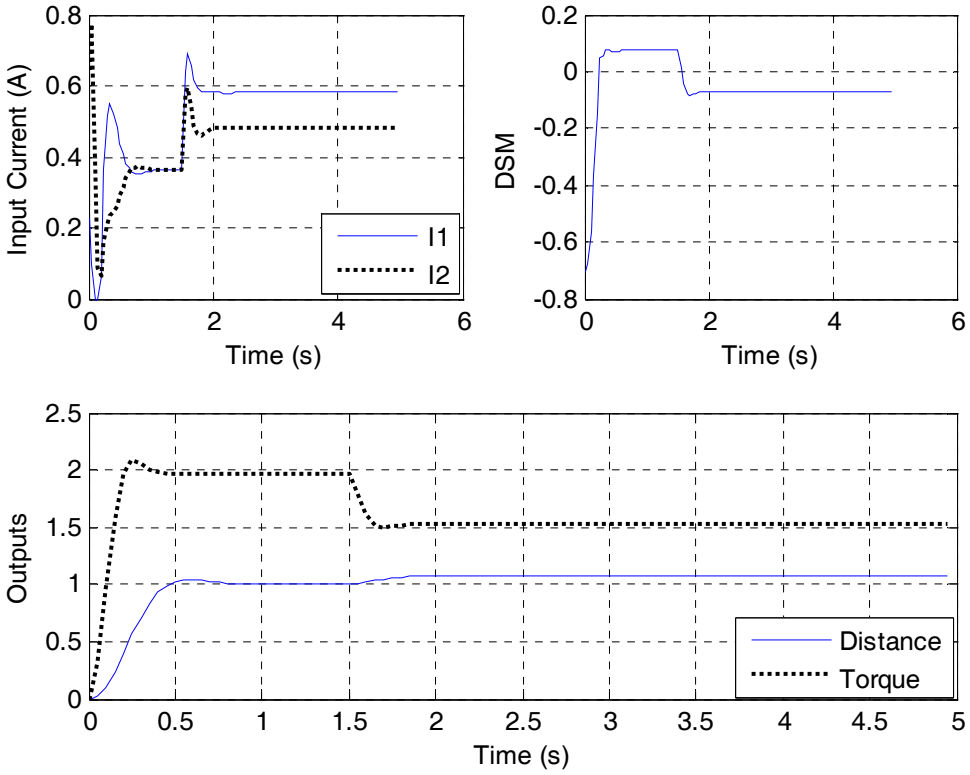


Figure 4-6: Tape-system response using MPC without DSM constraints

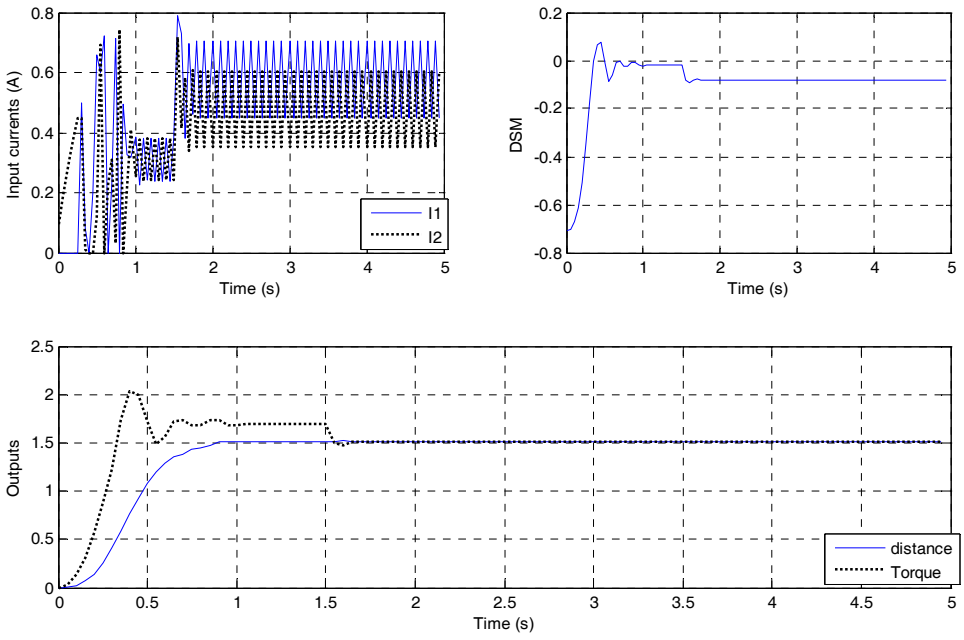


Figure 4-7: Tape-response using MPC with DSM and the same parameters of MPC without DSM

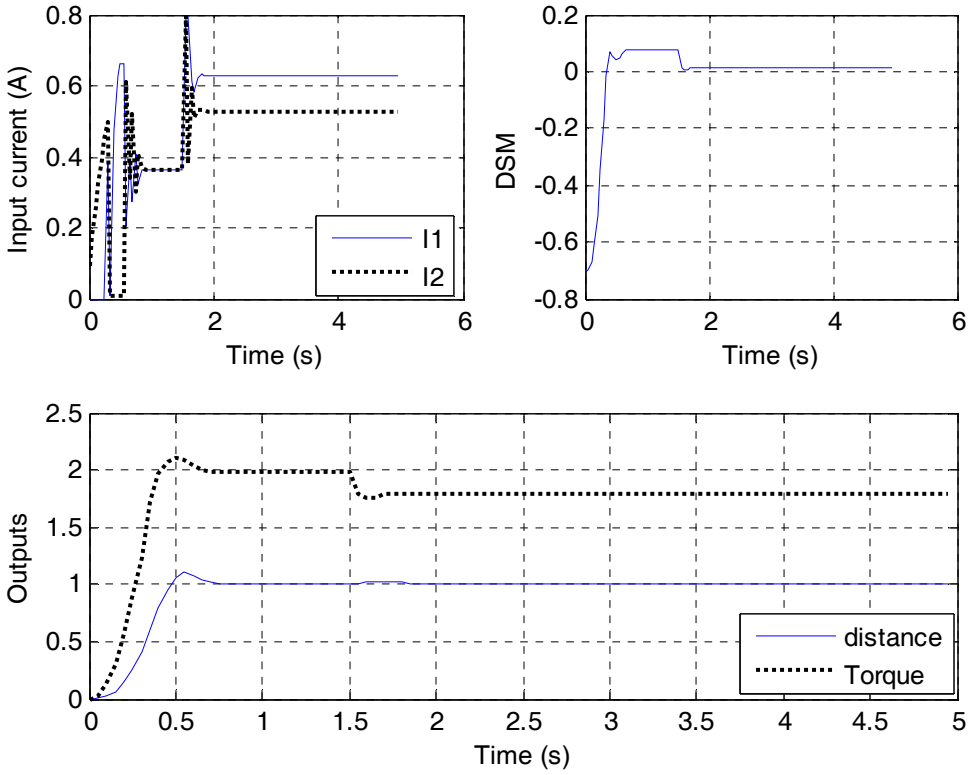


Figure 4-8: Tape-Response with DSM as hard constraints

The responses of the outputs and DSM (Figure 4-8) are improved, not only in normal operation but also in disturbance case.

According to the discussion about constrained MPC and the previous example (*Example* 4.2), it is clear that the computation burden to find a feasible solution is high in addition to the complexity of the algorithms. Therefore, in the following section two different methods are suggested to find a feasible solution and to reduce the on-line computation methods.

4.3.2.1.1 Softening the DSM constraints

Softening constraints is one systematic strategy for dealing with the infeasibility. That is to allow the constraints to be crossed occasionally, but only if really necessary, rather than regarding them as 'hard' boundaries that can never be crossed [181].

The strategy to soften constraints is to add new variables, so-called 'slack variables', which are defined in such a way that they are non-zero only if the constraints are violated. Then their non-zero values are very heavily penalized in the cost function, so that the optimizer has a strong incentive to keep at zero if possible [181].

Here, the distance vector **d** is taken as the slack variables. The 2-norm, 1-norm or ∞-norm of **d**(·) can be introduced as additional term in the main objective function (4.10). The objective function of the predictive controller in this case can be rewritten in the following form

$$J = \sum_{i=N_1}^{N} \left\| \hat{\mathbf{e}}_d (i+k|k) \right\|_{\overline{Q}_i}^2 + \sum_{\mathrm{i}=0}^{N_\mathrm{u}-1} \left\| \mathbf{u}(i+k) \right\|_{R_i}^2 \tag{4.19}$$

subject to (4.15),

where $\mathbf{e}_d(.) = \begin{bmatrix} \mathbf{e}(.) \\ \mathbf{d}(.) \end{bmatrix} \in \Re^{m+q}$, $\overline{\mathbf{Q}}_i = \begin{bmatrix} \mathbf{Q}_i & 0 \\ 0 & \mathbf{P}_i \end{bmatrix}$, and (4.20)

$$\mathbf{P}_i = \mathbf{P}_{io}\, diag\left\{ \left(1 - sign(\delta_j(k).1\right) e^{-d_j(k)} \right\} \; ; j = 1, \cdots q \,. \tag{4.21}$$

$\mathbf{P}_i$ is the weighting matrix for **d**, and it depends on the elements of **d** (*dj*); if δj is negative then the corresponding matrix $\mathbf{P}_i$ is increased and it is zero otherwise. $\mathbf{P}_{io}$ is a constant weighting matrix. The number of free-design parameters, in this case, is increased by $\mathbf{P}_{io}$. In this case, hard constraints are restricted to control inputs. The solution of the problem given in the form of (4.14) can be obtained by using either direct (one-shoot) optimization or dynamic programming [205], [206]. A brief description about the two methods is given in the following section.

a) One-shot Optimization

The control problem, which is formulated as the optimization of (4.19), will now be solved using one-shoot optimization method. Substituting (4.20) and (4.21) in (4.19) then the performance index can be written as

$$J = \left(\begin{bmatrix} \underset{\rightarrow}{\mathbf{e}} \\ \underset{\rightarrow}{\mathbf{d}} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_t & 0 \\ 0 & \mathbf{P}_t \end{bmatrix} \begin{bmatrix} \underset{\rightarrow}{e} \\ \underset{\rightarrow}{\mathbf{d}} \end{bmatrix} + \underset{\rightarrow}{\mathbf{u}} \mathbf{R}_t \underset{\rightarrow}{\mathbf{u}} \right) \tag{4.22}$$

where $\underset{\rightarrow}{\mathbf{e}}$, $\underset{\rightarrow}{\mathbf{d}}$ and $\mathbf{Q}_t$ are defined in the previous section,

$$\mathbf{P}_t \in \Re^{q.N \times q.N} = \begin{bmatrix} \mathbf{P}_1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & \mathbf{P}_N \end{bmatrix}.$$

The solution of (4.22) is deduced in *Appendix C,* and therefore the control law is defined as

$$\underset{\rightarrow}{\mathbf{u}} = \left[\mathbf{K}_y \underset{\rightarrow}{\mathbf{y}} + \mathbf{K}_d \mathbf{d}_t - \mathbf{K}_x \mathbf{x}(k)\right] \tag{4.23}$$

where

$$\mathbf{K}_y = \left[\mathbf{R}_t + \mathbf{C}_B{}^{\mathrm{T}} \mathbf{Q}_t \mathbf{C}_B + \mathbf{D}_b{}^{\mathrm{T}} \mathbf{P}_t \mathbf{D}_b\right]^{-1} \mathbf{C}_B{}^{\mathrm{T}} \mathbf{Q}_t$$

$$\mathbf{K}_d = \left[\mathbf{R}_t + \mathbf{C}_B{}^{\mathrm{T}} \mathbf{Q}_t \mathbf{C}_B + \mathbf{D}_b{}^{\mathrm{T}} \mathbf{P}_t \mathbf{D}_b\right]^{-1} \mathbf{D}_b{}^{\mathrm{T}} \mathbf{P}_t$$

$$\mathbf{K}_x = \left[\mathbf{R}_t + \mathbf{C}_B{}^{\mathrm{T}} \mathbf{Q}_t \mathbf{C}_B + \mathbf{D}_b{}^{\mathrm{T}} \mathbf{P}_t \mathbf{D}_b\right]^{-1} \left[\mathbf{C}_B{}^{\mathrm{T}} \mathbf{Q}_t \mathbf{C}_a + \mathbf{D}_b{}^{\mathrm{T}} \mathbf{P}_t \mathbf{D}_A\right]$$

The first component in $\underset{\rightarrow}{\mathbf{u}}$, namely $\mathbf{u}(k)$, is the control vector applied to the system. This control vector can be obtained from (4.23) as

$$\begin{aligned} \mathbf{u}(k) &= \left[\mathbf{I}_r : 0 : \cdots 0\right]\left[\mathbf{K}_y \underset{\rightarrow}{\mathbf{y}} + \mathbf{K}_d \mathbf{d}_t - \mathbf{K}_x \mathbf{x}(k)\right] \\ &= \left[\underline{\mathbf{K}}_y \underset{\rightarrow}{\mathbf{y}} + \underline{\mathbf{K}}_d \mathbf{d}_t - \underline{\mathbf{K}}_x \mathbf{x}(k)\right] \end{aligned} \tag{4.24}$$

The MPC structure with DSM as soften constraints is shown in Figure 4-9.

Despite the simplicity of the direct optimization algorithm, it needs much memory space because the matrices usually have large dimensions. Moreover, the problem could be numerically unstable when the horizons are very large. The derivation of (4.19) is explained in Appendix C.

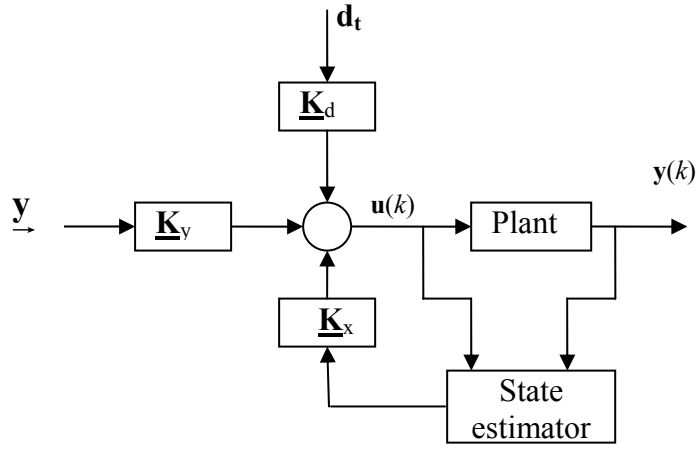


Figure 4-9: Block diagram of MPC with soften constraints

b) Dynamic programming

The solution of the control problem by applying dynamic programming is given by the

affine control law [205] but without integral action in the form of

$$\mathbf{K}_1(j) = \mathbf{K}(N-1)\left[\mathbf{u}_w(k) - \left[\mathbf{M} + \mathbf{B}^{\mathrm{T}}\ \mathbf{P}(j)\ \mathbf{A}\right]\mathbf{x}(k)\right] \tag{4.25}$$

where $\mathbf{u}_w$ represents the control vector due to the reference output ($\mathbf{y}_r$) and $\mathbf{u}_x$ the control action based on the state feedback

$$\mathbf{K}(N-1) = \left[\overline{\mathbf{R}} + \mathbf{B}^{\mathrm{T}}\ \mathbf{P}(N-1)\ \mathbf{B}\right]^{-1},$$

where $\overline{\mathbf{R}} = \mathbf{R}_j + \mathbf{D}^{\mathrm{T}}\overline{\mathbf{Q}}_{N-j}\mathbf{D}$. $\mathbf{P}(N\text{-}1)$ is calculated by solving the Riccati Difference Equation (RDE)

$$\mathbf{P}(j+1) = \mathbf{Q} + \mathbf{A}^T(j)\mathbf{A} - \left[\mathbf{M} + \mathbf{B}^T\mathbf{P}(j)\mathbf{A}\right]^{\mathrm{T}}\mathbf{K}_1(j) \text{ for } j = 1,\ldots,N\text{-}2$$

and a specific $\mathbf{P}(0) = \overline{\mathbf{C}}^{\mathrm{T}}\overline{\mathbf{Q}}_N\overline{\mathbf{C}}$. $\mathbf{K}_1$ is calculated from

$$\mathbf{K}_1(j) = \mathbf{K}(j)\left[\mathbf{M} + \mathbf{B}^{\mathrm{T}}\ \mathbf{P}(j)\ \mathbf{A}\right]^{-1},$$

where $\mathbf{M} = \overline{\mathbf{D}}^{\mathrm{T}}\overline{\mathbf{Q}}_{1j}\overline{\mathbf{C}}$ and $\mathbf{Q} = \overline{\mathbf{C}}^{\mathrm{T}}\overline{\mathbf{Q}}_{1j}\overline{\mathbf{C}}$. The matrices $\overline{\mathbf{C}}$ and $\overline{\mathbf{D}}$ are obtained from

$$\overline{\mathbf{C}} = \begin{bmatrix}\mathbf{C}\\ \mathbf{D}_a\end{bmatrix} \in \Re^{(m+q)\times n} \text{ and } \overline{\mathbf{D}} = \begin{bmatrix}\mathbf{D}\\ 0\end{bmatrix} \in \Re^{m+q}.$$

The matrices $\overline{\mathbf{Q}}_{1j}$ and $\mathbf{R}_j$ are defined as

$$\overline{\mathbf{Q}}_{1j} = \begin{cases}\overline{\mathbf{Q}}_{N-j} & \forall\, 1 \le j \le N - N1 - 1\\ 0 & \forall\, N - N_1 \le j \le N-1\end{cases}$$

$$\mathbf{R}_j = \begin{cases}\infty\mathbf{I} & \forall\, 1 \le j \le N - N_u - 1\\ \mathbf{R}_{N-j} & \forall\, N - N_u \le j \le N-1\end{cases}$$

The control vector $\mathbf{u}_w(k)$ is given by

$$\mathbf{u}_w(k) = \mathbf{D}_d^T\overline{\mathbf{Q}}_{1j}\mathbf{w}(k) + \mathbf{B}^{\mathrm{T}}\mathbf{p}(N\text{-}1)$$

where $w(k) = \begin{bmatrix}\mathbf{y}_r(k)\\ \mathbf{d}_c\end{bmatrix} \in \Re^{m+q}$; $\mathbf{p}(N\text{-}1)$ is obtained from

$$\mathbf{p}(j+1) = \left[\overline{\mathbf{C}} - \overline{\mathbf{D}}\mathbf{K}_1(j)\right]^{\mathrm{T}}\overline{\mathbf{Q}}_{1j}\left[\mathbf{w}(k+N-(j+1))\right] + \left[\mathbf{A} - \mathbf{B}\mathbf{K}_1(j)\right]^T\mathbf{p}(j)$$

and $\mathbf{p}(0) = \left[\overline{\mathbf{C}} - \overline{\mathbf{D}}\mathbf{K}_1(j)\right]^{\mathrm{T}}\overline{\mathbf{Q}}_N\left[\mathbf{w}(k+N)\right]$.

The solution is feasible if the control law calculated in (4.25) satisfies the constraints of control signal $\mathbf{u}(i+k)$. Otherwise, the objective function parameters should be adapted and the optimization problem has to be solved again until the control constraints are satisfied.

The advantage of using dynamic programming optimization instead of direct optimization is that the matrices dimensions are smaller. However, the number of calculation steps is increased.

The main advantage of using softening DSM constraints is that the control law has a fixed structure as in the case of no constraints (Figure 4-9), while its parameters change according to the distance vector (**d**)

4.3.2.1.2 Adapting weight method

As shown in *Example* 4.2, changing the weight matrices could be used to find a feasible solution. Therefore, a suggested method that adapts the MPC parameters in order to find a feasible solution and use a fixed controller structure is introduced in this section. The suggested method to find an optimal solution of the objective function (4.15) subject to (4.14) is based on solving the problem without constraints and tuning the weight matrices **Q** and **R** in order to satisfy the constraints.

The control law without constraints using direct optimization is in the form of (4.18).

By incorporating (4.18) in (4.15) then the condition for feasible solution will be:

$$\mathbf{d}_t + \mathbf{D}_a \hat{\mathbf{x}}(k) \geq \mathbf{D}_b \left(\mathbf{K}_y \underset{\rightarrow}{\mathbf{y}} - \mathbf{K}_x \hat{\mathbf{x}}(k) \right) \tag{4.26}$$

The problem here is to solve Matrix Inequality (MI) (4.26) to find the weighting matrices $\mathbf{Q}_t$ and $\mathbf{R}_t$. However, equation (4.26) is not easy to be solved because it is a nonlinear MI. Therefore, (4.26) can be satisfied by tuning $\mathbf{Q}_t$ and $\mathbf{R}_t$ around its nominal value using adaptive algorithm introduced in *Section* 4.2. Unfortunately, the rate of changes of DSM with respect to $\mathbf{Q}_t$ and $\mathbf{R}_t$ is difficult to obtain. Hence, the adaptive algorithm of *Section* 4.2 can not be easily applied in some cases. Consequently, a simplified algorithm is proposed in the following:

1. Determine $\mathbf{Q}_{to}$ and $\mathbf{R}_{to}$, which satisfy the desired performance at nominal situation.
2. Let $\mathbf{Q}_t(j) = \mathbf{Q}_{to}$ and $\mathbf{R}_t(j) = \mathbf{R}_{to}$; where j is the iteration number.
3. Calculate the control vector (4.18) based on $\mathbf{Q}_t(j)$ and $\mathbf{R}_t(j)$.

4. Check MI (4.26). if $\mathbf{Q}_t(j)$ and $\mathbf{R}_t(j)$ do not satisfy (4.23), then calculate the new weights based on the following equations:

$$\begin{aligned}\mathbf{Q}_t(j) &= \mathbf{Q}_t(j-1) + \alpha_1 \Delta\mathbf{Q} \\ \mathbf{R}_t(j) &= \mathbf{R}_t(j-1) + \alpha_1 \Delta\mathbf{R}\end{aligned} \tag{4.27}$$

5. Repeat step 3 and 4 until (4.26) satisfied or at least $\mathbf{x}(k+1)$ moves in the direction of Φ; where $\mathbf{Q}_t(j) \geq 0$ and $\mathbf{R}_t(j) \geq 0$.
6. The values of $\mathbf{Q}_t$ and $\mathbf{R}_t$, which satisfy (4.26) are the optimal weights of (4.14) that satisfy a feasible solution.

where $\alpha_1 \in \Re$ and $\alpha_2 \in \Re$ are the adaptation parameters. $\Delta\mathbf{Q}$ and $\Delta\mathbf{R}$ are the incremental weight matrices for $\mathbf{Q}$ and $\mathbf{R}$ respectively.

The values $(\Delta\mathbf{Q}, \Delta\mathbf{R}) \in \{(\Delta\mathbf{Q}_i, \Delta\mathbf{R}_i)\}$, where $\Delta\mathbf{Q}_i$ and $\Delta\mathbf{R}_i$ are the incremental weight matrices, which reduce the distance between the current state and the maximum violated constraint number i at instance k; $i \in \{1,2,\ldots,q\}$ and q is the number of constraints. These matrices $(\Delta\mathbf{Q}_i, \Delta\mathbf{R}_i)$ are designed off-line for each constraint individually.

The adaptation equation (2.27) can be replaced with

$$\begin{aligned}\mathbf{Q}_i(j) &= diag[\beta_1, \beta_2, \ldots, \beta_m]\ \mathbf{Q}_i(j\text{-}1) \\ \mathbf{R}_i(j) &= diag[\gamma_1, \gamma_2, \ldots, \gamma_m]\ \mathbf{R}_i(j\text{-}1)\end{aligned} \tag{4.28}$$

In order to avoid negative gain matrices in addition to a more simplification in the algorithm

where $\beta_l \geq 0$, $l \in \{1,\ldots,m\}$ and m is the number of outputs; $\gamma_p \geq 0$, $p \in \{1,\ldots,r,\}$ and r is the number of input; $\mathbf{Q}_i$ and $\mathbf{R}_i$ are the weight matrices at prediction horizon number i. β_l and γ_p are chosen based on the violated constraints. β_l and γ_p can be determined off-line for each constraint, and they are selected one line from the set of pre-designed values.

Although the DC motor is a simple example and a SISO system, it illustrates the effectiveness of each controller design method. It is a good example to compare between the different methods of MPC solution discussed above, since it is required to maintain DSM positive not only at steady state but also at transient in addition to obtain a faster response. Therefore, the introduced methods in this section are implemented on DC motor example.

Example 4.3

Consider the example of DC motor in *Example* 4.1, and the reference output is computed using a reference model, a first order system, in the form of

$$Y(s) = \frac{1}{1+0.5s} R(s) \tag{4.29}$$

where Y(s) and R(s) are the Laplace transform of the output and the input respectively, and s is the Laplace operator, Figure 4-10 shows the state trajectories and responses using MPC without constraints for 2 units a step input. The parameters of the controller are chosen as Q=70, R=0.01, N_1=1, N_u=6, and N= 10.

The response of MPC controller without considering DSM (Figure 4-10c) is accepted w.r.t. the error and rise-time but the state trajectory lies outside the safe boundaries at transient (Figure 4-10a) i.e. DSM is negative. The response is almost the same as in case of fixed parameters PID (Figure 4-2). To improve DSM at transient time, the controller should be redesigned according to DSM.

Figure 4-11 shows the response using a predictive controller with DSM as hard constraint. The controller parameters are chosen as Q=30, R=0.01, N_1=1, N_u=6, and N= 10. It is the best response among the other responses but the computation effort is high. In addition, the feasibility solution methods discussed before are not considered. The controller parameters are chosen manually, which produce a feasible solution in the whole operation time (transient and steady state).

Figure 4-12 shows the response using MPC with softening the constraints of DSM (*Section* 4.3.2.1.1). The DSM is improved. The controller parameters are $\mathbf{P}_0$=*diag*(300,300,0,0,0,0), $\mathbf{Q}_i$=[70], $\mathbf{R}_i$=[0.01], N=10, N_1=1, and N_u=6.

Figure 4-13 shows the response using MPC with adapted weight based on DSM (*Section* 4.3.2.1.2). The DSM is improved but the response is faster than the previous response. The controller parameters are, $\mathbf{Q}_i$=[70], $\mathbf{R}_i$=[0.01], N=10, N_1=1, N_u=6, β_i=0.95, and γ_i=1.4.

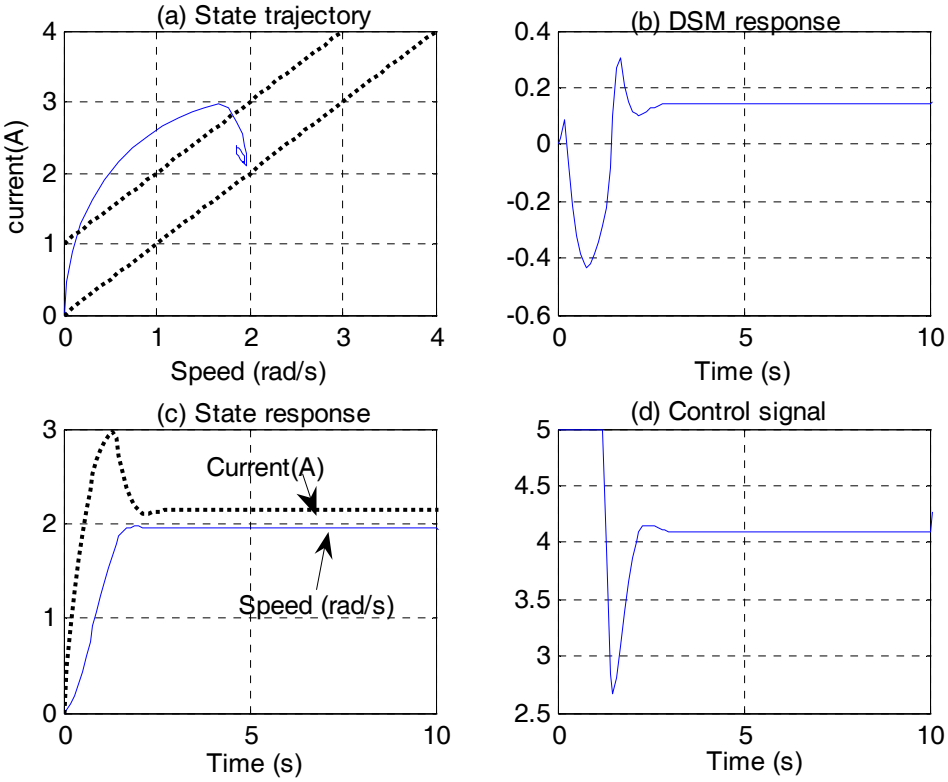


Figure 4-10: MPC without DSM

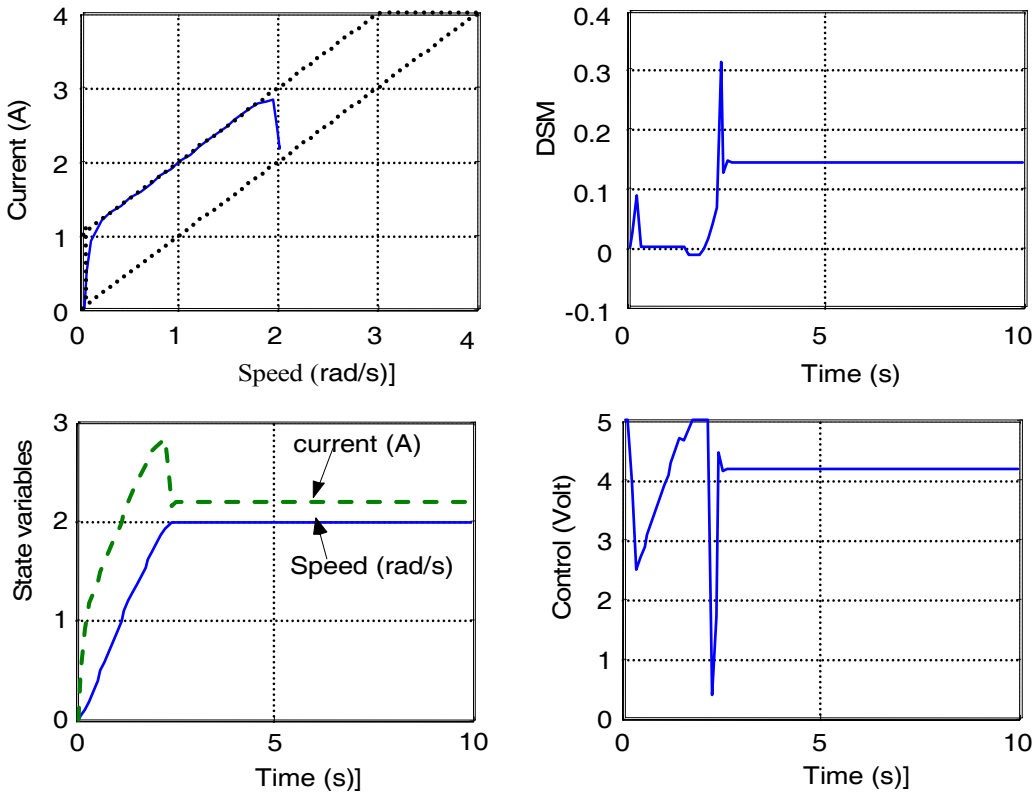


Figure 4-11: MPC with DSM as hard constraints

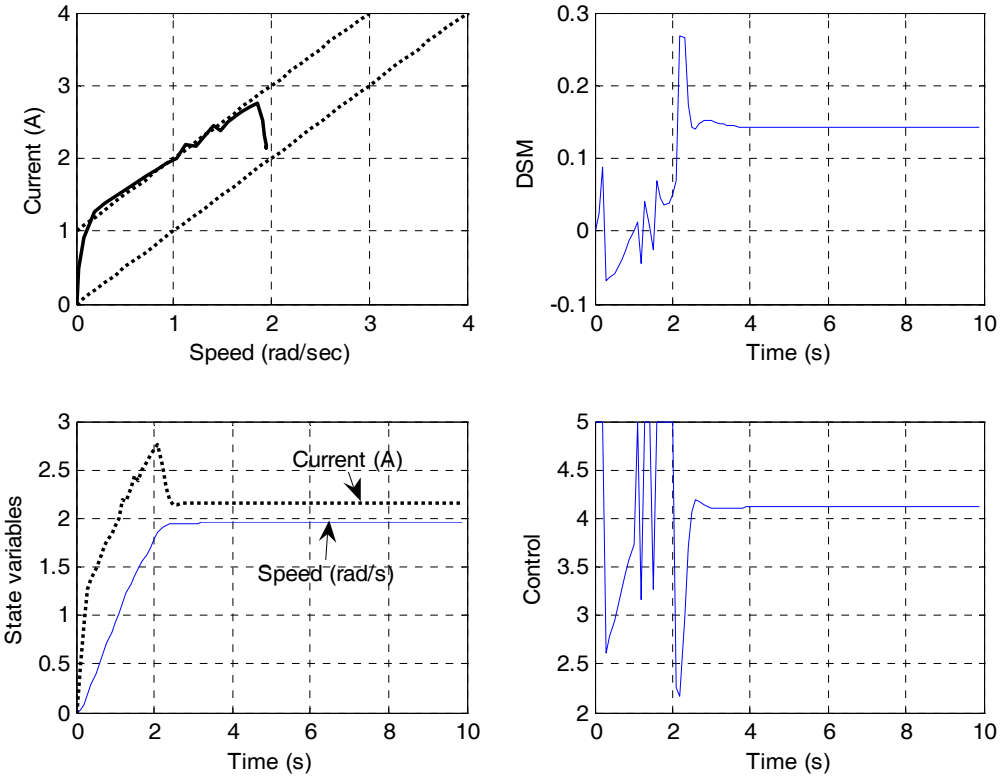


Figure 4-12: MPC with DSM as soft-constraints

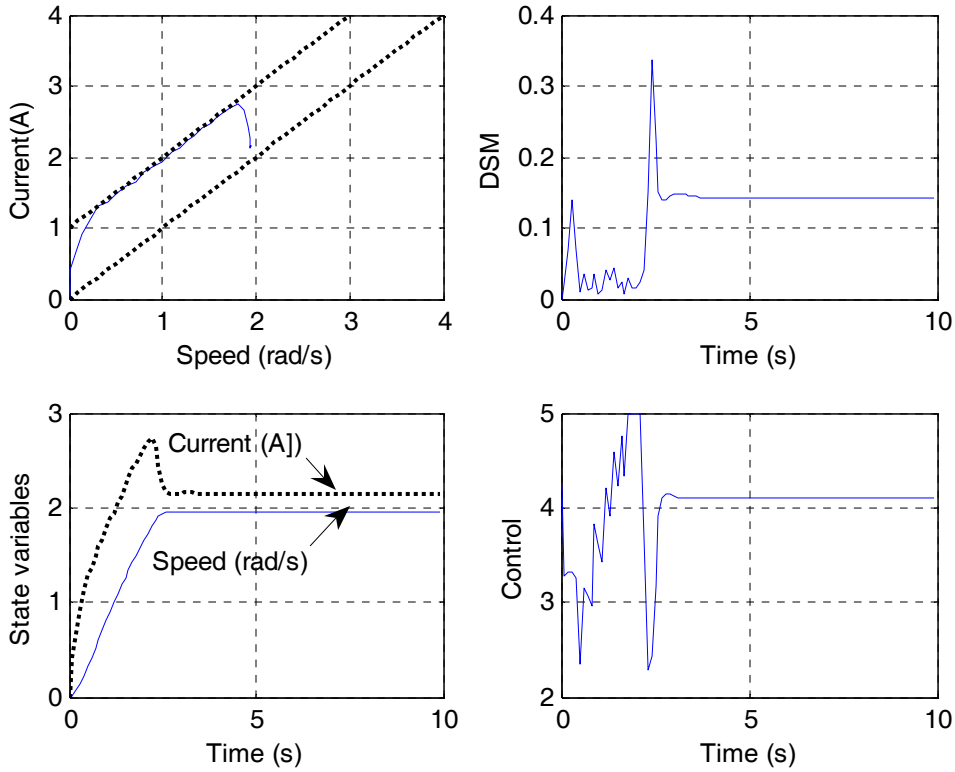


Figure 4-13: MPC with adapted weight

4.4 Frame Work of Fault Detection and Performance Recovery System

An FTC system is a performance recovery controller due to faults. Design techniques for FTC system can be classified as passive and active (PFTC and AFTC) [102], [103], [89]. A PFTC system may tolerate only a limited number of faults, which are assumed to be known prior to the design of the controller. Once the controller is designed, it can compensate for anticipated faults without any access of on-line fault information. PFTC systems treat the faults as if they were sources of modeling uncertainty. AFTC either compensates the effect of faults by selecting a pre-computed control law, or by synthesizing a new control law in real-time. Both methods need a fault detection and identification (FDI) algorithm to identify the fault-induced changes and to reconfigure the control low on-line [89]. To design fault tolerant control (FTC) system, one of the important issues to consider is whether to recover the original system performance or to accept some degree of performance degradation after the occurrence of a fault [102], [198]. The philosophy of recovering the pre-fault system performance is unrealistic for some faults. In practice, because of a faulty part, the degree of the capability of other system components could be significantly reduced. If the design objective is still to maintain the original system performance, the remaining parts may be forced to work beyond the nominal duty to compensate for the handicaps caused by the fault. This situation is highly undesirable in practice due to the physical limitation of the other parts. The consequence of the so-designed FTC system may lead to a worse behavior and still cause further damage. Therefore, trade-off between achievable performance and safety requirements of the operation should be carefully considered in FTC system design not only at steady state but also during transient (dynamic response) [102].

It is known that the information about the fault obtained from FDI, in many cases, is not sufficiently accurate. Moreover, the uncertainties exist in faulty models. Therefore, considering DSM constraints in the recovery controller, particularly in MPC, is useful in order to compensate the unavailable fault information and model uncertainties ([134]-[141]). Thus, the design of an FTC system that achieves an acceptable performance in case of system faults without violating the safety requirements of the overall system is the focus of the work presented here.

The idea of using MPC in FTC is firstly discussed in [103] and implemented on a simulation model of EL AL Flight 1862 in [119]. Both references point out that MPC

provides suitable implementation architecture for fault tolerant control. The representation of both faults and control objective is relatively natural and straightforward in MPC. Some faults can be represented by modifying the constraints in MPC problem definition. Other faults can be represented by modifying the internal model used by MPC [119], [181]. In addition, MPC has a good degree of fault tolerant to some faults; especially actuator faults, under a certain conditions, even if the faults are not detected (PFTC).

The degraded accepted performance can be handled in MPC by changing the objective function or using multi-objective function. According to the definition of DSM, MPC based on DSM constraints satisfies the safety requirements of the system and the accepted degraded performance. In addition, a control system, whose design is based on DSM, can compensate faults when it is difficult or just not possible to find an FDI system that provides full and exact information about the fault. Moreover, not all faults can be anticipated. In such situation, a DSM based FTC system could be very useful to overcome this problem, because controllers based on DSM can maintain a safety operation with acceptable degraded performance even in some cases of unanticipated faults. On the other hand, the proposed FTC system can be applied to active as well as passive FTC.

In the proposed FTC design, three controllers are configured and used for the following scenarios:

- Under normal operating conditions, a nominal controller is designed to guarantee the system's stability and performance in the presence of the modeling uncertainty or disturbance.
- When a fault occurs, the nominal controller should guarantee the system signal boundary by checking DSM until the fault is detected.
- After a fault is detected (DSM < 0), the nominal controller is replaced by MPC controller based on DSM using the nominal system model to compensate the effect of the fault. This controller may recover some control performances.
- If the fault is isolated, then the MPC with DSM is reconfigured again using fault information by selecting the suitable faulty model to improve the control performances.

Remark 1: In the case of PFTC, no FDI system is used. The controller is reconfigured according to the value of DSM. i.e., there are two control configurations; nominal controller and MPC based on DSM using the nominal model.

Remark 2: It is possible that, in some cases, the fault that has occurred cannot be isolated, for instance a fault whose functional structure is completely unknown a priori (i.e., does not belong to the fault set). Then, two controller configurations is used as in case of *Remark* 1.

The switching between the three types of controllers could cause a severe effect on the controlled system. Therefore, a smooth transition from one controller to another is mandatory. Different methods can be employed to smooth the transition such as gradual changing of the input see *Section 4.4.1.2*, and control input signals fusion using for example fuzzy logic or analogical gate circuit [151].

In this approach, the fault is isolated using the algorithm introduced in *Chapter 3*, the reconfigurable controller is designed using MPC based on DSM and a new faulty model selected according to FDI algorithm information. Figure 4-14 shows the general structure for the proposed reconfigurable control scheme, which includes a set of reference models, MPC using safe region constraints, a Multi-Model FDI system employing parameters and state estimation, and a supervisor.

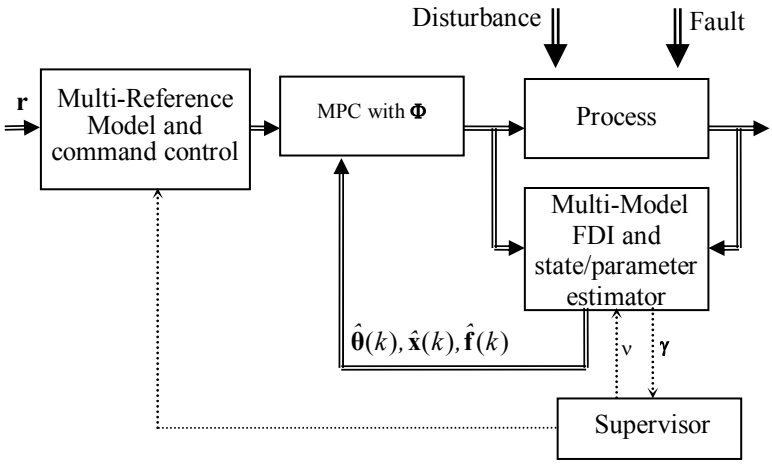


Figure 4-14: Overall structure of the proposed FTC system

4.4.1 Multi-Reference Model and Command Control Block

As mentioned above, it is necessary to reduce the overall system performance to an

acceptable degraded performance in some fault situation. Moreover, in some faulty situation, the system cannot be able to follow the command input and therefore, it should be changed in order to maintain an adequate system operation. The change of the performance can be achieved by modifying the objective function of MPC [204], [119] or by selecting a pre-designed reference model [102]. A combination of both methods is proposed here. The objective function is changed according to the DSM value in order to find a feasible solution of a constrained MPC. A degraded reference model can be used to reduce the effort to find a feasible solution for constrained MPC. The multi-reference and command control block (see Figure 3) is dedicated to select an acceptable degraded performance in case of a specified fault. In case of a specified fault, a new command input is selected in order to maintain the system availability. The design of degraded reference model and command input for actuator faults is addressed in [102]. This method can be generalized in most of the faulty case. This block is activated according to the information received from the supervisor.

4.4.1.1 Degraded Reference Model Design

Assume that the desired closed loop reference model of the system with no fault is represented by

$$\left.\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}_r\mathbf{x}(t) + \mathbf{B}_r\mathbf{r}(t)\\ \mathbf{y}(t) &= \mathbf{C}_r\mathbf{x}(k) + \mathbf{D}_r\mathbf{r}(t)\end{aligned}\right\} \tag{4.30}$$

The corresponding transfer function matrix of the desired reference model is then:

$$\mathbf{T}_r(s) = \mathbf{C}_r(s\mathbf{I} - \mathbf{A}_r)^{-1}\mathbf{B}_r + \mathbf{D}_r. \tag{4.31}$$

Assume that the eignvalues of the closed-loop system are represented as

$$\mathbf{\Gamma}_r = diag[\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_n].$$

After a fault has occurred, it is expected that the closed-loop system eignvalues of the degraded reference model will move towards the imaginary boundary of s-plane to reflect the loss of dynamic performance of the system as will as the reduction in stability margin.

Suppose that the eignvalues of the degraded reference model are represented as

$$\mathbf{\Gamma}_d = \Sigma^{-1}\,\mathbf{\Gamma}_r \tag{4.32}$$

where

$$\boldsymbol{\Sigma} = diag[\beta_1 \quad \beta_2 \quad \cdots \quad \beta_n],\ \beta_i \geq 1,\ \forall \quad i = 1, \cdots, n.$$

The transfer function matrix of the reference model of the degraded system then becomes

$$\mathbf{T}_d(s) = \mathbf{C}_d(s\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d + \mathbf{D}_d \tag{4.34}$$

Assume that $\mathbf{A}_d$ is diagonal then

$$\mathbf{A}_d = \Sigma^{-1}\,\boldsymbol{\Gamma}_r$$

It is important to note that the desired and degraded reference models should have steady-state gain for the purpose of commend input tracing. Therefore,

$$\lim_{s\to 0}\left(\mathbf{C}_d(s\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d + \mathbf{D}_d\right) = \lim_{s\to 0}\left(\mathbf{C}_r(s\mathbf{I} - \mathbf{A}_r)^{-1}\mathbf{B}_r + \mathbf{D}_r\right) \tag{4.35}$$

and

$$\mathbf{C}_d(\mathbf{A}_d)^{-1}\mathbf{B}_d + \mathbf{D}_d = \mathbf{C}_r(\mathbf{A}_r)^{-1}\mathbf{B}_r + \mathbf{D}_r \tag{4.36}$$

If it is assumed that $\mathbf{C}_d$=$\mathbf{C}_r$ and $\mathbf{D}_d$=$\mathbf{D}_r$, then

$$\mathbf{B}_d = \boldsymbol{\Sigma}^{-1}\boldsymbol{\Gamma}_r\mathbf{A}_r^{-1}\mathbf{B}_r \tag{4.37}$$

Hence the degraded reference model can be represented as

$$\left.\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}_d\mathbf{x}(t) + \mathbf{B}_d\mathbf{u}(t)\\ \mathbf{y}(t) &= \mathbf{C}_d\mathbf{x}(t) + \mathbf{D}_d\mathbf{u}(t)\end{aligned}\right\} \tag{4.38}$$

4.4.1.2 Command Input Control

In some faulty situations, the system cannot follow the command input and therefore, it should be changed in order to maintain system availability. A set of different command input for different fault mode can be previously designed and selected on line based on the detected fault. To avoid the transient effect due to the switching between the pre-fault and post fault command input, it is important to change the command input gradually (smooth change). Thus, based on smooth command input switching described in [102], the following modified command input $\mathbf{r}_m$ will be generated based on the selected command input $\mathbf{r}$ as

$$\mathbf{r}_m(k) = \mathbf{r}_m(k-1) + (1 - \mu\, e^{-\tau(k-k_D)})(\mathbf{r}(k) - \mathbf{r}_m(k-1)), \qquad k \geq k_D \tag{4.39}$$

where

$$\mathbf{r}(k) = \begin{cases} \mathbf{r}_n(k) & k < k_D \\ \mathbf{r}_f(k) & k \geq k_D \end{cases} \text{ and } \mu\,(0 \leq \mu \leq 1) \text{ and } \tau > 0$$

are design parameters to provide smooth switching between $\mathbf{r}_n$ and $\mathbf{r}_f$; $\mathbf{r}_n$ and $\mathbf{r}_f$ are the command inputs before and after the fault detection. For large k, $\mathbf{r}_m \to \mathbf{r}_f$. In fact, the command input $\mathbf{r}_m(k)$ is an interpolation between $\mathbf{r}(k)$ and $\mathbf{r}_m(k\text{-}1)$.

Special case

In some situations, it is difficult to find controller parameters, which can track the command input and achieve and the safety performance in addition to the stability. If the highest priority is given to the safety i.e. DSM should be positive, then the DSM index is used to determine the new command input as follows:

When the DSM is negative, at each instant there exists at least one constraint from the constraints set of safe region is violated. In this case, DSM is the distance between the current state and the nearest boundary constraint to the current state, i.e.

$$|\delta(k)| = |\delta_i(k)| = \|v_{io} - \mathbf{x}(k)\|_2$$

Assume that the violated constraint number $i \in \{1,2,\ldots,q\}$ is taken as the reference target to the system, and assume that **D**=0, then the new command will be

$$\mathbf{r}_c(k) = \mathbf{C}v_{io} \tag{4.40}$$

where

$$v_{io} = \arg\left(\min_{\mathbf{v}_i} \|(\mathbf{x} - v_i)\|_2^2 \right) \tag{4.41}$$

subject to

$$\mathbf{v}_i \in \{\mathbf{v} | \phi_i(\mathbf{v}) = 0\}$$

and ϕ_i is the nearest violated boundary constraint.

For a SISO system and linear safety constraints, the command input can be calculated by another way without solving (4.41). The DSM can be defined for the violated constraint i according to (2.12) as

$$\delta_i(k) = c_{1i} - \mathbf{a}_{1i}^T \mathbf{x}(k) \tag{4.42}$$

each vector $\mathbf{a}_{1i}$ can be written as

$$\mathbf{a}_{1i}^{T} = \mathbf{c} + \mathbf{c}_{is}$$

then

$$\begin{aligned}\delta_i(k) &= c_{1i} - (\mathbf{c} + \mathbf{c}_{is})\mathbf{x}(k) \\ &= r_c - y(k)\end{aligned}$$

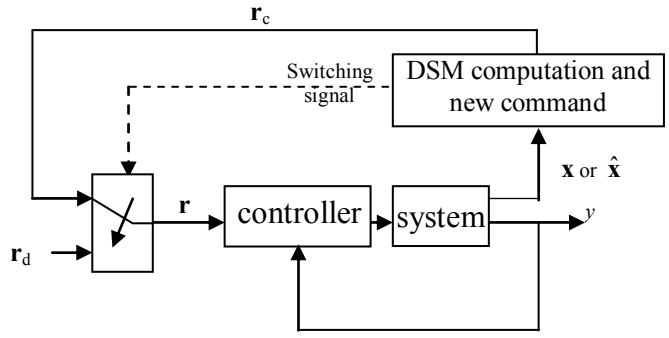
Therefore,

$$r_c(k) = c_i - \mathbf{c}_{is}\mathbf{x}(k) \tag{4.43}$$

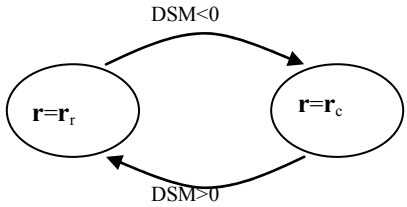
where $r_c(k)$ is the new reference which depends on the current state of the system, y is the output, $\mathbf{a}_{1i}^{T} \in \Re^n$ and $c_{1i} \in \Re$. Equation (4.43) is used in *Example* 2.1 S*ection* 2.3.1.1 where the DSM is selected as the error between the desired and output of the system.

It is clear that the reference is changed or adapted based on the system state location with respect to the safe region until the state reaches the safe region.

Figure 4-15 shows the block diagram of command input selection and adjusting based on DSM.



a) Block diagram of system with different command input



b) Supervisor automata

Figure 4-15: Block diagram of command input selection based on DSM

4.4.2 MPC Employing DSM Block

In the previous section, we discussed the MPC with constraints and showed how DSM can be introduced in MPC. In most faulty systems, the information from FDI is not accurate or sufficient to complete the fault description. Hence, an MPC complemented with DSM will insure a safety operation of the system and can compensate the missed information about the fault and uncertainties in the faulty model. The DSM index is used to specify the priority of the constraints that can be relaxed in order to find a feasible solution and to change the objective function parameters (weights).

4.4.3 Multi-model FDI and State and/or Parameter Estimation

The fault diagnosis and isolation subsystem described in *Chapter 3* is activated when $\delta(t) < 0$ and/or $d\delta(t)/dt < 0$. $d\delta(t)/dt < 0$ means that the state trajectory moves in the direction of unsafe operation. In general, faults can be divided into two types: additive faults, which can be simulated as an unknown external signal, and multiplicative faults, which represent the change in the parameters of the system. In both cases, it is necessary to estimate the unknown external input or the new system parameters, according to the fault type, in order to obtain information about the fault. Therefore, parameters and state estimation are considered as a subsection of the FDI system. The outputs of this block are the estimated state and faulty model parameters, which are submitted to MPC block. Status information is also an output that is sent to the supervisory block.

4.4.4 Supervisory Block

Based on the results of FDI block, the fault information is positive or negative. Positive information means that one of the faulty models can describe the fault. Negative information signifies that it is difficult to represent the fault by one model of the set. Thus, positive information is treated according to the scenario of fault recovery described before, and the reference model and command signal can be selected easily according to the history of the system operation and the operation experience. Contrarily, negative information is not easy to be handled, and therefore the supervisory controller should select the command input as well as the reference model and/or reconfigure the system in order to maintain the system availability. The supervisor receives the data γ from FDI block, which contain the fault type, the output performance index and the DSM value. It sends the signal ν to FDI in order to select the new model,

which has to be used by the MPC. The DSM value plays an important role in supervisory control. It can be considered as a safety index for the recovery control performance, which is described by MPC. It is used in adapting the command input signal and in configuring the reference models. The MPC recovery controller uses the nominal plant model, in case of negative fault information, considering the fault as uncertainties or disturbance in the system until a fault is diagnosed. MPC with DSM constraints can recover the performance to a certain accepted degraded performance. Min-Max MPC [200], [201] can be used in case of unknown fault if it can be considered as an additional unknown input with known bound.

4.5 Conclusions

Designing MPC and adapting PID controlled parameters based on DSM are introduced in this chapter. DSM index is also used in adapting the weights of the objective function of MPC, in order to find a feasible solution and satisfy the safety requirements for a predefined performance. The controller design based on DSM improves safety-assessment especially for safety-critical systems. Simulation results demonstrate the advantage of adapting PID and MPC design based on DSM, which maintains a margin of safety during transient state. FTC scheme based on DSM is introduced; MPC using DSM is discussed in the application of FTC system as well. A degraded performance has to be accepted in some faulty situation in order to increase the system availability. The accepted regarded performance can be achieved by changing either the command input or the model of the reference output or changing both. Therefore, multi-reference and multi-command selections are employed in the proposed FTC system.

CHAPTER 5

REAL-TIME IMPLEMENTATION AND EXPERIMENTS

5.1 Introduction

In the previous chapters, the idea of DSM and its applications were addressed. The algorithms of these applications, particularly DSM in FDI, FTC and performance recovery, are implemented on a laboratory process in order to evaluate the efficacy of DSM and its applications. Therefore, this chapter is devoted to describe the plant and the real-time realization of the DSM applications at the laboratory process. These tasks are implemented in real-time using host/target configuration. Today it is very common to use two computers in a host/target configuration to implement real time systems. The host is a computer not necessary with real-time requirements, in which the developed environment, data visualization and control panel in the form of Graphic User Interface (GUI) reside. The real time system runs in the target, which can be an embedded system based on a board with DSP (Digital Signal Processing), Micro-controller, or a second PC.

The separation between host and target is not necessary for small systems since hard real-time PC operating systems such as QNX, LynxOS, and RT-Linux have solved the problem of deterministic response time of real-times tasks, which exist together with non-real tasks on the same computers [156]. However, if the project has spread, then host/target architecture is more flexibility and modular in addition to reduction in the computation burden. An additional advantage is that the real-time system still works when the host crashes, the matter that increases the reliability of the system [156].

5.2 Plant Description and Real-Time Architecture

The process control laboratory plant uses standard industrial components, which introduce more realism and robustness into the experiments with control application [156]. Figure 5-1 shows an overview of the set-up. The plant consists essentially of two tanks of 100 l, a sump of 300 l, a pump (11kW), a heat exchanger, three control valves, seven on/off valves, six temperature sensors, three level sensors, 3 pressure sensors, and

one flow rate sensor. All these components are industrial ones. Valves are actuated by compressed air and all signals sensor/actuator and the computer systems are transmitted by using 4-20 mA standards. The plant works as follows: water is pumped from the sump and it circulates around the plant following a selected (by on/off valves) path to come back to the sump closing the loop. The pump works at a constant rotational speed and the flow rate is controlled by means of an electric modulating valve. Manual/automatic valves are used to change parameters and select different operating points. Figure 5-2 shows a photo of the manual operation panel, and Figure 5-3 shows the schematic diagram of the plant.

Two additional distinctive features make the plant very interesting [156]:

(a) Water temperature increases very fast because the pump dissipates about 1 kW power in the water closed loop. Thus, a heat exchange unit is necessary to avoid the system ending in thermal runaway.

(b) The two tanks are interconnected each other on the same stream in which the outlet flows are derived, therefore the dynamic model consists not only of differential equations but also of two implicit algebraic equations.

The hybrid characteristics of the plant are analyzed in [156] as follows:

1. Different physical modes: physical systems are normally modeled dynamically by a smooth state-space function noted by

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t), \tag{5.1}$$

where the vector field f is obtained by using principles of conservation of mass, energy and momentum. These systems are usually referred as modes. However, differential equations (i.e. continuous-valued state trajectories) should be frequently supplemented by algebraic implicit equations as well as by discrete equations. Equation (5.1) can be valid only within limitations. In this case, the mixture is given by differential equations and inequalities.

Figure 5-1: overview of laboratory plant

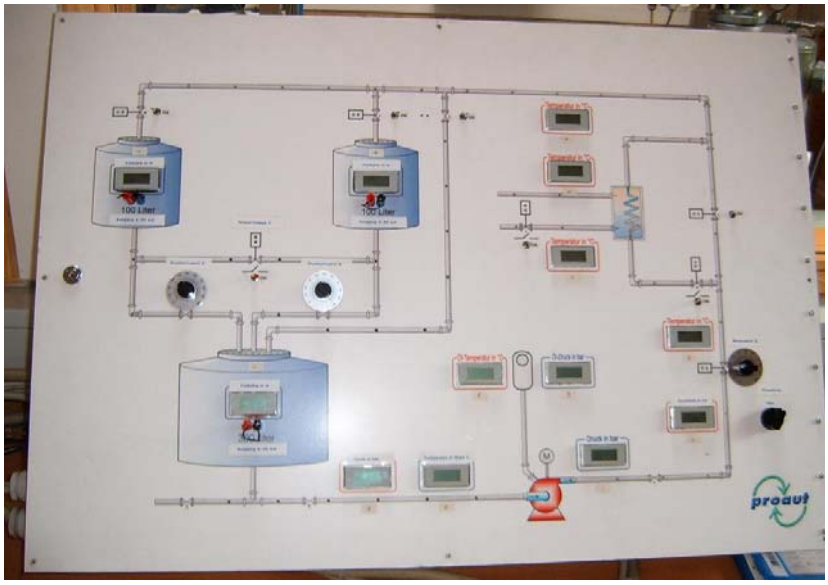


Figure 5-2: Manual operation panel

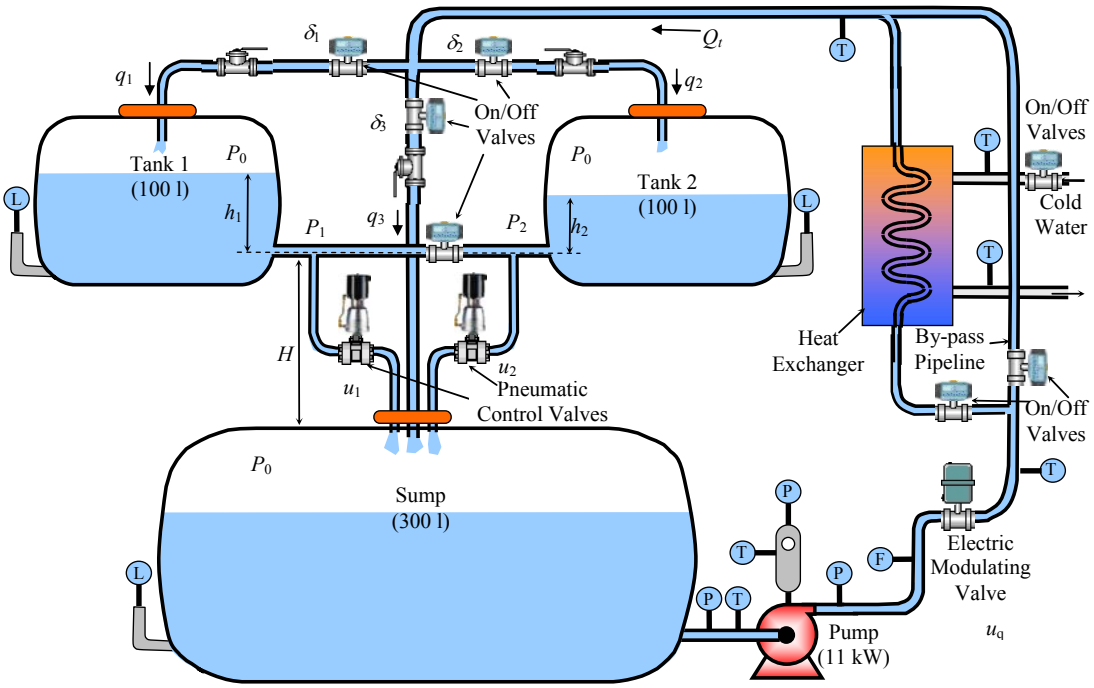


Figure 5-3: Schematic diagram of the two-tank system

Such system is obtained with the laboratory plant by opening and closing the interconnected valve. In addition, different equations are obtained depending on the water level due to the geometry of the tanks.

2. *Discontinuous inputs*: Switches and relays are also found in control systems and they can naturally be modeled as hybrid systems. This is the case of binary valves, which can be found in the plant actuators at the tank inlet flows, for changing the plant configuration and to enable/disable the cooling subsystem.
3. *Discontinuous outputs*: these outputs are given by discrete sensors. They are not explicitly implemented in this plant. However, discrete level indicators in the tanks as well as temperature indicators can be simulated easily using data from analog sensors.
4. *Discontinuous control*: on/off control can be used in this plant to control water level in the tanks if valves on the water inlet stream are used or to control flow temperature. This can be manipulated by three different valves such that a discrete controller based on an automaton can be implemented.

The process has the ability to be controlled either manually using on/off switches and proportional analog tuner or automatically using PC control program. Manual on/off switches are used to test the on/off valves or to change the flow distribution to avoid

over flow in case of fault in the system. Moreover, proportional analog tuners are used to test and regulate the control valves. Therefore, the signals (discrete or continuous) to each valve (on/off or proportional) are from either manual consol or from PC; the selection between them is specified manually using manual/automatic switch.

The PC based control system configuration, in this setup, is configured to satisfy real-time system requirements. The control tasks of the process are achieved using host/target configuration system supported by RT-Lab software. The host is a computer without real time requirment.

RT-LAB is an industrial-grade software package for engineers who use mathematical block diagrams for simulation, control, and related applications. The software use popular programming tools MATLAB/Simulink and MATRIXx/SystemBuild, and works with viewers such as Lab VIEW and Altia, and programming languages including Visual Basic and C++.

RT-LAB allows the user to readily convert *Simulink* or *SystemBuild* models to real-time simulations, via Real-Time Workshop (RTW) or *Autocode*, and run them over one or more PC processors. This is used particularly for Hardware-in-the-Loop (HIL) and rapid control prototyping applications. RT-LAB transparently handles synchronization, user interaction, and real-world interfacing using I/O boards and data exchanges for seamless distributed execution.

5.2.1 Hardware Configuration

RT-LAB software runs on a hardware configuration consisting of command station (host node), compilation node, target nodes, the communication links (real-time and Ethernet), and the I/O boards.

5.2.1.1 The Command Station

The command station is a PC workstation that operates under Windows, and serves as the user interface. The command station allows users to:

- edit and modify models;
- see model data;
- run the original model under its simulation software (Simulink, SystemBuild, etc.);
- generate and separate code;

- control the simulator's Go/Stop sequences.

5.2.1.2 Target nodes

The target nodes are real-time processing and communication computers that use commercial processors interconnected by an Ethernet adapter. These computers can also include a real-time communication interface like FireWire or cLAN (depending on the selected OS), as well as I/O boards for accessing external equipment.

The system may have a single target or multiple target configurations according to the size of the controlled process

5.2.1.2.1 Single target configuration

This configuration, as shown in Figure 5-4, is typically used for rapid control prototyping, in which a single computer runs the plant simulation or control logic. One or more hosts may connect to the target via an Ethernet link. The target can either run QNX or RedHawk Linux for applications where real-time performance is required or for fast simulations, or Windows XP as a simulation accelerator.

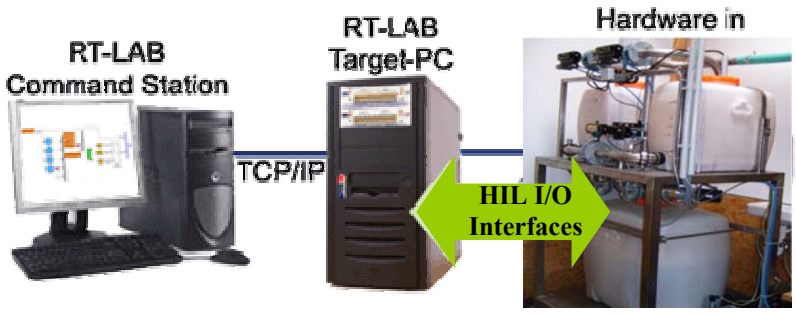


Figure 5-4: Single target Configuration

5.2.1.2.2 Distributed target configuration

The distributed configuration, as shown in Figure 5-5, allows for complex models to be distributed over a cluster of PCs running in parallel. The target nodes in the cluster communicate between each other with low latency protocols such as FireWire, SignalWire or InfiniBand, fast enough to provide reliable communication for real-time applications. The real-time cluster is linked to one or more host stations through a TCP/IP network. The user can build and expand the PC-cluster as needed, then redeploy

the PCs for other applications when the simulation is done. RT-LAB can accommodate up to 64 nodes running in parallel.

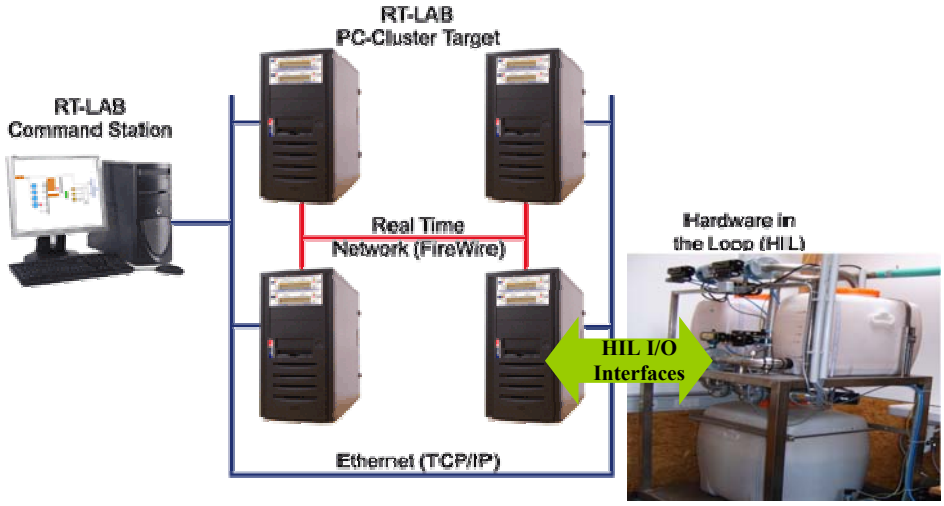


Figure 5-5: Distributed target Configuration

The real-time target nodes perform:

- Real-time execution of the model's simulation;
- Real-time communication between the nodes and I/Os;
- Initialization of the I/O systems;
- Acquisition of the model's internal variables and external outputs through I/O modules;
- Implementation of user-performed online parameters modification;
- Recording data on local hard drive, if desired;
- Supervision of the execution of the model's simulation, and communication with other nodes.

5.2.1.3 Compilation Node

The compilation node, which is one of the target nodes, is used to:

- compile C code;
- load the code onto each target node;
- debug the user's source code (S-function, User Code Block, etc.).

5.2.1.4 Communication

Different types of communication links are employed for the hardware configurations of RT-LAB. The command station and target node(s) communicate with each other using

Ethernet communication links. Both analog and digital I/O boards allow the connection between the target nodes and the external equipement for applications such as HIL. The communication between the target nodes and the synchronization between them and the I/O boards are performed using FireWire (IEEE P-1394) or cLAN interfaces.

Single target configuration (Figure 5-4) is sufficient for the application of the described laboratory process. The command node and target node are commercial PC's with different operating system. A PCI-626 I/O card (from Sensory Company Inc.) is used which satisfies all I/O requirements. Moreover, it is supported by QNX real-time operating system. In this configuration the only communication link used is between the target and command station using Ethernet communication. I/O board is attached directly to the target node without external communication link.

5.2.2 Software Configuration

Software: Integration with Matlab/simulink and Real-Time Workshop (RTW). RTW generates C codes directly from the *Simulink* model and construct a file that can be excuted in real time computer (target).

For more details see [209], [155], and [156].

RT-LAB software is configured on the *Command Station*. Simulations can be run entirely on the command station computer, but they are typically run on one or more target nodes. For real-time simulation, the preferred operating system for the target nodes is QNX.

The starting point for any simulation is a mathematical model of the system components that are to be simulated. Users design and validate a model by analyzing the system to be modeled, and implementing the model in the dynamic simulation software. RT-LAB is designed to automate the execution of simulations for models made with offline dynamic simulation software, like *Simulink* or *SystemBuild*, in a real-time multiprocessing environment. RT-LAB is fully scalable, allowing users to separate mathematical models into blocks to be run in parallel on a cluster of machines, without subtly changing the model's behavior, introducing real-time glitches, or causing deadlocks.

Using block diagrams for programming simplifies the entry of parameters, and guarantees complete and exact documentation of the system being modeled. Once the model is validated, the user separates it into subsystems and inserts appropriate

communication blocks. Each subsystem will be executed by target nodes in RT-Lab's distributed system.

When the C coding and compilation are complete, RT-LAB automatically distributes its calculations among the target nodes, and provides an interface so users can execute the simulation and manipulate the model's parameters. The result is high-performance simulation that can run in parallel and in real-time.

Users can interact with RT-LAB during a simulation by using the console, a command terminal operating under Windows (NT, 2000, or Xp). Communication between the console and the target nodes is performed through a TCP/IP connection. This allows users to save any signal from the model, for viewing or for offline analysis. It is also possible to use the console to modify the model's parameters while the simulation is running.

For the above configuration of RT-Lab, the software in the command station (console) is Windows XP, and the simulation software is Matlab-Simulink to program the simulation and control tasks. The simulation program is coded into C code in the consol unit and transferred to the target node, which has QNX operating system [208]. The target unit compiles and executes the C code file in parallel with the simulation program in the console. The data is transferred on-line between the target and console throw communication Ethernet. In the consol station, the program is written in two main blocks (Consol-Master) as shown in Figure 5-6. The Consol block contains the supervisor control commands, such as manual/automatic switch, operating points, etc., and these commands can be changed and transferred to the target not on-line during the run of the program. In Master block, all control task programs are grouped, such as FDI, FTC, etc., and it cannot be activated on-line. The arrows between the two blocks represent the on-line data transfer.

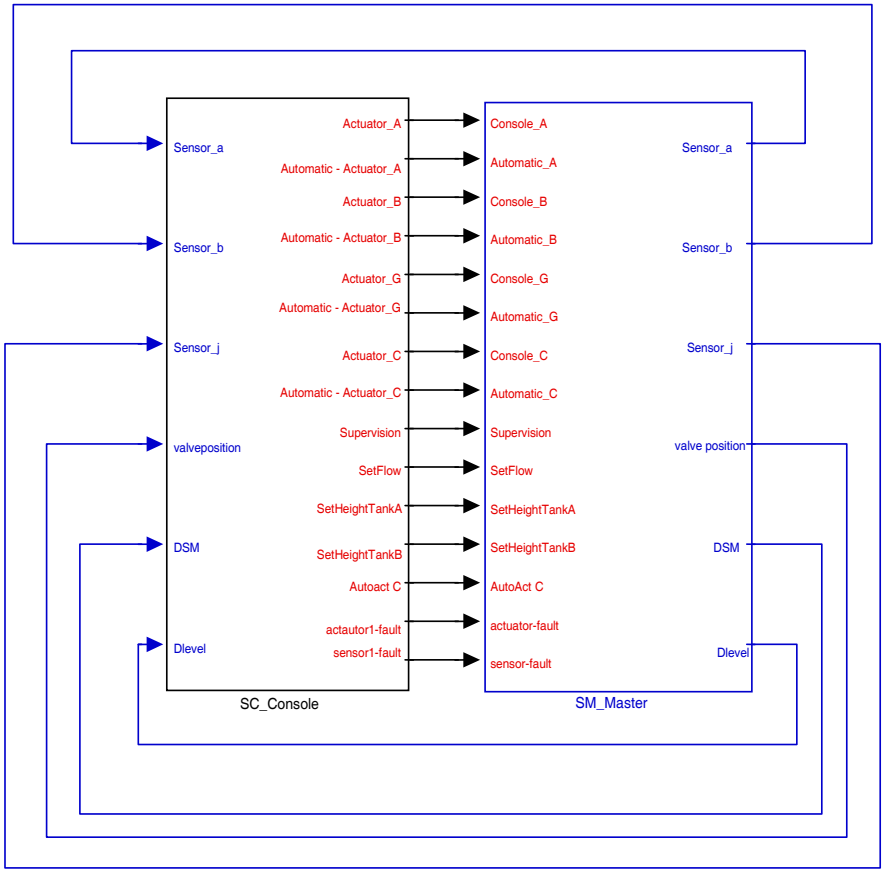


Figure 5-6: Consol-Master connection

5.3 Experimental Results

Practical implementation of the FDI and the FTC based on DSM is carried out on the laboratory setup described before. Therefore, many experiments have been tested in the laboratory process, which are grouped into two categories: 1) fault detection results; 2) performance recovery and controller adjusting.

The experimental setup has a hybrid characteristic due to the combination of discrete and continuous actuators. The complete hybrid model of the two-tank system without considering the heat-exchange unit, as shown in Figure 5-3, has been derived in [156] as follow:

$$\frac{dh_1}{dt} = \frac{1}{A(h_1)}\left(\delta_1 q_{i1} - q_1(t)\right) \tag{5.2}$$

$$\frac{dh_2}{dt} = \frac{1}{A(h_2)}\left(\delta_2 q_{i2} - q_1(t)\right) \tag{5.3}$$

where

$$q_1(t) = C_1\sqrt{\rho g h_1 + (P_0 - P_1)}\,, \tag{5.4}$$

$$q_2(t) = C_2\sqrt{\rho g h_2 + (P_o - P_2)} \tag{5.5}$$

The outflow rates are given by

$$q_{o1}(t) = C_{v1}K_{u1}\sqrt{\rho g H + (P_1 - P_o)}u_1(t) \tag{5.6}$$

$$q_{o2}(t) = C_{v2}K_{u2}\sqrt{\rho g H + (P_2 - P_o)}u_2(t) \tag{5.7}$$

$$Q_t = \delta_1 q_{i1} + \delta_2 q_{i2} + \delta_3 q_{i3} \tag{5.8}$$

q_{i1}, q_{i2}, *and* q_{i3} are the input flow to the tank number 1, 2 and the sump tank respectively; h_1 and h_2 are the levels in the first and second tank respectively; u_1 and u_2 are the input signals to the control valves of each tank; k_{u1} and k_{u2} are constant factors of the valves; C_1 and C_2 are the overall conductance of each tank; C_{v1} and C_{v2} are the conductance of the control valve 1 and 2; H is the height of the pipeline; δ_1, δ_2 and δ_3 are discrete signals $\in \{0,1\}$that represent the state of each discrete valve feeding each tank, 0 means that the valve is closed and contrarily 1 is open; Q_t is the total input flow controlled by the flow valve.

The flow rates must satisfy mass balance equations, i.e.

$$q_1 = q_{1o} + q_{12} \text{ and } q_2 = q_{2o} - q_{12} \tag{5.9}$$

where

$$q_{12} = \delta_{12}\,\mathrm{sgn}(P_1 - P_2)C_{12}\sqrt{|P_1 - P_2|} \tag{5.10}$$

C_{12} is the conductance of the inter-connected valve; δ_{12} is the discrete signal $\in \{0,1\}$ that represent the state of each interconnected valve

Introducing (5.3)-(5.6) and (5.9) in (5.8) implicit equations

$$C_1\sqrt{\rho g h_1 + (P_1 - P_o)} - C_{v1}K_{u1}\sqrt{\rho g H + (P_2 - P_o)}u_1(t) - \delta_{12}\,\mathrm{sgn}(P_1 - P_2)C_{12}\sqrt{|P_1 - P_2|} = 0 \tag{5.11}$$

$$C_2\sqrt{\rho g h_2 + (P_2 - P_o)} - C_{v2}K_{u2}\sqrt{\rho g H + (P_2 - P_o)}u_2(t) + \delta_{12}\,\mathrm{sgn}(P_1 - P_2)c_{12}\sqrt{|P_1 - P_2|} = 0 \tag{5.12}$$

are obtained, which have to be solved for P_1 and P_2.

Since the distances between the outlet pipelines and the corresponding tank are small with respect to the distance between the two tanks, the outlet pipeline of each tank may be considered as it is direct connected to the tank. Therefore, the model of the two-tank system could be approximated as follows:

$$\begin{aligned}
\frac{dh_1}{dt} &= \frac{1}{A(h_1)}\left(\delta_1 q_{i1} - C_{v1}K_{u1}\sqrt{\rho g(h_1 + H)}u_1 - \delta_{12}sig(h_1 - h_2)C_{12}\sqrt{\delta g|h_1 - h_2|}\right) \\
\frac{dh_2}{dt} &= \frac{1}{A(h_2)}\left(\delta_2 q_{i2} - C_{v2}K_{u2}\sqrt{\rho g(h_2 + H)}u_2 + \delta_{12}sig(h_1 - h_2)C_{12}\sqrt{\delta g|h_1 - h_2|}\right)
\end{aligned} \tag{5.13}$$

In our experiments, the setup is set as one-tank or two-tank configuration. The input flow (Q_t) is set to 1 l/sec by controlling the flow value (u_v) either manual or automatic. In one-tank configuration, the discrete signals δ_2 , δ_3 and δ_{12} are set to zero, while δ_1 is set to one, i.e. $Q_t = q1 = 1$ l/s and $q_{12}=0$. The level is controlled through the outflow control valve (u_1). The system is nonlinear and the discrete linearized state space model at the operating point (h_1=0.3m, u_1=50%) of one-tank is shown in Table 5-1. The discrete linear model is a second order that represents the dynamic of the tank and the valve movements.

In two tank configuration, the discrete signals δ_2 ,and δ_3 are set to zero, while δ_1 and δ_{12} are set to one, i.e. $Q_t = q_1 = 1$ l/s and q_{12} is calculated from (5.1). The linearized discrete model of two tank system about the operating point (h_1=h_2=0.3m, u_1=35% and u_2=10%) is shown in Table 5-2. The control input is u_1, and the input u_2 represents the load disturbance or leakage. The controlled variable, in this case, is h_1, while h_2 is floating.

Table 5-1: Linear state-space model of the one-tank system

A	**B**
$\begin{bmatrix} 0.999741 & -6.94e-4 \\ 0 & 0.740818 \end{bmatrix}$	$\begin{bmatrix} -1.0932e-5 \\ 0.25918177 \end{bmatrix}$
C	**D**
$\begin{bmatrix} 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 \end{bmatrix}$

Table 5-2: Linear state-space model of the two-tank system

A	**B**
$\begin{bmatrix} 0.9748 & 0.0019 & -0.0146 \\ -0.1616 & -0.2104 & 0.5555 \\ -2.4323 & -1.1408 & 0.2307 \end{bmatrix}$	$\begin{bmatrix} -0.0004 \\ -0.0105 \\ -0.0173 \end{bmatrix}$
C	**D**
[1 0 0]	[0]

5.3.1 Fault Detection and Isolation Results

The robust FDI algorithm, which is described in *Chapter 3*, is implemented on two-tank system configurations for different types of faults, especially control actuator, leakage and sensor faults. The parameters of the identified system model have uncertainties due to two reasons:

1. The input flow rate (Q_t) is not fixed at 1 l/s, but it varies within the interval [0.92, 1.05] because of the high rate of the pump (11Kw), which makes any small variation in the flow control valve increase the flow rate with bigger amount.
2. The cross section areas of both tanks are not constant due to the geometry of the tank, since they depend on the level height, i.e. $A_i=f(hi)$; $i \in \{1,2\}$.

There is also a disturbance due to the opening of uncontrolled valve of the right tank (u_2) within [0,10%], which represents the load disturbance.

Thus, the system has uncertainties in the model parameters and uncontrolled input disturbance together. The uncertainties are not completely known.

Hence, the FDI system should have the capability to detect different faults and isolate them correctly.

Before applying the FDI algorithm, the safe region of the system operation should be determined. Based on the experimental measurements of the system operation the safe operation region is defined as follow:

1. one-tank system operation

$$dh_1/dt + 0.8\, v_i - 0.08 \le 0;$$

$$dh_1/dt + 0.75\, v_i + 0.14 \ge 0;$$

$$-0.4 \le dh_1/dt \le 0.4; \qquad (5.14)$$

$-0.5 \leq v_i \leq 0.5$,

$0.25 \leq h_1 \leq 0.35$

2. two-tank system operation

The safe operation region is defined by the same constraints of the one-tank configuration, in addition to the following constraint:

$0. \leq (h_1 - h_2) \leq 0.05$.

where the valve opening is normalized within [-0.5, 0.5] i.e. 0.5 means fully opened and -0.5 completely closed. The level rate change (dh_1/dt) is in [mm/s].

Note that, firstly the state vector is $\mathbf{x} = [h_1 \; h_2 \; v_i]^T$ in case of two tank system and $\mathbf{x} = [h_1 \quad v_i]^T$ in case of one tank system; secondary one of the safety variables is *dh/dt*, which is not the state directly, where $dh/dt = f(\mathbf{x})$ and $f: \Re^n \rightarrow \Re$ is a nonlinear function of **x,** therefore *dh/dt* is taken as an independent variable, which can be easily computed from h_1 in order to have linear constraints.

The three fault mode parameters are defined as in *Example* 3.2. In case of leakage fault, the system model as in (3.53) where $\mathbf{b}_a = [1\ 0\ 0]^T$.

5.3.1.1 Actuator fault results

The actuator fault can be either bias or draft fault as discussed in *Chapter 3*. Actuator bias fault is tested in the following experiments.

Consider that an actuator bias fault of 30% opening within the time interval [700s, 1100s] after the system operation. An adapted PID controller is used to control the level of the left tank at 0.3 m. In this experiment, we assume that the fault set is the actuator fault, leakage fault, and fault free.

Figure 5-7 shows the response, and the normalized DSM variation due to the actuator fault bias. Figure 5-8 shows the estimated leakage and actuator fault. Figure 5-9 shows the DSM variation of the actual system compared with DSM of the fault-free, the actuator bias fault, and internal leakage fault models.

Based on the FDI method described in *Chapter* 3, DSM signals of both fault-free model and actual system are negative. Therefore, the fault is detectable. The fault is detected after 20 s. The estimated actuator and leakage faults are within the allowable range. However, the integral error between the actual DSM and the estimated from the

actuator fault model is the smallest one among the other errors of the fault set (leakage fault and the fault-free case), in addition to the threshold value. Thus, the isolated fault is the actuator fault and the estimated value is about 0.25. The error between the estimated actuator fault and the actual one is due to the uncertainties in the model and the nonlinearities.

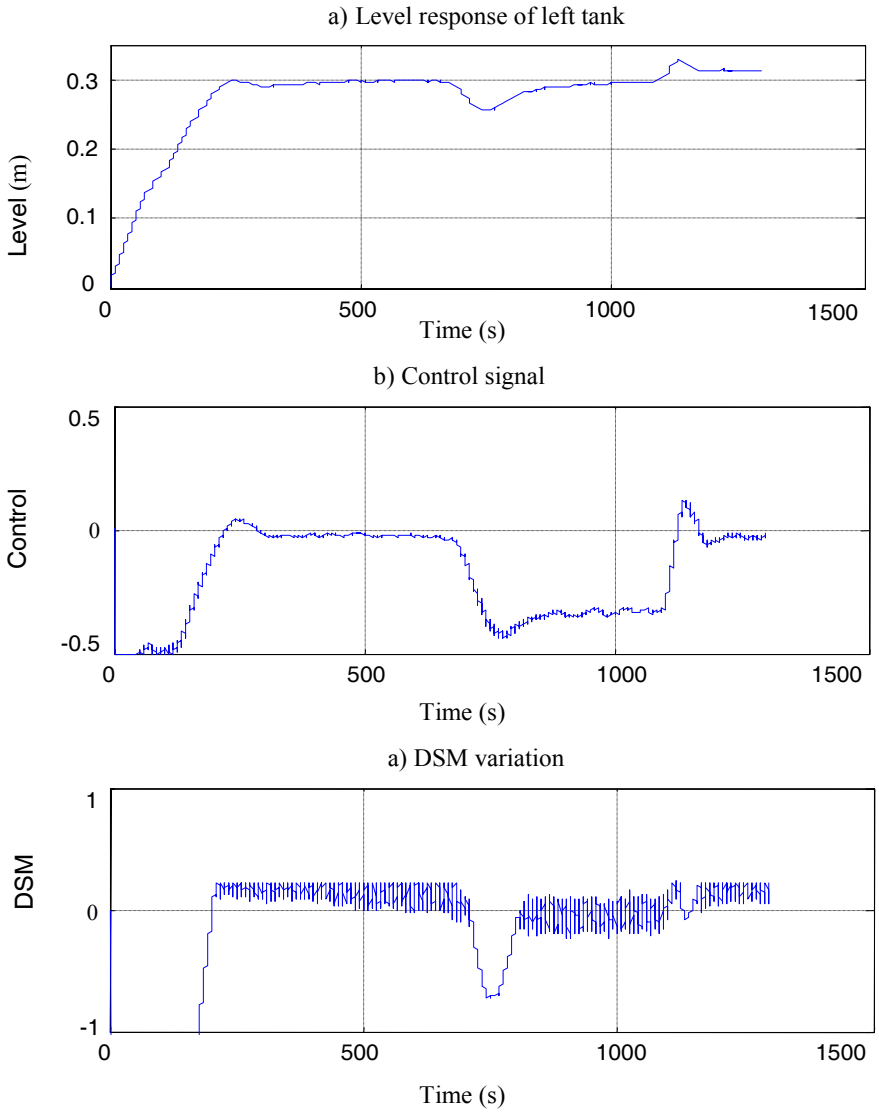


Figure 5-7: Level response and DSM variation in case of actuator fault

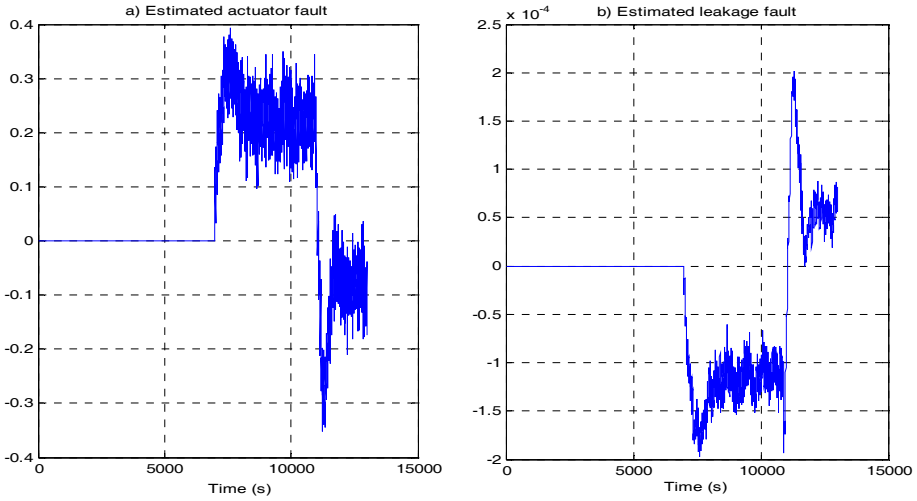


Figure 5-8: Estimated actuator and leakage fault

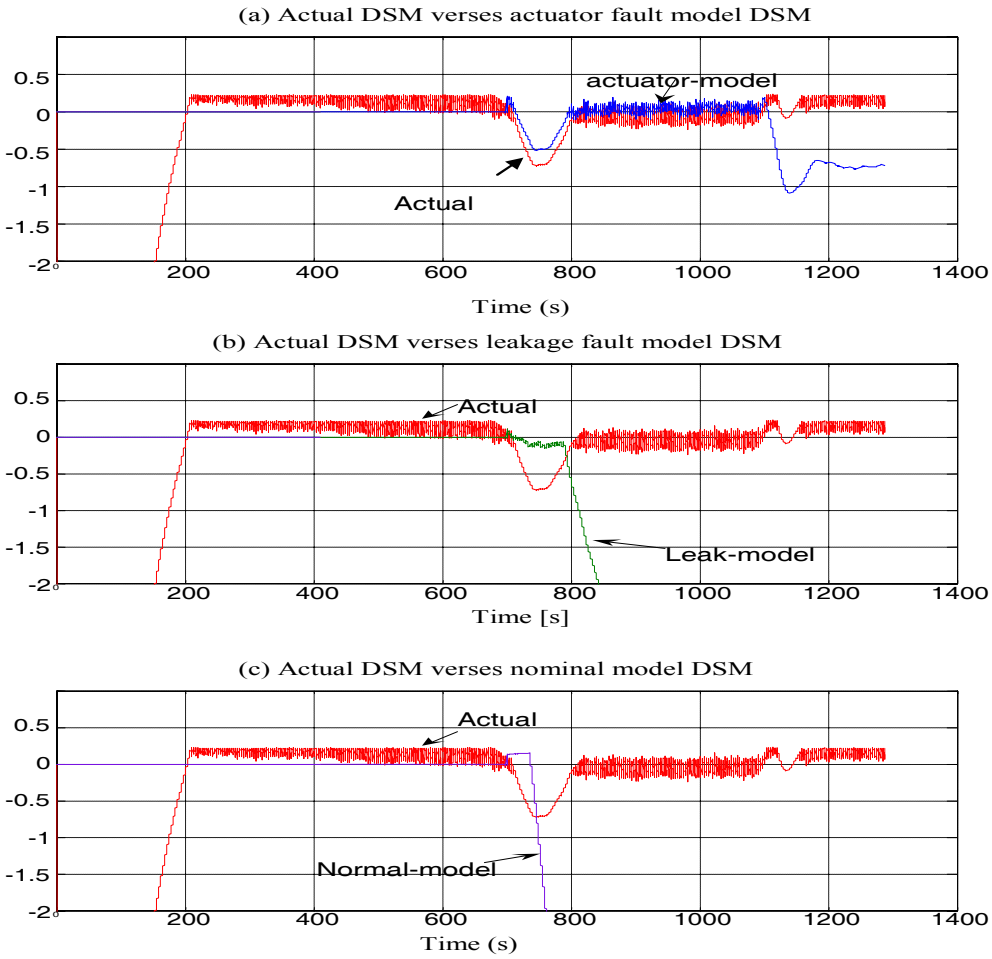


Figure 5-9: DSM variation for different fault modes

The same experiment is repeated, but the desired level height is 0.2 m instead of 0.3 m; the same model of the two-tank system is used in order to check the robustness of fault

detection where the parameters of the model, in this case, differ from the model at 0.3 m operating point. Figure 5-10, Figure 5-11 and Figure 5-12 show the response, estimated faults and DSM variation for different fault mode respectively due to actuator bias fault 20 % closing after 900s. The fault is also detected after 20 s and isolated after 25 s. The estimated leakage fault is positive, which means that its value is out of the allowable range.

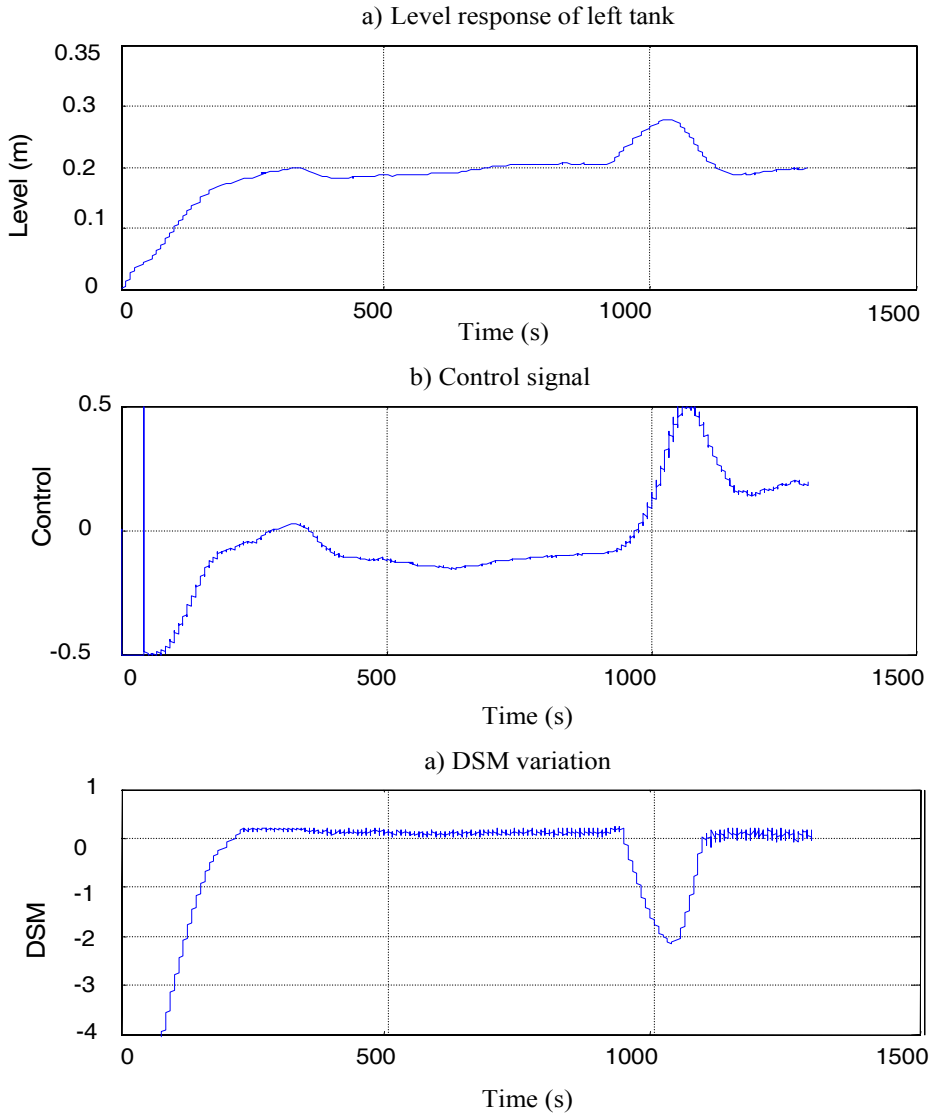


Figure 5-10: Level response and DSM variation in case of actuator fault at 0.2m level

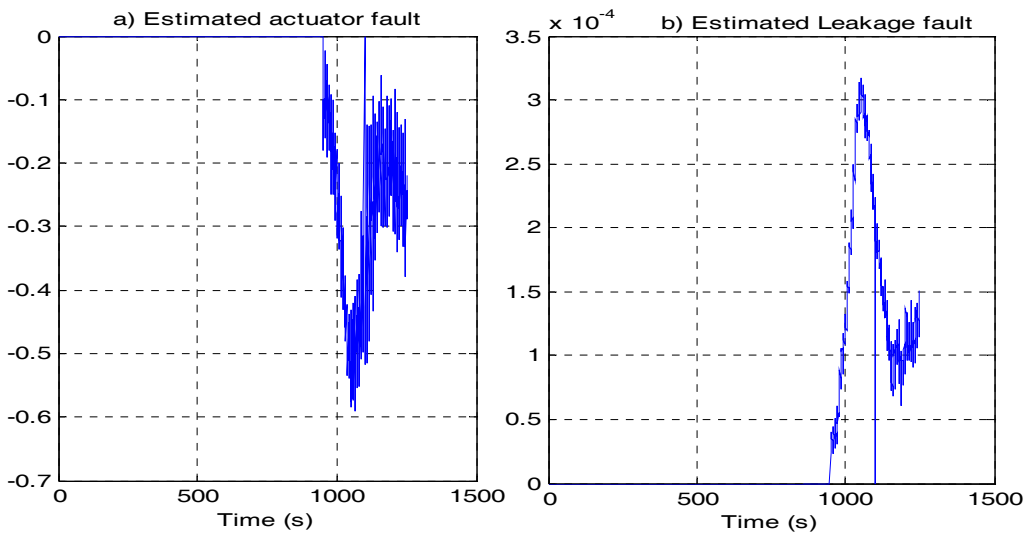


Figure 5-11: estimated actuator and leakage fault in case of actuator fault at 0.2m level

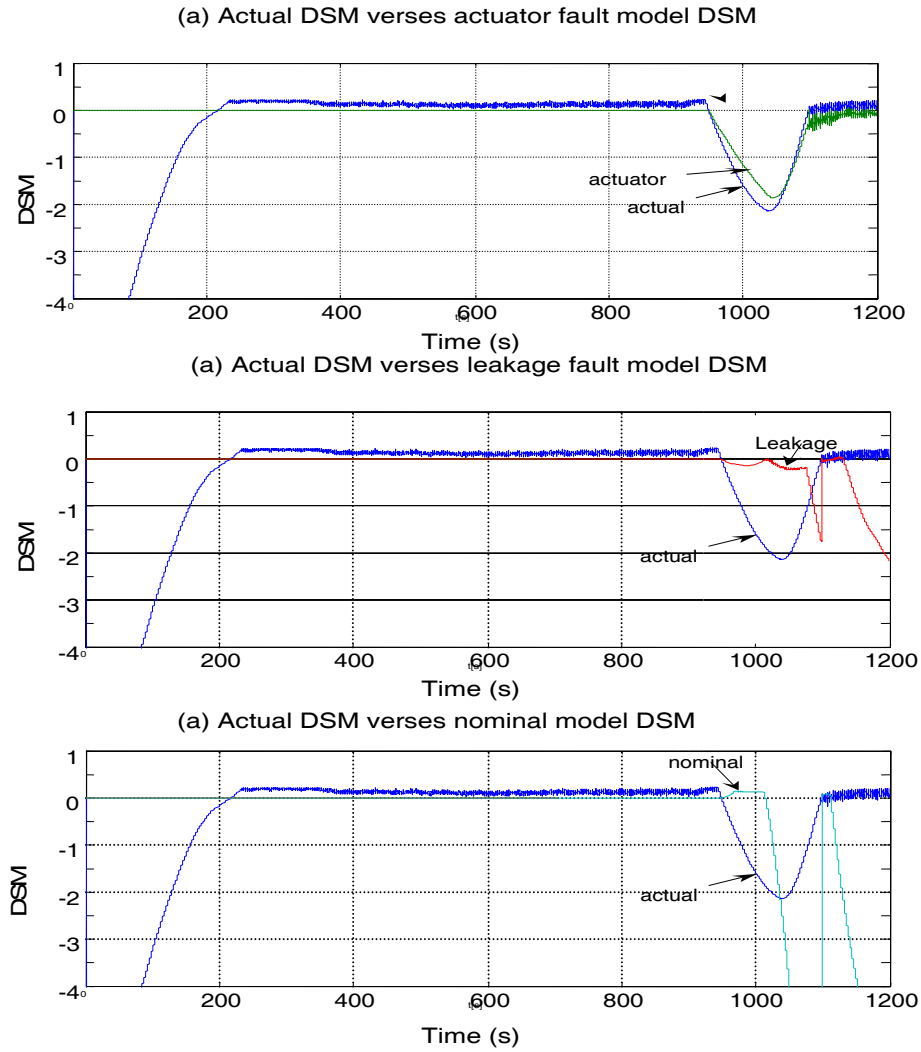
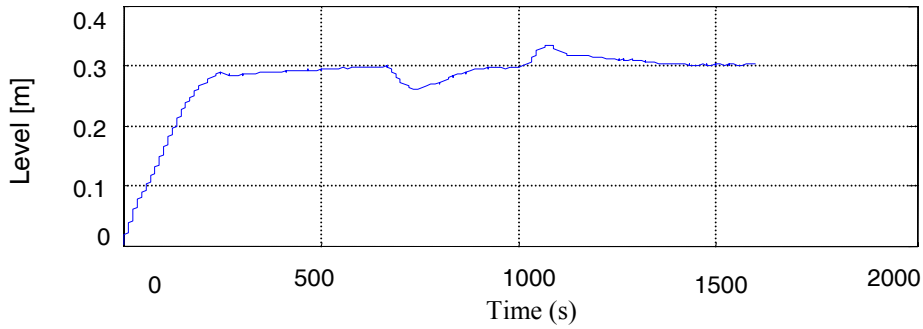


Figure 5-12: DSM variation of different fault mode in case of actuator fault at 0.2m level

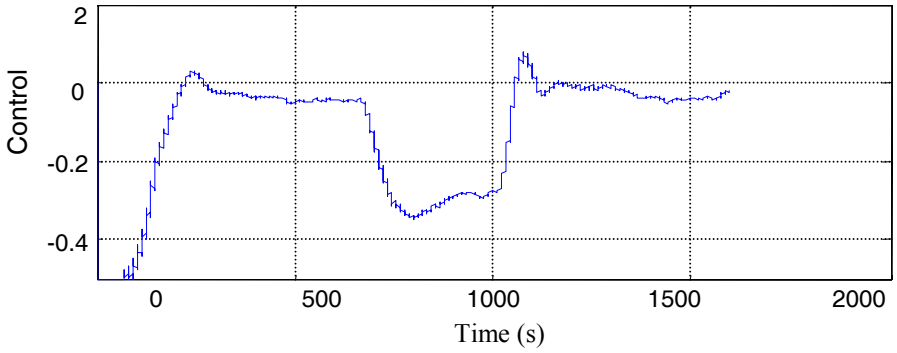
5.3.1.2 Leakage fault results

Figure 5-13 shows the response and DSM variation due to the internal leakage simulated by opening the leakage valve 30% after 700s. Figure 5-14 shows the estimated leakage and actuator fault in case of leakage fault. Figure 5-15 shows the DSM variation of the actual system with respect to the other models of actuator bias, fault-free, and internal leakage. The fault is also detectable, and it is detected after 20 s and isolated after 25 s.

a) Level response of left tank



b) Control signal



a) DSM variation

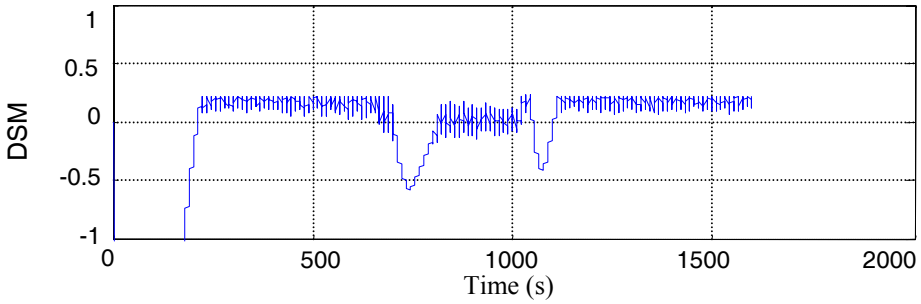


Figure 5-13: Level response and DSM variation in case of leakage fault

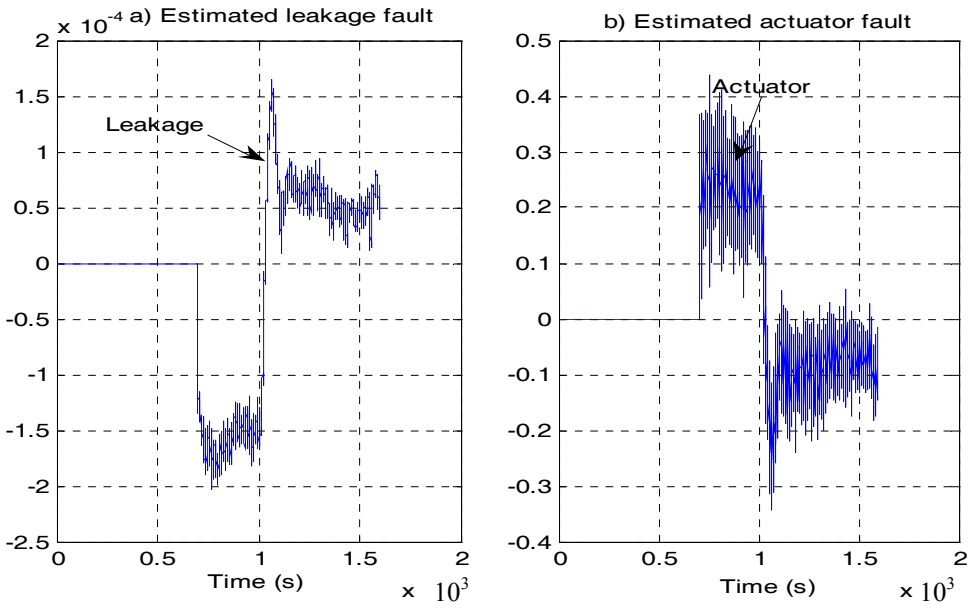


Figure 5-14: Estimated leakage and actuator fault in case of leakage fault

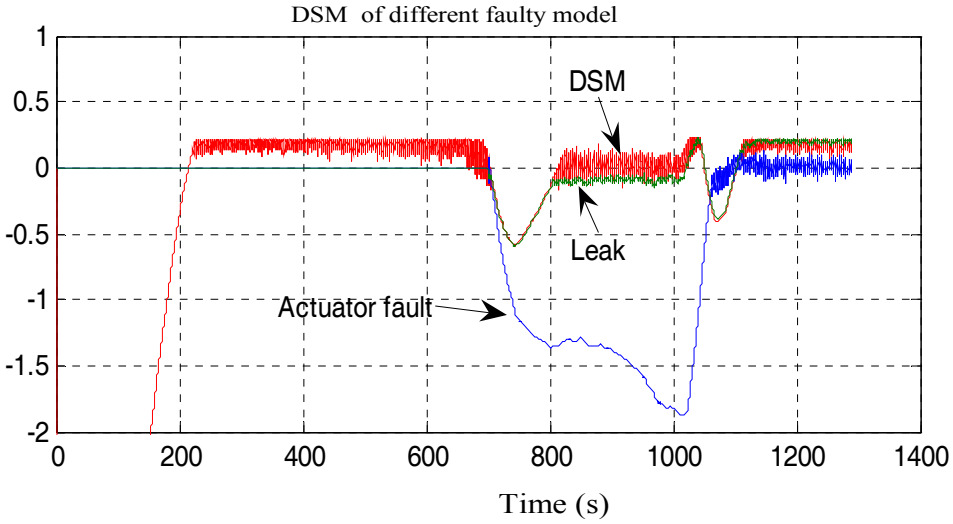


Figure 5-15: DSM variations for different fault mode in case of leakage fault

Table 5-3 summarize the experimental results of the two faults (actuator and leakage)

Table 5-3 Actuator and leakage fault results

	$\hat{f}_a$	$\hat{f}_I$	T_0	T_1	T_2
ABF: $f_a = 0.3$	0.25	0.0001	$> T_{th}$	$\leq T_{th}$	$> T_{th}$
LF: $f_I = -0.0001$	0.26	-.00015	$> T_{th}$	$> T_{th}$	$\leq T_{th}$

where ABF is the abbreviation of actuator bias fault and LF is leakage fault, f_a, and f_l, are the actuator and leakage fault respectively; $\hat{f}_a$ and $\hat{f}_I$ are the estimated ones. T_i is the integral error between actual DSM and the computed one from each faulty model, $i \in \{0,1,2\}$ is the fault mode, and T_{th}=0.1 is a threshold error with integration step N=100.

5.3.1.3 Sensor fault results

Consider that a sensor bias fault about 0.05 exists within the time interval [850s, 1100s] in the level sensor of the left tank. An adapted PID controller is used to control the level of the left tank at 0.3 m. In this experiment, we assume that the fault set is the actuator fault, leakage fault, sensor fault and fault free.

Figure 5-16 shows the response and DSM variation due to a bias in the level sensor of the left tank after 800s. In this experiment, a load disturbance about 10% opining of the u_2 exists from the starting time of the experiments in addition to the input flow disturbance ($\mathbf{Q}_i$), a matter that lead to a change in the model parameters. Figure 5-17 shows the estimated sensor, actuator and leakage faults. Figure 5-18 shows the DSM variation of the actual system and sensor, actuator and internal leakage models. The fault is also detected after 10 s and isolated after 20 s. Table 5-4 summarize the senor fault results

Table 5-4: Sensor fault result

	$\hat{f}_a$	$\hat{f}_l$	$\hat{f}_s$	T_0	T_1	T_2	T_3
Sensor fault bias=0.05	0.1	-0.0001	0.08	$>T_{th}$	$>T_{th}$	$>T_{th}$	$<T_{th}$

where f_a, f_l and f_s, are the actuator, leakage and sensor fault respectively; $\hat{f}_a$, $\hat{f}_I$ and $\hat{f}_s$ are the estimated ones; T_0, T_1, T_2 and T_3 are the integral error between actual DSM and the computed one from nominal, actuator, leakage, and sensor fault models respectively, i.e. 4 fault mode; T_{th}=0.1 is a threshold error with integration step N=100.

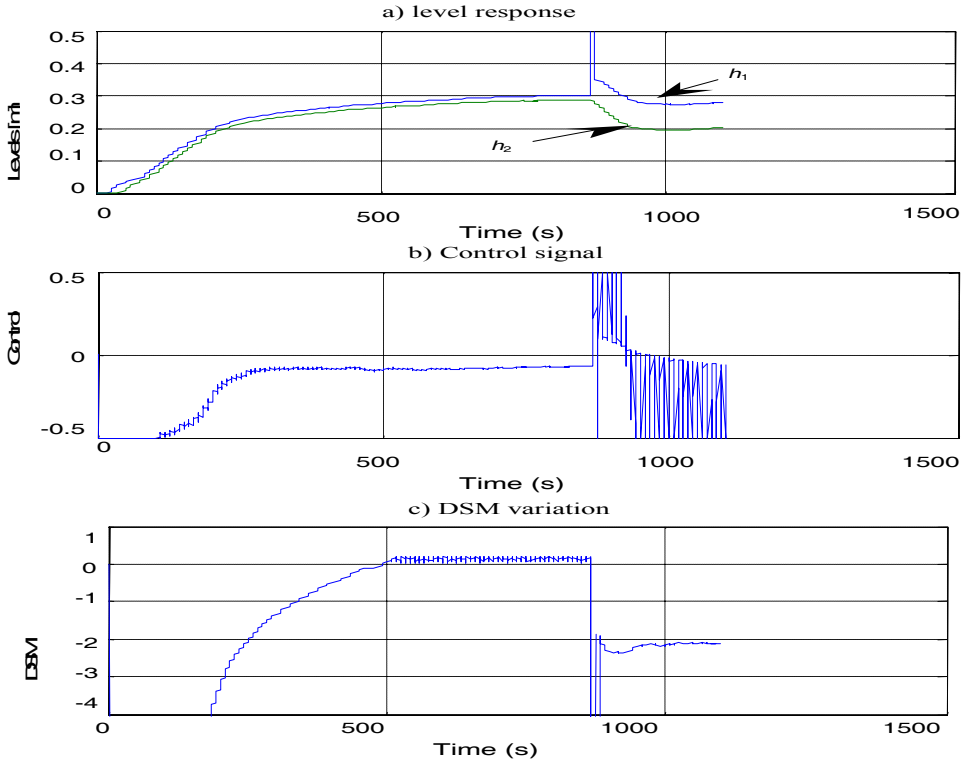


Figure 5-16: Level response and DSM variation in case of sensor fault

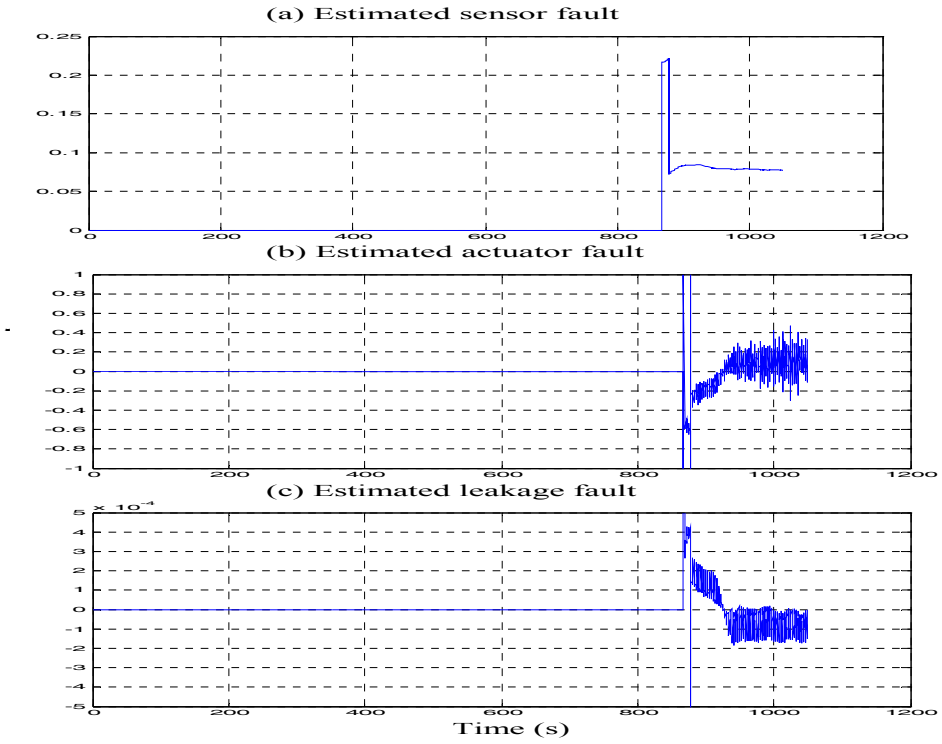


Figure 5-17: Estimated actuator fault in case of sensor fault in case of sensor fault

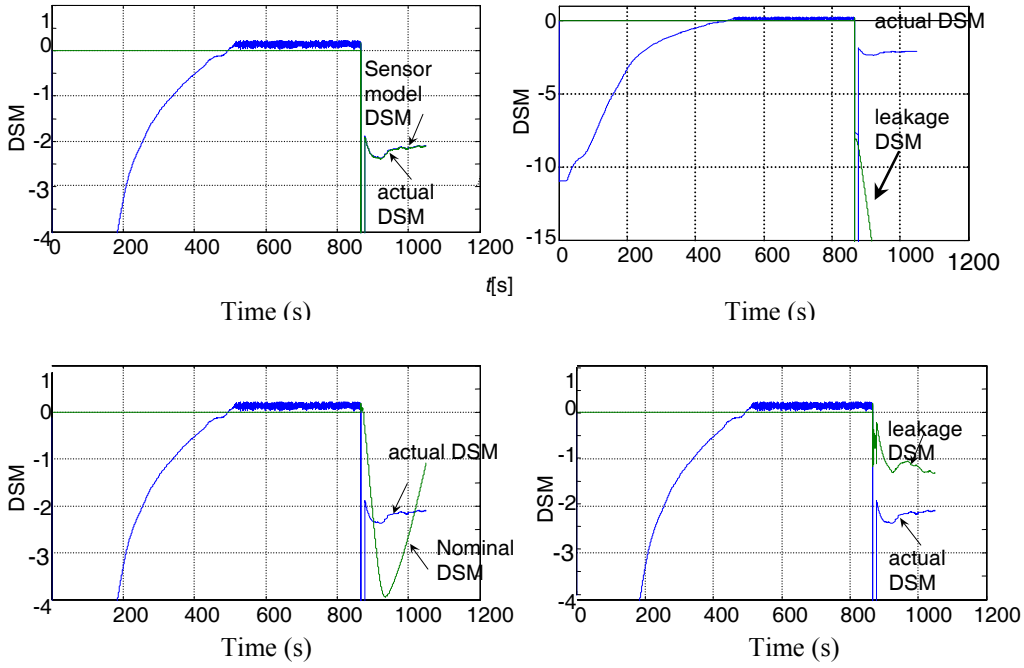


Figure 5-18: DSM variation in case of sensor fault in case of sensor fault

5.3.2 Performance Recovery and Safety Control Results

Different controllers design based on DSM, which are discussed in the previous *Chapter*, are implemented in order to recover the output performance and to maintain the system state inside the safe operation region due to the existence of large disturbance or fault. These methods are implemented practically on the experimental set-up explained before in order to demonstrate the fruitfulness of this design. The experiments are grouped into two groups: 1) performance recovery and safety margin control due to unknown disturbance; 2) FTC system (passive and active).

5.3.2.1 Performance Recovery for Disturbed System

Adapted PID and MPC based on DSM are tested, which are explained in *Chapter 4*.

5.3.2.1.1 Adapted PID controller parameters based on DSM

In this experiment, the plant is configured (Figure 5-3) where the level in the left tank (h) was selected as controlled variable and the control signal u is applied to the left control

valve. On the right tank, the valve was selected at a variable opening to simulate different load disturbance (output flow) of the left tank. The interconnecting valve is commanded according to the following criteria: the valve becomes off, before the level in the left tank reaches the desired value and then on after that (Figure 5-19). At the first instance, the plant behaves as a one-tank system until the level of the left tank reaches a certain steady state limit and two-tank system after the interconnected valve is opened. Figure 5-19 shows the hybrid automaton of this experiment.

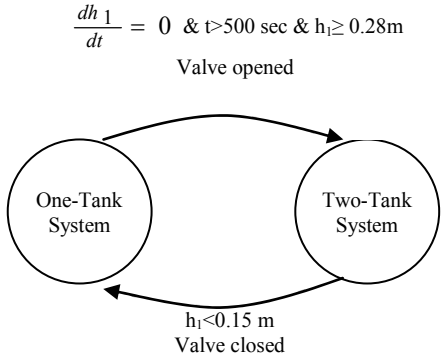


Figure 5-19: Hybrid automaton of two-tank system

Figure 5-20 shows the real time response, control signal and DSM variation using fixed PID controller parameters (K_P=4, K_I=0.08, K_D=0.1), and the disturbance valve was opened with the sequence 0%, 10%, 30%, 50% and 40% respectively, as shown in Figure 5-20a.

Figure 5-21 shows the real-time response and control signal using linear adapted proportional gain of the PID controller as in (2.22) with the same disturbances as Figure 5-20, where α_{i1}=2 and α_{i2}=0. Comparing the two responses (fixed PID parameters and adapted proportional PID), it is clear that in case of one-tank or two-tank system, the system response using adapted PID controller based on safety boundary is better than fixed PID, for either a normal or a disturbed system. The results insure that considering DSM in adapting controller parameters improves system performance.

Figure 5-22 shows real time response using fuzzy adaptation as in (2.24) for the same disturbance sequence as in Figure 5-21. The fuzzy supervisor has one input (deviation from the safe boundary), one output (incremental proportional gain) with input/output membership function shown in Figure 2.20, and Fuzzy allocation matrix shown in Table 2-1. Normalized input and output signal of fuzzy controller can help to generalize the fuzzy supervisor for more than one parameter adaptation.

The level responses of Figure 5-21 and Figure 5-22 have not changed with leakage 10% and 30%, but it began to change with 50% leakage with small rate and recovered at 40% leakage.

It is clear that adapting controller parameters, based on DSM, improves the system output performance and can help in safety control of safety critical system.

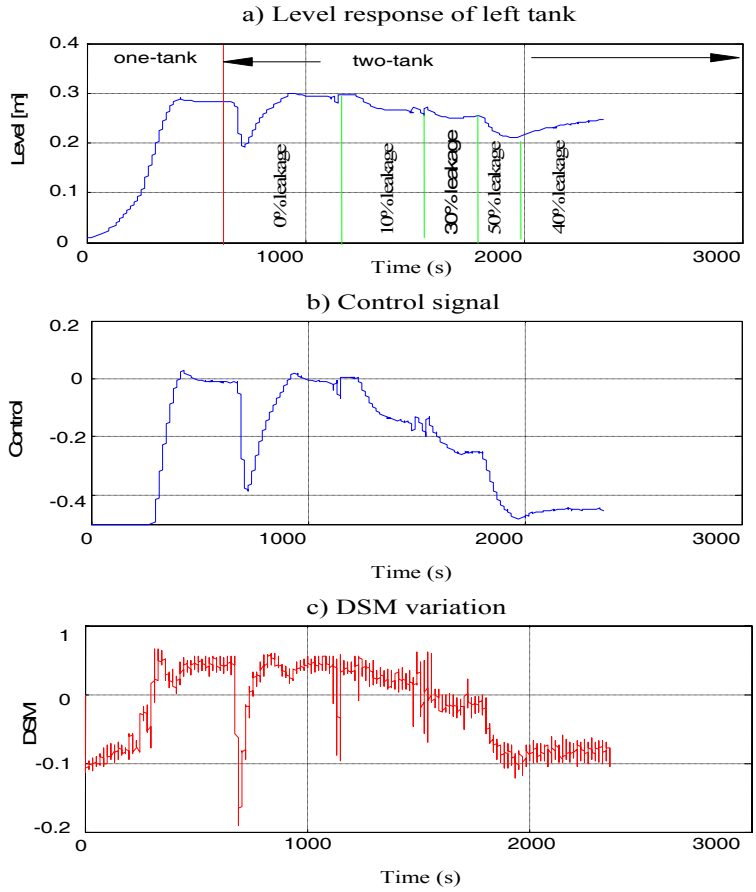


Figure 5-20: Level response and DSM using fixed PID parameters

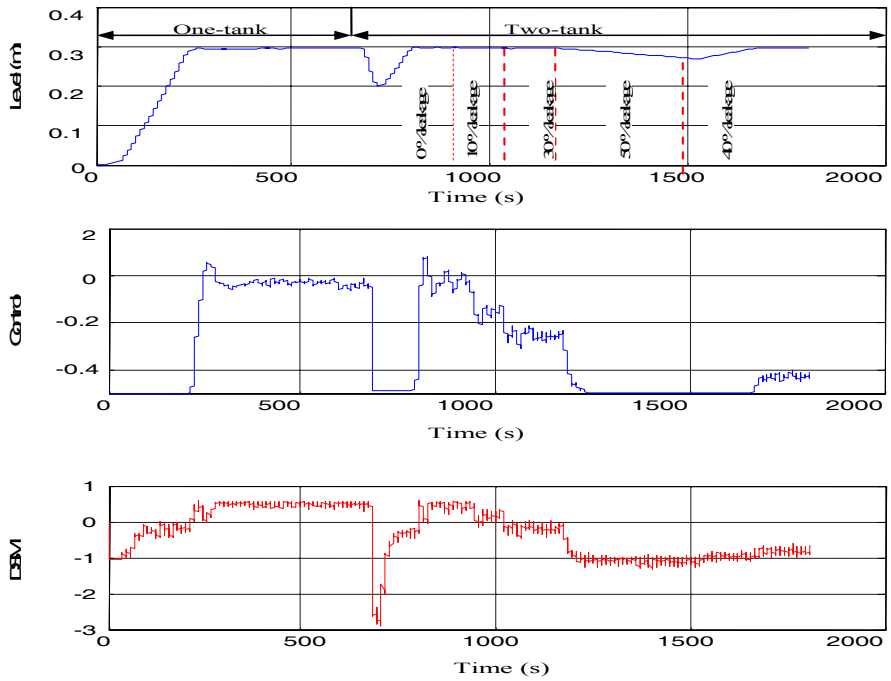


Figure 5-21: Level response and DSM using adapted PID parameters

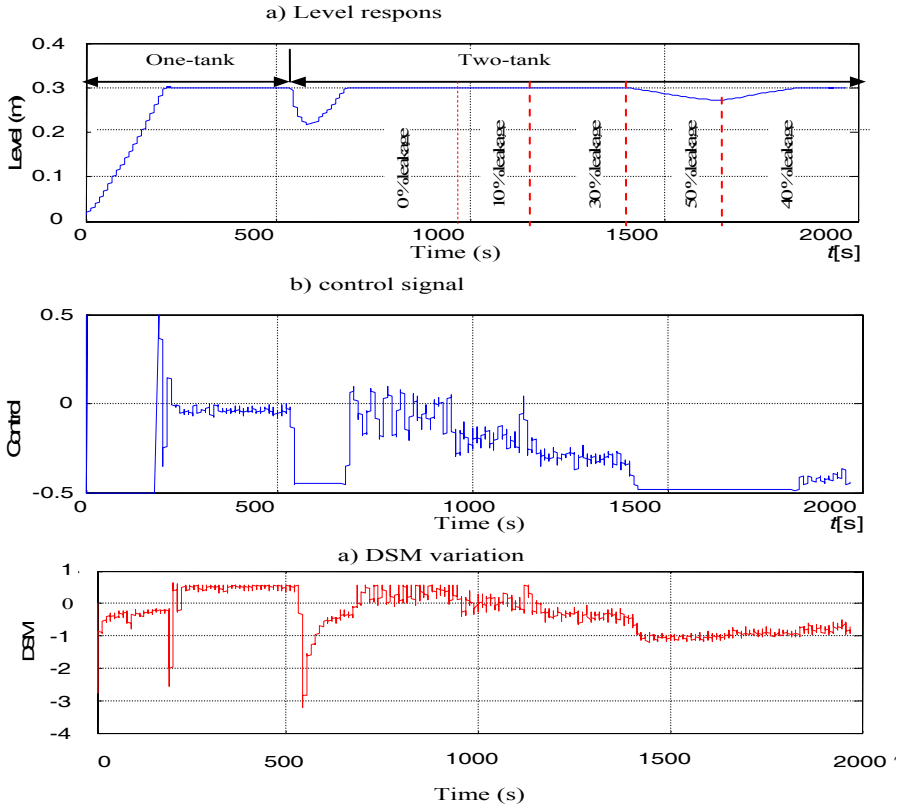


Figure 5-22: Level response and DSM using adapted Fuzzy PID parameters

5.3.2.1.2 Predictive control based on DSM

The MPC design without constraints and with DSM constraints, either soft or hard, are discussed in *Chapter 4*. The algorithms of MPC without constraints and with DSM constraints as hard constraints are tested in real-time operation. In the current experiment, the interconnecting valve is fully opened, the disturbance valve (control valve of 2nd tank) was adjusted to simulate a different load discharge disturbance and the control valve, of the first tank, is used to adjust the level in both tanks. The two-tank system is fed at constant flow 1 l/s in the first tank. The discrete linear model of the system at sampling rate equals to 10 Hz is given in Table 5-2.

Figure 5-23 shows the real-time results without considering DSM in predictive controller for the actual two-tank system when the leakage valve is opened 10% after 500 sec, 30 after 650 sec, and 50% after 800 sec, in order to regulate the level of the left tank at a set point of 0.3 m. The MPC controller parameters are $\mathbf{Q}_i$ = [30], $\mathbf{R}_i$ = [0.001]; N=10, N_1=1, and Nu=5. Figure 5-24 shows the real-time results of DSM in predictive controller as hard constraints for the same faults. It is clear from Figure 5-24 that in case of fault, the controller has the ability to operate the system within the safety limit until the fault is repaired or isolated.

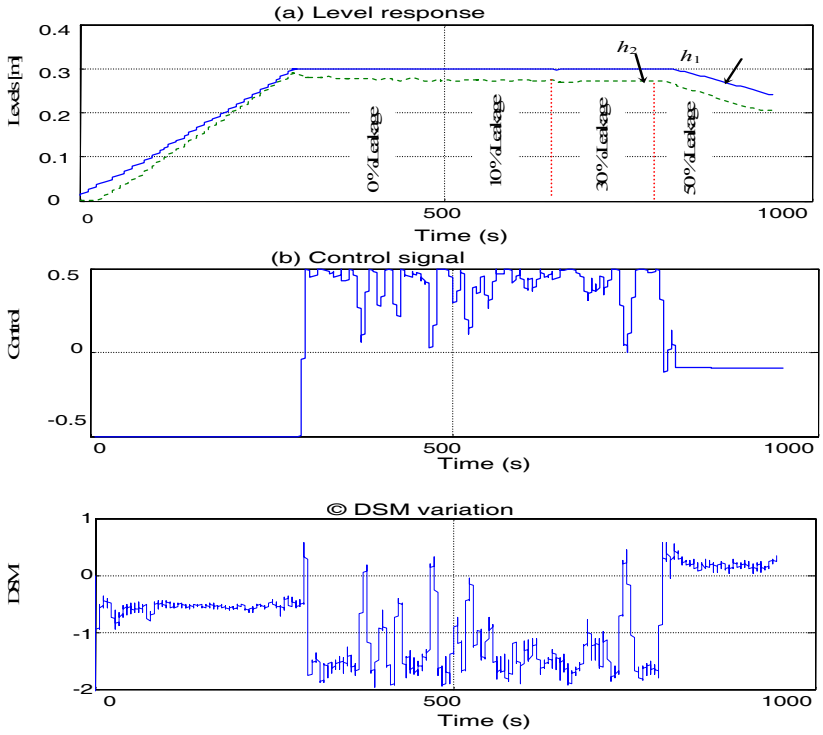


Figure 5-23: Predictive control without DSM

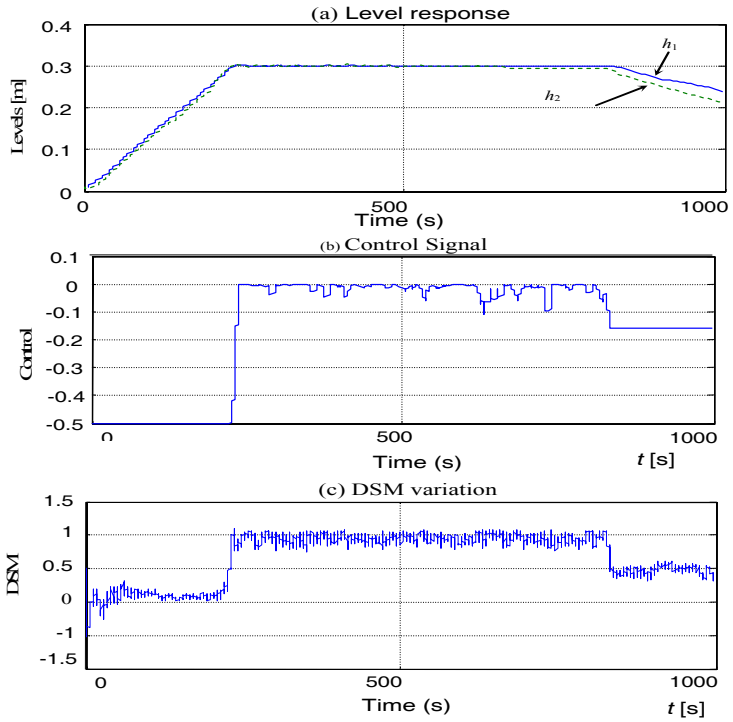


Figure 5-24: Predictive control with DSM

5.3.2.2 FTC Results

The FTC algorithm, discussed in *Chapter 4*, is implemented in real-time operation on an experimental laboratory process. A model predictive controller is used with and without DSM to regulate the level of the left tank at the set point of 0.3 m in case of actuator fault.

5.3.2.2.1 PFTC result

In this experiment, it is assumed that there is no FDI algorithm, or there is no information about the fault (*Remark 1 Chapter 4*). MPC with and without DSM constraints are used to recover the output performance and improve DSM.

Figure 5-25 shows real-time implementation of the FTC algorithm for the two-tank system in case of bias fault 30% in the control valve after 500 s until 1500 s (fault scenario). MPC without DSM is used as a nominal controller from the beginning until a fault occurs with the following parameters: $\mathbf{Q}_i$ = [50], $\mathbf{R}_j$ = [0.01], $i \in \{1, 2,...,N\}$, N=5, $j \in \{1,2,\ldots,Nu\}$, Nu=5, and N_1=1. After fault, the MPC with DSM constraint as soft constraints, which discussed in *Chapter 4 Section 4.3.2.1.1*, is used with the following

parameters $\mathbf{Q}_i$ = [30], $\mathbf{R}_i$ = [0.01], $\mathbf{P}_0$ = *diag*(10,10,10,10,10,10), N=5, N_1=1, and Nu=5. As shown in Figure 5-25, the DSM (Figure 5-25b) is negative after the fault. According to FTC algorithm another MPC with DSM constraints is used until the fault is identified. It is assumed that there is no FDI subsystem; therefore, the second controller (MPC with constraints) has been used alone to recover the performance. It is clear that the output performance (Figure 5-25a) has improved using the second controller and the DSM value as well.

Figure 5-26 shows the real-time results in case of repeated 20% actuator bias fault between "350:500s" and "700:920s". The nominal controller has been used from the start time (t=0) until the second fault (t≥700), i.e. the nominal controller has been used to recover the performance in the first fault. MPC with constraints is used to recover the second fault after DSM<0 as in Figure 5-26. It is clear that MPC with DSM constraints has improved the system performance, and the safety margin is better than nominal MPC.

5.3.2.2.2 AFTC result

In this experiment, the information obtained from FDI subsystem is used to reconfigure the controller in order to improve the output performance and DSM. Figure 5-27 shows real-time implementation of the FTC algorithm for the two-tank system in case of bias fault 20% in the control valve after 320s until 480s (fault scenario). Three controllers are used: MPC without DSM in normal operation until DSM<0 (0:370s), MPC with DSM constraints when DSM<0 until fault diagnosis (370:400s), and MPC with DSM using faulty model (actuator fault model) after fault diagnosis (after 400s). It is clear that the output and the safety performance are better than Figure 5-25 and Figure 5-26 using two controllers only; the recovery time is shorter, the steady state error is smaller and the DSM is better than the previous results for the same fault. The parameters of MPC with DSM constraints are chosen as

$\mathbf{Q}_i$ = [50], $\mathbf{R}_i$ = [0.01], $\mathbf{P}_0$ = *diag*(10,10,10,10,10,10), N=5, N_1=1, and Nu=5

in the second controller configuration, while they are

$\mathbf{Q}_i$ = [30], $\mathbf{R}_i$ = [0.01], $\mathbf{P}_0$ = *diag*(5,5,1,1,1,1), N=5, N_1=1, and Nu=5

in the third controller configuration.

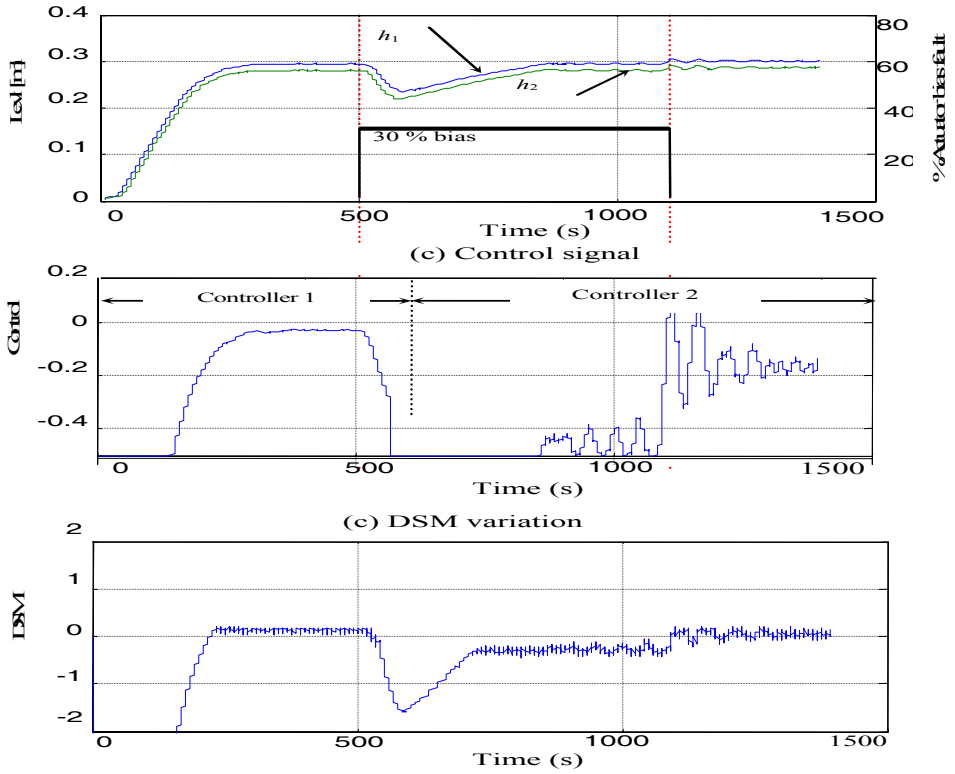


Figure 5-25: Predictive controller for Passive FTC based on DSM

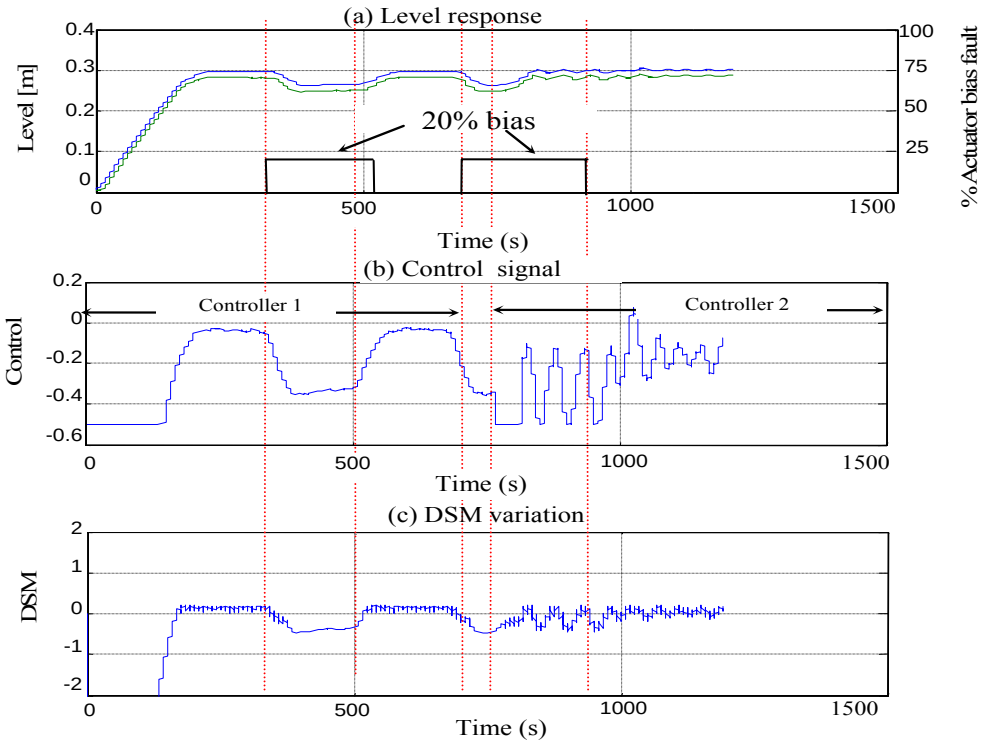


Figure 5-26: Predictive controller for PFTC with and without DSM consideration

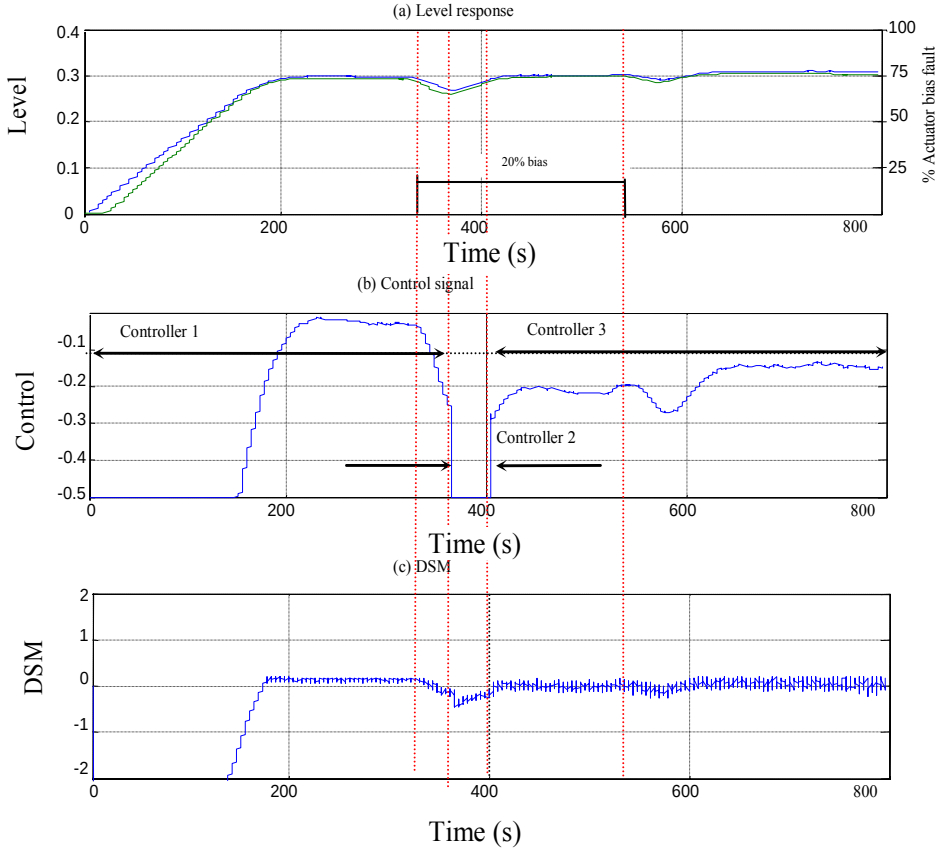


Figure 5-27: Predictive controller for AFTC system based on DSM

5.4 Conclusions

In this Chapter, an experimental laboratory process consisting of two-tank system is introduced. The hardware and software required to implement the control tasks of the process are explained. Several experiments have been tested on the process in order to show the applications of DSM. These experiments are classified as: a) a robust FDI based on DSM results; b) controller design based on DSM especially PID and MPC; c) FTC based on DSM. The theoretical background of the applications is discussed in previous chapters.

Three types of faults are tested; actuator fault, internal leakage fault and sensor fault. A discrete linearized model of the system is identified experimentally. The experimental setup model parameters are time variant due to the non-linearties of the system model and input flow variation, in addition to the existence of load disturbance. The practical FDI results demonstrate the advantage and robustness of this approach. The main advantage of the proposed approach of using DSM in FDI is the reduction in the number of diagnostic

variables. Moreover, it is not restricted to a special type of faults or models. In addition, DSM and its derivative is more sensitive to system parameter variation i.e. DSM in FDI introduces robust fault-detection schemes.

Two types of controller design based on DSM are tested, PID and MPC. Adapting the controller parameter based on DSM improves the system response, mainly the system that is exposed to non-considerable and non-measurable disturbance, whether the system model is well known or there is uncertainity in the system parameters. Adapting PID controller based on DSM, linear and fuzzy adaptation, has been implemented on an experimental hybrid plant. The main advantage of this adaptation method is that the exact model of the system is less important, and we do not need to identify the system parameter each time to reconfigure the controller. MPC without DSM and with DSM as hard constraints are implemented. Using predictive controller based on DSM gives better response than PID one, but the algorithm is complex and the computation time is considerably high. The controller design based on DSM improves safety-assessment of safety-critical systems

MPC based on DSM in the application of FTC system is implemented on a two-tank process. PFTC (one or two controller configuration) and AFTC (three controller configuration) based on DSM results demonstrate the advantage of the proposed FTC. MPC based on DSM can compensate the effect of disturbance and uncertainties of the isolated fault result.

CHAPTER 6

CONCLUSIONS AND DISCUSSION

The FDI and FTC systems are important topics in the modern control system design. During the last 3 decades, excessive work has been exerted in the field of FDI and FTC systems. A robust FDI system and reliable FTC system are necessary to increase the overall system dependability. A more dependable system is the system that has the ability to tolerate faults and prevents them from developing into failures at a subsystem or plant level.

Designing a robust FDI system and a performance recovery controller based on a new performance index called DSM are the main aspects of this work in order to design a reliable FTC system.

In *Chapter 1*, a comprehensive overview and literature survey of FDI and FTC systems have been presented. Furthermore, the main difficulties in designing FDI and FTC systems have been discussed.

The DSM definition and computation have been introduced in *Chapter 2*. Its computation methods for safe region defined by linear boundaries have been deduced too. Furthermore, its applications and limitations have been stated. Advantages of controller design based on DSM have been discussed as well. DSM index can be used as a new quality measure to compare between different controller design methods. A controller design based on DSM maintains a predefined margin of safety not only at steady state but also during transient operation. It also decreases the disturbance effect, and help speeding up performance recovery in case of some system faults.

The uncertainties in the system model parameters and the disturbances are the main difficulties in designing a FDI system, which affect the behaviour of FDI system. Therefore, the FDI system has to be robust to such modelling error and disturbance. The robustness of FDI system has been discussed in *Chapter 3*, and the existing techniques to design a robust FDI system and their limitations have been discussed. Since each robust FDI scheme has limitations and is applied in a special application, a robust FDI problem has not been fully solved. Thus, design a FDI system based on DSM is introduced in *Chapter 3*. The main advantage of the proposed approach of using DSM in FDI is the reduction of the number of diagnosis variables. In addition, DSM and its

derivative are more sensitive to system parameter variation than the measured signal output or residual, i.e. using DSM in FDI introduces robust fault-detection schemes. The proposed FDI scheme does not restrict to a special type of faults, but it can be applied for different type of faults whether additive or multiplicative.

The controller design based on DSM improves safety-assessment of safety-critical systems. Since PID controller is one of most popular and robust controller in particular for SISO systems, and MPC is an effective controller for the MIMO systems due to its ability to deal with hard constraints, the MPC design and the PID controlled parameters adaptation based on DSM are introduced in *Chapter 4*. The MPC using DSM has been discussed in the application of FTC system as well. Adapting the weights of MPC objective function based on DSM index has been highlighted too, in order to find a feasible solution and satisfy the safety requirements for a predefined performance. Finally, a general frame work of FTC system design based on DSM has been introduced and discussed. The proposed FTC scheme employs the introduced FDI scheme in addition to the controllers design based on DSM, in particular MPC with DSM. In some faulty situation, recovering the system performance to the nominal one can not be achieved. As a result, reducing the output performance is necessary in order to increase the system availability. Thus, the selection of degraded reference model and command input have been discussed and included in the proposed FTC scheme. The combination of controller design and FDI based on DSM with accepted degraded performance generates a reliable FTC system, which enhance the overall system dependability.

DSM applications in FDI, controller adaptation and design, and FTC system, which have been introduced in *Chapter 3* and *Chapter 4*, have been implemented in real-time on an experimental laboratory process in *Chapter 5*. Different fault simulations have been tested in real time; actuator fault, internal leakage fault and sensor fault. The practical FDI results demonstrate the advantage and robustness of this approach. Two types of controller design based on DSM has been tested, PID and MPC, as well. The MPC based on DSM in the application of FTC system either PFTC or AFTC design have been implemented. The results of the real-time implementation demonstrate the advantages and show the applicability of the proposed shames.

The key issue of the DSM application is the determination of the safe region. The better specified safe region is, the more powerful benefits can be obtained, such as robust FDI system, robust controller and dependable FTC. The choice of the state variables relevant to the safety is not unique because it depends on the operation

experience of the process (knowledge based). The safe operation region can be considered as an invariant set and can be constructed by the same procedures if the disturbance and uncertainties belong to compact sets.

However, in some processes, it is quite difficult to determine the safe operation region. Moreover, the mathematical formulation of the DSM is not easy to obtain for some safety regions such as non-convex region and/or nonlinear boundaries; a knowledge-based model (fuzzy, neural, etc.) can be used in this case. Thus, more investigation about safety region construction and DSM computation will be the focus of future work.

The applicability and DSM computation for large-scale system, and using DSM in fault prognosis are important topics, which will be covered as well.

APPENDIX A

DSM Computation

A.1 Vector Algebra Method

Let the number of state variables of interest are all state variable (*m=n)* to generalize the algorithm. Consider the safe region is defined by *q* linear inequalities in the form

$$\phi_i(\mathbf{x})=\mathbf{a}_i^T.\,\mathbf{x} - c_i \le 0;\ \ i=1,2,\ldots,q \tag{A.1}$$

Then the boundary equation can be written in the form

$$\phi_i(\mathbf{x}_i)= \mathbf{a}_i^T\,.\,\mathbf{x}_i - c_i = 0 \tag{A.2}$$

where $\mathbf{a}_i^T \in \Re^n$ is constant vector, and, $\mathbf{x}_i \in \{\ \mathbf{x}\ |\phi_i(\mathbf{x})=0\}$, and $(\mathbf{x}-\mathbf{x}_i)$ is the distance vector between $\mathbf{x}$ and $\mathbf{x}_i$.

Consider three dimension state vector (n=3), and let $\mathbf{x}_{i1}$, $\mathbf{x}_{i2}$ two vector on the boundary ϕ_i (FigureA1) then

$$\phi_i(\mathbf{x}_{i1})- \phi_i(\mathbf{x}_{i2})=\mathbf{a}_i^T(\mathbf{x}_{i1}- \mathbf{x}_{i2})=0$$

This indicates that $\mathbf{a}_i$ is the orthogonal vector on the boundary ($\mathbf{a}_i^T \perp \phi_i(\mathbf{x})$)

$\bar{\mathbf{p}} = \dfrac{\mathbf{a}_i}{\|\mathbf{a}_i\|}$ is the direction of orthogonal vector on the boundary as shown in Figure A1

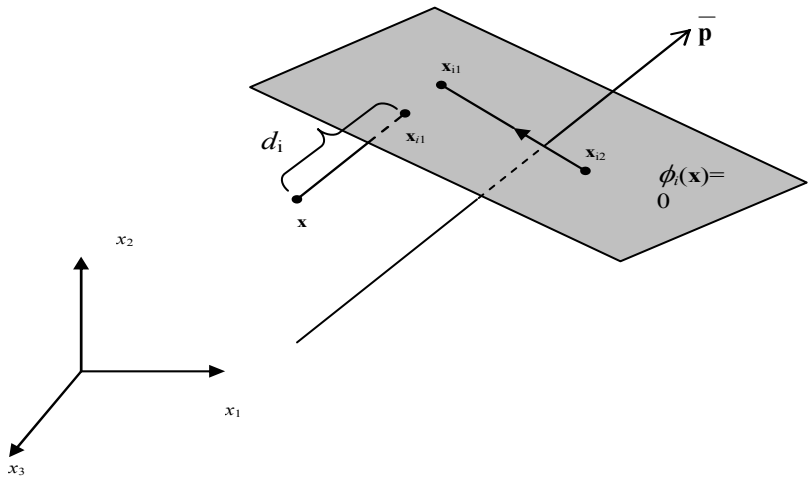


Figure A1: boundary surface

The minimum distance between any state vector (**x**) in state space and the boundary must be the norm of a vector in the direction of $\pm\bar{\mathbf{p}}$, start from **x** and terminate at state vector on the boundary **($\mathbf{x}_i$)**.

$$d_i = \left\| \mathbf{x}_i - \mathbf{x} \right\|_2$$

$$\bar{\mathbf{p}} = \frac{\mathbf{a}_i}{\left\| \mathbf{a}_i \right\|_2} = \begin{cases} + \dfrac{(\mathbf{x}_i - \mathbf{x})}{\left\| \mathbf{x}_i - \mathbf{x} \right\|_2} \text{ if } \mathbf{x} \text{ satisfy the constrain } \phi(\mathbf{x}) \le 0 \\ - \dfrac{(\mathbf{x}_i - \mathbf{x})}{\left\| \mathbf{x}_i - \mathbf{x} \right\|_2} \text{ if } \mathbf{x} \text{ does not satisfy the constrain } \phi(\mathbf{x}) \le 0 \end{cases} \quad \text{(A.3)}$$

Multiply both side of (A.3) by $\mathbf{a}_i^T$ and replace $\left\| \mathbf{x}_i - \mathbf{x} \right\|_2$ by d_i then

$$d_i = \frac{\left| \mathbf{a}_i^{\mathrm{T}} . \mathbf{x}_i - \mathbf{a}_i^{\mathrm{T}} . \mathbf{x} \right|}{\left\| \mathbf{a}_i \right\|_2}$$

Substitute from (A.2)

$$d_i = \frac{c_i - \mathbf{a}_i^{\mathrm{T}} . \mathbf{x}}{\left\| \mathbf{a}_i \right\|_2} \begin{cases} \ge 0 \text{ iff } \mathbf{x} \text{ satisfy the constrain } \phi(\mathrm{x}) \le 0 \\ \le 0 \text{ iff } \mathbf{x} \text{ satisfy does not the constrain } \phi(\mathrm{x}) \le 0 \end{cases} \quad \text{(A.4)}$$

The absolute value of the result of equation (A.4) gives the minimum distance between the boundary and any state vector in state space and the sign indicates the satisfaction of the constrain.

In general if **x** (t) is the system state vector at any time t then $d_i(t) = \dfrac{c_i - \mathbf{a}_i . \mathbf{x}(t)}{\left\| \mathbf{a}_i \right\|_2}$

A.2 Optimization Method

$$\min_{\mathbf{x}_i} \left\| (\mathbf{x} - \mathbf{x}_i) \right\|_2^2 \quad \text{(A.5)}$$

subject to

$$\phi_i(\mathbf{x}) = \mathbf{a}_i^T \mathbf{x}_i - \mathrm{c}_i = 0 \quad \text{(A.6)}$$

using Lagrange principle the objective function will be

$$\min_{\mathbf{x}_i, \lambda} \left(\left\| (\mathbf{x} - \mathbf{x}_i) \right\|_2^2 + \lambda \left(a_i^T . x_i - c_i \right) \right) \quad \text{(A.7)}$$

Taking the derivative with respect to $\mathbf{x}_i$ and λ, it leads to the equations

$$\begin{bmatrix} \frac{\partial J}{\partial \mathbf{x}} \\ \frac{\partial J}{\partial \lambda} \end{bmatrix} = \left[\begin{array}{cccc|c} 2 & 0 & \cdots & 0 & a_{i1} \\ 0 & 2 & & \vdots & a_{i2} \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 2 & a_{in} \\ \hline a_{i1} & \cdots & & a_{in} & 0 \end{array}\right] \begin{bmatrix} (x_1 - x_{i1}) \\ (x_2 - x_{i2}) \\ \vdots \\ (x_n - x_{in}) \\ \hline \lambda \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ (c_i - \mathbf{a}_i \mathbf{x}) \end{bmatrix} = 0 \qquad \text{(A.8)}$$

$$\text{Hence, } \mathbf{x} - \mathbf{x}_i = \mathbf{M}^{-1}\mathbf{c}_x = \frac{1}{\|\mathbf{a}_i\|_2^2} \mathbf{a}_i^T (c_i - \mathbf{a}_i \mathbf{x}) \qquad \text{(A.9)}$$

$$\text{where } \mathbf{M}^{-1} = \left[\begin{array}{cccc|c} 2 & 0 & \cdots & 0 & a_{i1} \\ 0 & 2 & & \vdots & a_{i2} \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 2 & a_{in} \\ \hline a_{i1} & \cdots & & a_{in} & 0 \end{array}\right]^{-1}, \mathbf{c}_x = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ (c_i - \mathbf{a}_i \mathbf{x}) \end{bmatrix}, \ \mathbf{x} - \mathbf{x}_i = \begin{bmatrix} (x_1 - x_{i1}) \\ (x_2 - x_{i2}) \\ \vdots \\ (x_n - x_{in}) \end{bmatrix}, \text{and } \mathrm{x}_i \text{ is}$$

the solution of optimization problem.

It is sufficient to compute the last raw of $\mathbf{M}^{-1}$, where all component of $\mathbf{c}_x$ are zero instead of the last element. Therefore,

$$\mathbf{x} - \mathbf{x}_i = \frac{1}{2^{n-1}\|\mathbf{a}_i\|_2^2} \left[XX \quad \begin{vmatrix} 2^{n-1} a_{i1} \\ 2^{n-1} a_{i2} \\ \vdots \\ 2^{n-1} a_{in} \end{vmatrix} \right] \mathbf{c}_x = \frac{1}{\|\mathbf{a}_i\|_2^2} \mathbf{a}_i^{\mathrm{T}} (c_i - \mathbf{a}_i \mathbf{x}) \qquad \text{(A.10)}$$

Distance vector $d_i = \|(\mathbf{x} - \mathbf{x}_{io})\|_2$ then $d_i(t) = \dfrac{c_i - \mathbf{a}_i^{\mathrm{T}} . \mathbf{x}(t)}{\|\mathbf{a}_i\|_2}$

APPENDIX B

Analogical Exclusion-XOR Gate

The Boolean gates are used when the inputs are limited within two values, 0 and 1. If all the four Quadrants of the input space are considered i.e. each input variable $\in$[-max, +max], then the Boolean gates are no longer suitable. Quantization of the input space could be used to solve this difficulty. On the other side, the number of input space is increasing proportionally with the resolution required. The analogical gates are a generalisation of the Boolean gates when the four Quadrant of the input space are considered [151].

Analogical XOR gate, which is used in this thesis to behave as the anti-windup circuit for PID controller, is explained here.

The functionality description of analogical XOR:

Refereeing to Figure B.1, The output is identically zero if both inputs equal in magnitude. If one input is zero, the output is equal to the present non-vanishing input.

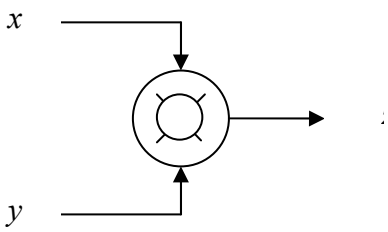


Figure B1: Analogical XOR

$$z = x \,¤\, y = sig(x+y)\left\|x\right|\zeta\left(-\left|y\right|,\left|x\right|\right) - \left|y\right|\zeta\left(\left|x\right|,-\left|y\right|\right)\Big|$$

$$\zeta(x,y) = e^{-\left(\frac{ax^2+bxy}{x^2+y^2}\right)}$$

where a=1.02889, b=0.3574 and $x,y \in \Re$

Basic Characteristics:

$x \text{¤} y = y \text{¤} x$

$c(x \text{¤} y) = cx + cy,\ c \in \Re$

$x \text{¤} x = 0$

$x \text{¤} 0 = 0$

$x \text{¤} -x = 0$

$min(x,y) \le x \text{¤} x \le max(x,y)$

$\partial (x \text{¤} y) / \partial x \big|_{x=0} = 0$

$\partial (x \text{¤} y) / \partial y \big|_{y=0} = 0$

Anti-rest wind-up Network for PID controller

The Strategy for anti-rest wind-up is as follows:

a) In linear control range, neither the magnitude nor the sign of the integeral-gain (K_I) is changed.
b) When commend-saturation occurs, the magnitude of the K_I gain is reduced first.
c) As the difference between the saturated (u) and the unsaturated command (u_o) further increases, the sign of K_I is made negative together with further decrease of the magnitude.

This strategy can be implemented using a single XOR analogical-gate,

$$K_I = K_{Io}[((u-u_o)/u_o) \text{¤} (u/u_o)]$$

where the first input $x=((u-u_o)/u_o)$ and the second one $y=u/u_o$

If the controller under normal operation (unsaturated) then $x=0$ otherwise x is negative. Negative x means that the command input is saturated; y is always positive. Therefore, the value of integral action ,K_I, is controlled based on x.

APPENDIX C

MPC Solution Using One Shot Optimization

1. *MPC without DSM Constraints*

The objective function of MPC in (4.10)

$$J = \sum_{i=N_1}^{N} \left\| \hat{\mathbf{e}}(i+k|k) \right\|_{\mathbf{Q}_i}^2 + \sum_{i=0}^{N_u-1} \left\| \mathbf{u}(i+k|k) \right\|_{\mathbf{R}_i}^2 \quad \text{C.1}$$

Considering the state space model of the system, this equation can be written in the form of

$$J = \underset{\rightarrow}{\mathbf{u}}^T \mathbf{M} \underset{\rightarrow}{\mathbf{u}} + 2\mathbf{H}\underset{\rightarrow}{\mathbf{u}} + c_r \quad \text{C.2}$$

The problem here is that minimize equation C. 2 with respect to the control sequence $\underline{\mathbf{u}}$ with out considering the state and control constraints.

where

$$\begin{aligned} \mathbf{M} &= \mathbf{C}_B{}^T \mathbf{Q}_t \mathbf{C}_B + \mathbf{R}_t; \\ \mathbf{H} &= (\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a \mathbf{x}(k))^T \mathbf{Q}_t \mathbf{C}_B; \\ \mathbf{c}_r &= (\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a \mathbf{x}(k))^T \mathbf{Q}_t (\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a \mathbf{x}(k)); \end{aligned} \quad \text{C.3}$$

Note that all the matrices, used here, are defined in *Chapter 4 Section 4.3.2*

The control sequence is deduced using one shot optimization as fallow:

According to the optimality principle, $\frac{dJ}{d\underline{\mathbf{u}}} = 0$ at the optimal control sequence ($\underline{\mathbf{u}}^*$)

then

$$2\mathbf{M}\underset{\rightarrow}{\mathbf{u}} + 2\mathbf{H}^{\mathrm{T}} = 0 \quad \text{C.4}$$

i.e.

$$\underset{\rightarrow}{\mathbf{u}}^* = -M^{-1}\mathbf{H}^{\mathrm{T}} \quad \text{C.5}$$

Substituting M and H from C.3 then

$$\underset{\rightarrow}{\mathbf{u}} = -\left[\mathbf{C}_B{}^T \mathbf{Q}_t \mathbf{C}_B + \mathbf{R}_t\right]^{-1} \left[(\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a \mathbf{x}(k))^T \mathbf{Q}_t \mathbf{C}_B\right]^{\mathrm{T}}$$

$$\underset{\rightarrow}{\mathbf{u}} = -\left[\mathbf{C}_B{}^T \mathbf{Q}_t \mathbf{C}_B + \mathbf{R}_t\right]^{-1} \mathbf{C}^{\mathrm{T}}{}_B \mathbf{Q}_t (\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a \mathbf{x}(k)) \quad \text{C.6}$$

And the current input vector is

$$\mathbf{u}(k) = \left[\mathbf{I}_r : 0 : \cdots 0\right]\left[\mathbf{K}_y \underset{\rightarrow}{\mathbf{y}} - \mathbf{K}_x \mathbf{x}(k)\right] \quad \text{C.7}$$

$$\mathbf{K}_y = \left[\mathbf{R}_t + \mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{C}_B\right]^{-1}\mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t$$

$$\mathbf{K}_x = \left[\mathbf{R}_t + \mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{C}_B\right]^{-1}\left[\mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{C}_a\right]^*$$

2. MPC with DSM constraint as soft constraint

The objective function of MPC with softening DSM constraint (4.22)

$$J = \left(\begin{bmatrix}\underset{\rightarrow}{\mathbf{e}} \\ \underset{\rightarrow}{\mathbf{d}}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_t & 0 \\ 0 & \mathbf{P}_t\end{bmatrix}\begin{bmatrix}\underset{\rightarrow}{\mathbf{e}} \\ \underset{\rightarrow}{\mathbf{d}}\end{bmatrix} + \underset{\rightarrow}{\mathbf{u}}\mathbf{R}_t\underset{\rightarrow}{\mathbf{u}}\right) \quad \text{C.8}$$

can also be written in the form

$$J = \underset{\rightarrow}{\mathbf{u}}^T\mathbf{M}^*\underset{\rightarrow}{\mathbf{u}} + 2\mathbf{H}^*\underset{\rightarrow}{\mathbf{u}} + c^*_r \quad \text{C. 9}$$

where

$$M^* = \left[\mathbf{R}_t + \mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{C}_B + \mathbf{D}_b{}^{\mathrm{T}}\mathbf{P}_t\mathbf{D}_b\right] \quad \text{C. 10}$$

$$\begin{aligned}\mathbf{H}^* &= (\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a\mathbf{x}(k))^T\mathbf{Q}_t\mathbf{C}_B + (d_t - D_a\mathbf{x}(k))^T\mathbf{P}_tD_b; \\ \mathbf{c}^*r &= (\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a\mathbf{x}(k))^T\mathbf{Q}_t(\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a\mathbf{x}(k)) + (d_t - D_a\mathbf{x}(k))^T P_tD_b;\end{aligned} \quad \text{C. 11}$$

And the optimal sequence is obtained as C.5

$$\underset{\rightarrow}{\mathbf{u}}^* = -M^{*-1}\mathbf{H}^{*\mathrm{T}}$$

Substituting from C. 10 and C. 11 into C. 9, then

$$\underset{\rightarrow}{\mathbf{u}}^* = \left[\mathbf{K}_y\underset{\rightarrow}{\mathbf{y}} + \mathbf{K}_d\mathbf{d}_t - \mathbf{K}_x\mathbf{x}(k)\right]$$

Where

$$\mathbf{K}_y = \left[\mathbf{R}_t + \mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{C}_b + \mathbf{D}_b{}^{\mathrm{T}}\mathbf{P}_t\mathbf{D}_b\right]^{-1}\mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t$$

$$\mathbf{K}_d = \left[\mathbf{R}_t + \mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{C}_B + \mathbf{D}_b{}^{\mathrm{T}}\mathbf{P}_t\mathbf{D}_b\right]^{-1}\mathbf{D}_b{}^{\mathrm{T}}\mathbf{P}_t$$

$$\mathbf{K}_x = \left[\mathbf{R}_t + \mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{C}_B + \mathbf{D}_b{}^{\mathrm{T}}\mathbf{P}_t\mathbf{D}_b\right]^{-1}\left[\mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{C}_a + \mathbf{D}_b{}^{\mathrm{T}}\mathbf{P}_t\mathbf{D}_a\right]$$

APPENDIX D

Magnetic-Tape-Drive system

The Magnetic-tape-Drive system is a MIMO system shown in Fig.4.6. There is an independently controllable drive motor on each end of the tape; therefore, it is possible to control the tape position over the read head, x_3, as well as the tension in the tape. The tape is modeled to be a linear spring with small amount of viscous damping. The goal of the control system is to enable commanding the tape to specific position over the read head while maintaining a specified tension in the tape at all times. The desired specifications are that the tape position must be adjusted if the tape head is moved 1mm with 1% settling time of 2.50 sec and overshoot less than 20%. The tape tension, T_e, should be controlled to 2 N with constraint that $0 < T_e < 4$. The current is limited to 1A at each drive motor.

The equation of motion of the system [203] is

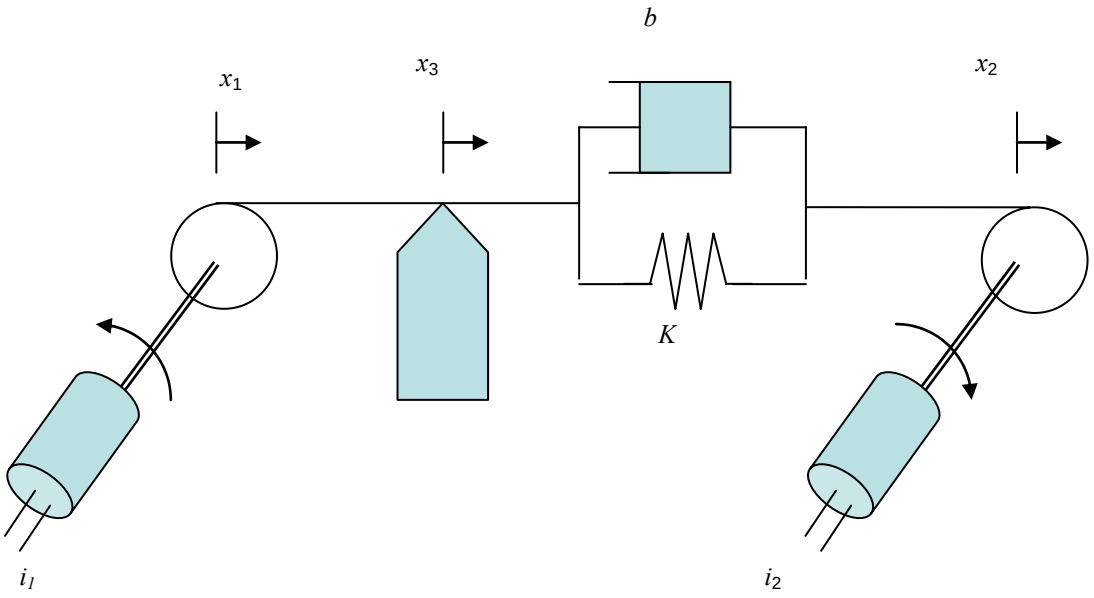


Figure C1: Schematic diagram of magnetic tape drive

$$
\begin{aligned}
J\ddot{\theta}_1 &= -T_e r + K_m i_1, \\
J\ddot{\theta}_2 &= -T_e r + K_m i_2, \\
T_e &= k(x_2 - x_1) + b(\dot{x}_1 - \dot{x}_2), \\
x_3 &= (x_2 + x_1)/2,
\end{aligned}
\tag{D.1}
$$

where i_1 and i_2 are the current into drive motors 1, 2, respectively, T_e tension in tape (N), θ_1 , θ_2 angular position of motor, r assembly, $x_1$and x_2 position of tape over read head (mm),

J=0.006375 kg.m2, motor and capstan inertia,

r =0.1 m, radias,

K_m=0.544 N.m/A, motor torque constant,

k=2.113 N/m, tape spring constant

b=0.375 N sec/m, tape damping constant.

Equation (4.16) is the system state space model where the state vector $\mathbf{x}=[x_1\ x_2\ \omega_1\ \omega_2]^T$, input vector $\mathbf{u}=[i_1\ i_2]^T$ and the output vector $\mathbf{y}=[x_3\ T_e]^T$ is

REFERENCES

[1] I. Sommerville, Software *Engineering,* 6th ed. Addison-Wesley, 2000.

[2] M. Blanke, M. Kinnaert, I. Lunze, and M. Staroswiecki, *Diagnosis and fault-Tolerant Control*. Springer, 2003.

[3] R. Izadi-Zammanabadi, "Practical Approach to Reliability, Safety, and Active Fault-tolerance," Lecture Notes, Dept. Control Eng., Aalborg University, Denmark, 2000

[4] N. G. Leveson, *Software-system safety and computers.* Addison Wesley, 1995.

[5] Winfrid G. Schneeweiss, *Fault Tree Methods* (*in the Field of Reliability and Safety Technology*). Hagen, Germany :LiLoLe-Verlag GmbH, 1999.

[6] D.H. Stamatis, *Failure Mode Effect Analysis: from Theory to Execution,* 2nd ed. ASQ Quality press, June 2003.

[7] J. Patton, P. Frank, and R. Clark, *Issues of Fault Diagnosis for Dynamic System.* Springer, 2000.

[8] V. Venkatasubramanian, R Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis-Part1: Quantitative model based methods", *Computers and Chemical Engineering,* vol. 27, pp. 293-311, 2003.

[9] Sourabh Dash and Venkat Venkatasubramanian, "Challenges in the industrial application of fault diagnostic system," *Computer and Chemical Engineering*, vol. 24, pp. 785-791, 2000.

[10] D. Füssel, P. Balle, and R. Isermann, "Closed loop fault diagnosis based on a nonlinear process model and automatic fuzzy rule generation," in *Proc. IFAC SAFEPROCESS'97 symposium*, Hull, UK, 1997, pp. 359-364.

[11] P. Balle and R. Isermann, "Fault detection and isolation for nonlinear processes based on local linear fuzzy models and parameter estimation," in *Proc. American Control Conference*, Philadelphia, PA, 1998, pp. 1605-1609.

[12] F. Filippetti, G. Franceschini, C. Tassoni, and P. Vas, "Recent developments of induction motor drives fault diagnosis using AI techniques," *IEEE Trans. Ind. Electron*., vol. 47, no. 5, pp. 994-1004, Oct. 2000.

[13] S. Simani, C. Fantuzzi, and R. J. Patton, *Model-based Fault Diagnosis in Dynamic Systems using Identification Techniques*. London Limited: Springer-Verlag , 2003

[14] M. Borairi and H. Wang, “Actuator and sensor fault diagnosis of nonlinear dynamic systems via genetic neural networks and adaptive parameters estimation technique,” in *Proc. IEEE Int. Conf. Control Application*, Trieste, Italy, Sept. 1998, pp. 278-282.

[15] S. M. El-Shal and A. S. Morris, “A fuzzy expert system for fault detection in statistical process control pf industrial processes,” *IEEE Trans. System Man Cybern.,* vol. 30, pp. 281-289, May 2000.

[16] J. Chen, C. J. Lopez-Toribio, and R. J. Patton, “Non-linear dynamic systems fault detection and isolation using fuzzy observers,” *Proceeding of Institution of Mechanical Eng. Part I: Journal of Systems and Control Eng.*, vol. 213, no. 6, pp. 467-476, 1999.

[17] K. S. Lee and J. Vagners, “Reliable decision unit utilizing fuzzy logic for observer based fault detection systems,” in *Proc. IFAC SAFEPROCESS’97 symposium*, Hull, UK, 1997, pp. 693-698.

[18] G. C. L. G. Betta and A. Pietrosanto, “An advanced neural-network based instrument fault detection scheme,” *IEEE Trans. Instrumentation and Measurement,* vol. 47, no. 2, pp. 507-513, 1998.

[19] H. N. Kovio, “Artificial neural networks in fault diagnosis and control,” *Control Eng. Practice*, vol. 2, no. 7, pp. 89-101, 1994.

[20] Y. Maki and K. A. Loparo, “A neural-network approach to fault detection and diagnosis in industrial processes”, *IEEE Trans. Control System Tech*., vol. 5, no. 6, pp. 529-541, Nov. 1997.

[21] M. J. de la Fuente and P. Vega, “A neural network based approach for fault detection and diagnosis: application to a real process,” in *Proc. IEEE Int. Conf. Control Applications*, Trieste, Italy, Sep. 1998, pp. 188-193.

[22] H. Guo, J. A. Crossman, Y. L. Murphey, and M. Coleman, “Automotive signal diagnostics using Wavelets and machine learning,” *IEEE Trans*. *Vehicular Technology*, vol. 49, no. 5, pp. 1650-1662, Sept. 2000.

[23] S. Yoon, J.n Landry, N. Kettaneh, W. Pepe, and S. Wold, “Multivariate process monitoring and early fault detection (MSPC) using PCA and PLS,” in *Proc. Plant Automation and Decision Support Conf.,* Hyatt Regency, San Antonio, Rexas, Sept. 21-24, 2003.

[24] T. Kourti, “Process analysis and abnormal situation detection: From theory to practice,” *IEEE control system Magazine*, vol. 22, no. 5, pp. 10-25, October 2002.

[25] S. Zhao and Z. Xu, “Design of a novel knowledge-based fault detection and isolation scheme,” *IEEE Tran. System man and cybernetics*, vol. 34, no. 2, pp. 1089-1095, April 2004.

[26] D. Mylaraswamy and V. Venkatasubramanian, “A hybrid framework for large scale process fault diagnosis,” *Computer and Chem. Eng*., vol. 21, pp. 935–940, 1997.

[27] G. Biswas, R. Kapadia, and X. W. Yu, “Combined qualitative-quantitative steady-state diagnosis of continuous-valued systems,” *IEEE Trans. System, Man, Cybern. A,* vol. 27, pp. 167–185, Mar. 1997.

[28] L. Magni, R. Scattolini, and C. Rossi, “A fault detection and isolation method for complex industrial systems,” *IEEE Trans. System, Man, Cybern*. A, vol. 30, pp. 860–865, Nov. 2000.

[29] L. Magni, R. Scattolini, and C. Rossi, “A design methodology for diagnostic strategies for industrial systems,” *Int. J. Systems Science,* vol. 33, pp. 505–512, 2002.

[30] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, “Failure diagnosis using discrete event models,” *IEEE Trans. Control Systems Technology*, vol. 4, no. 2, pp. 105–124, Mar. 1996.

[31] A. Barigozzi, L. Magni, and R. Scattolini, “A Probabilistic Approach to Fault Diagnosis of industrial systems,” *IEEE Trans. Control Systems Technology*, vol. 12, no. 6, pp. 950-955, 2004.

[32] S. Narasimhan, F. Zhao, G. Biswas, and E. Hung, “Fault isolation in hybrid system combining model based diagnosis and signal processing,” in *Proc. IFAC Symposium SAFEPROCESS’00*, Budapest, Hungary, June 2000, pp. 14-16.

[33] S. Persin, B. Tovornik, N. Muskinjia, and D. Valh, “Increasing process safety using analytical redundancy,” *Electrotehniski Vestnik,* vol. 69, no. 3-4, pp. 240-246, 2002.

[34] M. Nyberg, “Model based fault diagnosis methods theory and automotive engine applications,” Ph.D. dissertation, Linköping Uni., Sweden, 1999.

[35] R. Isermann, “Model-based fault detection and diagnosis - status and applications,” *16th symposium on Automatic Control in Aerospace*, Petersburg, Russland, 14-18 June, 2004.

[36] I. J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. NY: Marcel Dekker Inc., 1998.

[37] R. J. Patton and J. Chen, “Robust fault detection using eigenstructure assignment: tutorial consideration and some new results,” in *Proc. 30th IEEE Conf. Decision and Control*, Brighton, England, 1991, pp. 2242-2247.

[38] R. J. Patton and S. M. Kangethe, *Robust Fault Detection Using Eigenstructure Assignment of Observers in Fault Diagnosis in Dynamic Systems- Theory and Application.* New York: Prentice Hall International, 1989.

[39] L. C. Shen, S. K. Chang, and P.L. Hsu, “Robust fault detection and isolation with unstructured uncertainty using eigenstructure assignment,” *AIAA Journal of Guidance*, Control, and Dynamics, vol. 21, no. 1, pp. 50-55, 1998.

[40] M. Hou and P. C. Muller, “Fault detection and isolation observers,” *Int. Journal of Control,* vol. 60, no. 5, pp. 827-846, 1994.

[41] J. Chen, J. R. Patton, and Y. H. Zhang, “Design of unknown input observer and robust fault detection filters,” *Int. Journal of Control,* vol. 63, no. 1, pp. 85-105, 1996.

[42] P. M. Frank, “Enhancement of robustness in observer based fault detection,” *Int. Journal of Control*, vol. 59, no. 4, pp. 955-981, 1994.

[43] R. K. Douglas and J. L. Speyer, “Robust fault detection filters design”, *AIAA Journal of Guidance, Control and Dynamics,* vol. 19, no.1, pp. 214-218, 1996.

[44] P. M. Frank and X. Ding, “Frequency domain approach to optimality robust residual generation,” *Automatica,* vol. 30, no. 5, pp. 789-804, 1994.

[45] A. Guasch, J. Quevedo, and R. Milne, “Fault diagnosis for gas turbines based on the control systems,” *Engineering Application of Artificial Inelegance*, vol. 13, no. 4, pp. 477-483, 2000.

[46] J. Y. Keller and M. Darouach, “A new estimation for dynamic stochastic systems with unknown inputs: application to robust fault diagnosis”, in *Proc. IFAC Symposium* SAFEPROCESS`97, Hull, UK, 1997, pp. 177-180.

[47] N. E. Wu, Y. M. Zhang, and K. Zhou, “Detection, estimation, and accommodation of loss of control effectiveness,” *International Journal of Adaptive Control and Signal processing*, vol. 14, no. 7, pp. 775-795, 2000.

[48] Y. M. Zhang and X. R. Li, "Detection and diagnosis of sensor and actuator failures using IMM estimator," *IEEE Trans. Aerospace and Electronic Systems,* vol. 34, no. 4, pp. 1293-1313. 1998.

[49] D. Rupp, G. Ducard, E. Shafai, and H. P. Geering, "Extended multiple model adaptive estimation for the detection of sensor and actuator faults," *in Proc.* 44th *IEEE Conf. Decision and Control, and European Control Conf.* 2005, Seville, Spain, December 2005, pp. 3079-3084.

[50] G. Ducard and H. P. Geering, "A reconfigurable flight control system based on the EMMAE method," in *Proc. American Control Conference*, Minneapolis, MN, USA, June 2006, pp. 5499-5504.

[51] C. Commault, J. M. Dion, O. Sename, and R. Motyeian, "Observer-based fault detction and isolation for structures system", *IEEE Trans. Automatic Control*, vol. 47, no. 12, pp. 2074-2079, December 2002.

[52] H. P. b. Dassanayke, C. Roberts, and C. J. Goodman, "Architecture for system-wide fault detection and isolation," *Proc*. *Institution of Mechanical Eng., Part I: Journal of systems and control Engineering*, vol. 215, no 1, pp. 37-43, 2001.

[53] P. M. Frank, "Fault diagnosis in dynamic system using analytical and knowledge based redundancy- A survey and some results," *Automatica,* vol. *26,* no. 3, pp. 459-474, 1990

[54] E. A. Garcia and P. M. Frank, "Deterministic nonlinear observer-based approaches to fault diagnosis: A survey," *Control Eng. Practice*, vol. 5, no. 5, pp. 663-760, 1997.

[55] H. Hermans, M. Kinneart, and E. H. E Yaagoubi, "Observer-based approach to fault detection and isolation for nonlinear systems," *IEEE Trans*. *Automatic Control*, vol. 44, no. 10, pp. 1879-1884, 1999.

[56] Chee Pin Tany and Maki K. Habib, " Robust sensor fault reconstruction for an inverted pendulum using right eigenstructure assignment," in *Proc. IEEE Int. Conf. on Control Applications,* Taipei, Taiwan, September 2-4, 2004, pp. 1236-1241.

[57] J. Gertler, "Fault detection and isolation using Parity relations," *Control Eng. Practice*, vol. 5, no. 5, pp. 653-661, 1997.

[58] E. Y. Chow and A. S. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Trans. Automatic Control*, vol. 29, no. 7, pp. 603-614, 1984.

[59] V. Cocquempot, M. Staroswiecki, and T. Elmezyani, "Switching time estimation and fault detection for hybrid system using structure parity residuals," in *Proc. IFAC Symposium SAFEPROCESS'03,* Washington, USA, December 2003, pp. 2045-2055.

[60] V. Cocquempot, M. Staroswiecki, and T. Elmezyani, "Fault detection and isolation for hybrid system using Structure Parity Residuals," in *Proc*. *Asia Control Conference* (*ASCC* 2004), Washington, Australia, December 2004, pp. 1203-1211.

[61] H. Ye, G. Wang, and S. X. Ding, "A new parity space approach for fault detection based on stationary wavelet transform," *IEEE Trans. Automatic Control*, vol. 49, no. 2, pp. 281-287, Feb. 2004.

[62] A. Hagenblad, F. Gustafsson, I. Klein, "A comparison of two methods for stochastic fault detection: The parity space approach and principle components analysis," in *Proc. 13th IFAC symposium on system identification* (SYSID 2003), Rotterdam, Netherlands, August 2003, pp. 27-29.

[63] R. Isermann, "Process fault detection based on modeling and estimation methods – A survey," *Automatica*, vol. 20, no. 4, pp. 387-404, 1984.

[64] D. Yogg, E. Shafai, and H. P. Geering, "A fault diagnosis for heat pumps," in *Proc. IEEE International Conf. Control Applications*, Mexico City, September 2001, pp. 70-76.

[65] D. Yogg, "Fault diagnosis for heat pump system," Ph.D. dissertation, ETH Nr. 14402, Zurich, Switzerland, 2001

[66] R. Isermann, "Fault diagnosis of machines via parameter estimation and knowledge processing," *Automatica,* vol. 29, pp. 815-836, 1993.

[67] R. Isermann, "Supervision, fault detection and fault diagnosis methods – An introduction," C*ontrol Eng. Practice*, vol. 5, no. 5, pp. 639-652, 1997.

[68] F. J. Hermans and M. B. Zarrop, "Parameter estimation using sliding mode principles," in *Proc. IFAC Symposium SAFEPROCESS`97*, Hull, UK, 1997, pp. 282-287.

[69] Z. Han and P. Frank, "Physical parameter estimation based FDI with neural networks," in *Proc. IFAC Symposium SAFEPROCESS`97*, Hull, UK, 1997, pp. 294-299.

[70] B. K. Walker and K. Y. Huang, "FDI by extended Kalman filter parameter estimation for industrial actuator benchmark," *Control Eng. Practice*, vol. 3, no. 12, pp. 1769-1774, 1995.

[71] E. P. Gatzke, "Moving horizon estimation using multiple models and qualitative constraints," *Journal of Process Control,* vol. 12, no. 2, pp. 339-352, 2002.

[72] E. Nobrega, M. Abdalla, and K. M. Grigoriadis, "LMI based approach to fault detection and isolation," in *Proc. 39th IEEE conf. on Decision and Control*, Sydney, Australia, 2000, pp. 4329-4334.

[73] D. Sauter and Frederic Hamelin, "Frequency-domain optimization for robust fault detection and isolation in dynamic systems," *IEEE Trans. Automatic Control*, vol. 44, no. 4, pp. 868-882, April 1999.

[74] M. J. Khorowjerdi, R. Nikoukhah, and N. Safari-Shad, "Fault detection in a mixed H_2/ H_∞ setting," *IEEE Trans. Automatic Control*, vol. 50, no. 7, pp. 1063-1068, July 2005.

[75] H. Hammouri, P. Kabore, and M. Kinnaert, "A geometric approach to fault detection and isolation for bilinear systems," *IEEE Trans. Automatic Control,* vol. 46, no. 9, pp. 1451-1455, 2001

[76] A Rakar and D. Juricic, P. Balle, "Transferable belief model in fault diagnosis," *Engineering Application of Artificial Intelligence,* vol. 12, no. 5, pp. 555-567, 1999

[77] A. Rakar and D. Juricic, "Diagnostic reasoning under conflicting data: the application of the transferable belief model," *Journal of Process Control,* vol. 12, no. 1, pp. 55-67, 2002.

[78] J. Chen and J. R. Patton, *Robust Model-Based Fault Diagnosis for Dynamic System.* Kluwer, 1999

[79] A. Emani-Naeini, M. M. Athter, and S. M. Rock, "Effect of model uncertainty on failure detection: the threshold selection," *IEEE Trans. Automatic Control*, vol. 33, no. 12, pp. 1106-1115, 1988.

[80] B. Jiang, J. Wang, and Y. Chai Soh, "Robust fault diagnosis for a class of linear systems with uncertainty," *AIAA Journal of Guidance, Control and Dynamics*, vol. 22, pp. 736-741, 1999.

[81] B. Jiang, J. Wang, and Y. Chai Soh, "Adaptive technique for robust diagnosis of faults with independent effects on system outputs," *Int. Journal of Control*, vol. 75, no. 11, pp. 792-802, 2002.

[82] B. Darkhovski and M. Staroswiecki, "A game-theoretic approach to decision in FDI," *IEEE Trans. on Automatic Control*, vol. 48, no. 5, pp. 853-858, 2003.

[83] W. H. Chung and J. L. Speyer, "A game theoretic fault detection filter," *IEEE Trans. on Automatic Control*, vol. 43, no. 2, pp. 143-161, Feb. 1998.

[84] Hao Ye, S. X. Ding, and G. Wang, "Integrated design of fault detection system in time-frequency domain," *IEEE Trans. Automatic Control,* vol. 47, no. 2, pp. 384-390, Feb 2002

[85] R. M. Agustin, R. S. Mangoubi, Roger M. Hain, and Neil J. Adams, "Robust failure detection for reentry vehicle attitude control systems," *AIAA Journal of Guidance and Dynamics,* vol. 22, no. 6, pp. 49-61, November-December 1999.

[86] B. Rinner, and U. Weiss, "Online monitoring by dynamically refining imprecise models," *IEEE Trans. system, Man and Cybernetics-Part B: Cybernetics,* vol. 34, no. 4, pp. 1811-1822, August 2004.

[87] A. Alessandri, T. Hawkinson, A. J. Healey, and G. Veruggio, "Robust model-based fault diagnosis for unmanned underwater vehicles using sliding mode-observers," *11th International Symposium on Unmanned Undeterred Submersible Technology* (UUST'99), August 22-25, 1999.

[88] X. Zhang, M. M. Polycarpou, and T. Parisini, "A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems," *IEEE Trans. Automatic Contro*l, vol. 47, no. 4, pp. 576-593, April 2002.

[89] M. Mahmoud, J. Jiang, and Y. Zhang, "Active fault tolerant control systems: Stochastic analysis and synthesis," Lecture Notes in Control and Information Science, Berlin: Springer, 2003.

[90] Y. M. Zhang and J. Jiang, "Design of integrated fault detection, diagnosis and fault-tolerant control systems," in *Proc. 38th IEEE conf. Decision and Control,* Phoenix, Arizona, USA, 1999, pp. 3593-3598.

[91] M. Bodson, "Multivariable adaptive algorithms for reconfigurable flight control," *IEEE Trans. Control System Technology*, vol. 5, no. 2, pp. 217-229. March 1997.

[92] H. Noura, D. Sauter, Frederic Hamelin, and Didier Theilliol, "Fault-tolerant control in dynamic systems: Application to a winding machine," *IEEE Control System Magazine,* vol. 20, no. 1, pp. 33-49, Feb. 2000

[93] R. J. Patton, "Fault-tolerant control: the 1997 situation," in *Proc. IFAC Symposium SAFEPROCESS'97*, Hull, UK, 1997, pp. 1033-1055.

[94] S. M. Joshi, "Design of failure accommodation multi-loop LQR type controller," *IEEE Trans. Automatic Control*, vol. 32, no. 8, pp. 740-741, 1987.

[95] R. J. Veillette, L. V. Medanic, and W. R. Porking, "Design of reliable control systems," *IEEE Trans. Automatic Control,* vol. 37, no. 3, pp. 290-304, 1992.

[96] Q. Zhao and J. Jiang, "Reliable state feedback control design against actuator failures," *Automatica*, vol. 34, no. 10, pp. 1267-1272, 1998.

[97] M. Modson and J. Groszkiewiez, "Multivariable adaptive algorithm for reconfigurable flight control," *IEEE Trans. Cont. Sys. Tech.*, vol. 5, no. 2, pp. 217-229, 1997.

[98] G. Tao, S. M. Joshi, and X. Ma, "Adaptive state feedback and tracking control of system with actuator failures," *IEEE Trans. Automatic Control*, vol. 46, no. 1, pp. 78-95, January 2001.

[99] A. Marx, D Kornig, and D. Georges, "Robust fault-tolerant control for descriptor systems," *IEEE Tarns on Automatic Control*, vol. 49, no. 10, pp. 1869-1876, Oct. 2004.

[100] K. Zhou and Z. Ren, "New controller architecture for high performance robust, and fault tolerant control," *IEEE Trans. on Automatic control,* vol. 46, no. 10, pp. 1613-1618, Oct. 2001.

[101] J. Stoustrup and V. D. Blondel, "Fault tolerant control: A simultaneous stabilizing results," *IEEE Trans. Automatic Control,* vol. 49, no. 2, pp. 305-310, Feb. 2004.

[102] Y. Zhang and J. Jiang, "Fault tolerant control system design with explicit consideration of performance degradation," *IEEE Trans. Aerospace and electronic system*, vol. 39, no. 3, pp. 838-848, June 2003.

[103] M. Blanke, M. Staroswiecki, and N. Eva Wu, "Concepts and methods in fault-tolerant control," *Invited Tutorial at American Control Conf.*, Washington, USA, June 2001, pp. 22-24.

[104] M. Mahmoud, J. Jiang, and Y. Zhang, "Stabilization of active fault tolerant control systems with imperfect fault detection and diagnosis," *Stochastic Analysis and Application*, vol. 21, no. 3, pp. 673-701, 2003.

[105] P. S. Mayback and R. D. Stevens, "Reconfigurable flight control via multiple model adaptive control methods," *IEEE Trans. Aerospace and Electronic Systems,* vol. 27, no. 2, pp. 470-479, 1991.

[106] D. D. Moerder, J. R. B. N. Halyo, and A. K. Caglayan, "Application of precomputed control laws in reconfigurable aircraft flight control system," *Journal of Electronics*, vol. 45, no. 3, pp. 325-333, 1989.

[107] H. E. Rauch, "Autonomous control reconfiguration," *IEEE Control Systems Magazine,* vol. 15, no. 6, pp. 37-48, 1995.

[108] Y. Zhang and J. Jiang, "Integrated active fault-tolerant control using IMM approach," *IEEE Trans. Aerospace and Electronic Systems,* vol. 37*,* no. 4*,* pp. 1221-1235, 2001.

[109] M. Maki, J. Jiang, and K. Hagino, "Stability guaranteed active fault tolerant control system against actuator failure," in *Proc. 40th IEEE conf. Decision and control, Orlando*, FL, USA, 2001, pp. 1893-1898.

[110] Z. Gao and P. J. Antsaklis, "Pseudo-inverse method for reconfigurable control with guaranteed stability," in *Proc. 11th IFAC World Congress on Automatic Control*, Tallinn, USSR, 1990, pp. 16-22.

[111] Z. Gao and P. J. Antsaklis, "Stability of pseudo-inverse method for reconfigurable control systems," *Int. Journal of Control,* vol. 53, no. 3, pp. 717-729, 1991.

[112] J. Jiang, "Design of reconfigurable control systems using eigenstructure assignment," *Int. Journal of Control,* vol. 59, no. 2, pp. 395-410, 1994.

[113] Y. Zang and J. Jiang., "Integrated design of reconfigurable fault-tolerant control systems," *AIAA Journal of Guidance, Control and Dynamics*, vol. 24, no. 1, pp. 133-136, 2001.

[114] H. Noura, D. Theilliol, and D. Sauter, "Actuator fault-tolerant control design: Demonstration on a three-tank-system," *Int. Journal of Systems Science*, vol. 31, pp. 1143-1155, 2000.

[115] W. D. Morse and K. A. Ossman, "Model following reconfigurable flight control system for the AFTI/F-16," *AIAA Journal of Guidance, Control and Dynamics*, vol. 13, no. 6, pp. 969-976, 1990.

[116] C. Y. Huang and R. F. Stengel, "Restructurable control using proportional-integral implicit model following," *AIAA Journal of Guidance, Control and Dynamics*, vol. 13, no. 2, pp. 303-309, 1990.

[117] W. K. Son, O. K. Kwon, and M. E. Lee, "Fault tolerant model based predictive control with application to boiler systems," in *Proc. IFAC Symposium, SAFEPROCESS'97,* Hull U.K., 1997, pp. 1240-1245.

[118] J. M. Maciejowski, "Modeling and predictive control: enabling technologies for reconfiguration," in *proc. IFAC Symposium System Structure Control*, October 1997, pp. 253-258.

[119] J. M. Maciejowskiand and C.N. Jones, "MPC fault-tolerant flight control case study: Flight 1862," in *Proc. IFAC Safe Process Conf.*, Washington DC, 9-11 June 2003.

[120] U. Demirci and F. Kerestecioglu, "Fault tolerant control with re-configuring sliding- mode schemes," *Turkish J. Electric Engineering*, vol. 13, no. 1, pp. 175-187, 2005.

[121] J. Stoustrup and H. Niemann, "Fault tolerant feedback control," in *Proc. Eropean Control Conf.*, Portugal, September 2001, pp. 1970-1974.

[122] J. Stoustrup and H. Niemann, "Reliable control using the primary and dual Youla parameterizations," in *Proc.* 41st *IEEE Conf. on Decision and Control*, Las Vegas, NV, vol. 4, December 2002, pp. 4353-4350.

[123] J. Petersen, M. Bodson, "Constrained quadratic programming techniques for control allocation," *IEEE Trans. Control System Technology,* vol. 14, no. 1, pp. 91-98, 2006.

[124] J. Petersen, M. Bodson, "Interior-point algorithm for control allocation," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 28, no. 3, pp. 471-480, 2005.

[125] W. Chen and J. Jang, "Fault-tolerant control against stuck actuator faults," *IEE Control Theory and Application*, vol. 152, no. 2, pp. 138-146, March 2005.

[126] Y. Ochi and K. Kanai, "Design of restructurable flight control using feedback linearization," *AIAA Journal of Guidance, Control and Dynamics*, vol. 14, no. 5, pp. 903-911, 1991

[127] X. Q. Xie, D. H. Zhou, Y. H. Jin, and B. D. Liu, "Sensor adaptive fault tolerant control for non-linear processes," *Int. Journal of Systems Science,* vol. 33, no. 5, pp. 313-321, 2002.

[128] X. Zhang, T. Parisini and M. Polycarpou, "Adaptive Fault-Tolerant control of nonlinear uncertain systems: An information-based diagnostic approach.," *IEEE Trans. on Automatic Control,* vol. 49, no. 8, pp. 1259-1274, August 2004.

[129] Ding-Li Yu, T. K. Chang, and Ding-Wen Yu, "Fault tolerant control of multivariable processes using auto-tuning PID controller," *IEEE Trans. Man, and Cybernetics, Part B: cybernetics*, vol. 35, no. 1, pp. 32- 43, Feb. 2005.

[130] F. Ahmed-Zaid, P. Ioannou, K. Gousman, and R. Rooney, "Accommodation of failure in F-16 aircraft using adaptive control," *IEEE Control System Magazine,* vol. 11, no. 1, pp. 73-78, 1991.

[131] M. Bodson, "Reconfigurable nonlinear autopilot," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 26, no. 5, pp. 719-727, September-October 2003.

[132] C. Bonivento, A. Isidori, L. Marconi, and A. Paoli, "Implicit fault-tolerant control: application to induction motors," *Automatica,* vol. 40, no. 3, pp. 335-371, 2005.

[133] Y Diao and K. M. Passino, "Stable fault-tolerant adaptive fuzzy/neural control for a Turbine Engine," *IEEE Trans. Control System Technology*, vol. 9, no. 3, pp. 494-509, May 2001.

[134] E. Badreddin and M. Abdel-Geliel, "Dynamic safety margin principle and application for safety critical system," *in Proc*. *IEEE Int. Conf. Control Application*, August 1-4, 2004, pp. 689-694.

[135] M. Abdel-Geliel and E. Badreddin, "Adaptive controller using dynamic safety margin for hybrid laboratory plant," in *Proc. American Control Conf.*, Portland, Oregon, USA, June 2005, pp.1443-1448.

[136] M. Abdel-Geliel and E. Badreddin, " Dynamic safety margin in fault diagnosis and isolation," in *Proc. European Safety and Reliability Conference* (ESREL2005), Tri city Poland, June 2005, pp.1-6.

[137] M. Abdel-Geliel, E. Badreddin, and A. Gambier, "Dynamic safety margin in fault-tolerant predictive controller," in *Proc. IEEE Int. Conf. Control Application*, Toronto, Canada, Sep. 2005, pp.803-808.

[138] M. Abdel-Geliel, E. Badreddin and A. Gambier, "Application of model predictive control for fault tolerant system using dynamic safety margin," in *Proc. American Control Conf.*, Minneapolis, Minnesota, USA, June 2006, pp. 5493-5498.

[139] M. Abdel-Geliel, A. Gambier and E. Badreddin, "Adaptive model predictive control based on dynamic safety margin for fault tolerant system," in *Proc.* 6th *Asian Control Conf.*, Bali, Indonesia, July 2006, pp. 375-383.

[140] M. Abdel-Geliel, E. Badreddin and A. Gambier, "Robust fault detection based on dynamic safety margin," in *Proc.* 6th *Asian Control Conf.*, Bali, Indonesia, July 2006, pp. 367-374.

[141] M. Abdel-Geliel, E. Badreddin and A. Gambier, "Application of dynamic safety margin in robust fault detection and fault tolerant control," to be *published in IEEE International Conf. Control Application,* Munich, Germany, 2006.

[142] R. R. Leitch, Q. Shen, G.M. Coghill, and M.j. Chantler, "Choosing the right model," *IEE Control Theory App*lication, vol. 146, no. 5, September 1999.

[143] R. A. Horn and C. R. Jonson, *Matrix Analysis.* Cambridge University Press, 1985.

[144] F. Blanchini, "Set invariance in control," *Automatica,* vol. 35, no. 11, pp. 1747-1767, 1999.

[145] E. Kerrigan, "Robust constraint satisfaction: Invariant set and predictive control," PhD thesis, University of Cambridge, 2000

[146] G. Bitsoris and E. Gravalou, "Design techniques for the control of the discrete-time system subject to state and control constraints," *IEEE Trans. Automatic Control*, vol. 44, no. 5, pp. 1057-1061, May 1999.

[147] C. N. Jones, E. C. Kerrigan, and J. M. Maciejowski, "Equality set projection: A new algorithm for the projection of polytope in half-space representation," Technical Report CUED/F-INFENG/TR.463, University of Cambridge, March 2004.

[148] F. Blanchini, " Feedback control for linear time-invariant systems with state and control bounds in the presence of disturbances," *IEEE Trans. Automatic Control*, vol. 35, no. 11, pp. 1231-1233, Nov. 1990.

[149] P. O. Gutman and M. Cwikel, "Admissible sets and feedback control for discrete-time linear dynamical systems," *IEEE Trans. Automatic Control*, vol. 31, no. 4, pp. 373-376, 1986.

[150] Elena De Santis, "On positively invariant sets for discrete-time linear systems with disturbance: An application of Maximal disturbance sets," *IEEE Trans. Automatic Control*, vol. 39, no. 1, pp. 245-249, 1994.

[151] E. Badreddin, "Analogical gates: A network logical gates approach to fuzzy control with applications to a non-holonomic autonomous mobile robot," *Int. Journal of Intelligent Automation and Soft Computing*, 1997

[152] Frank L. Lewis, Vassilis L. Syrmos, *Optimal Control,* 2nd ed. New York: John Wiley, 1995.

[153] D. Sauter, F. Hamelin, and H. Noura, "Fault tolerant control in dynamic system using convex optimization.," in *Proc. IEEE ISIC/CIRA/ISAS Joint Conf., Gaithersburg, MD*, 1998, pp. 187-192.

[154] G. Simon, G. Karsai, G. Biswas, S. Abdelwahed, N. Mahadevan, T. Szemeth, G. Peceli and T. Kovacshaz, " Model-based fault-adaptive control of complex dynamic system," *IMTC 2003- Int. Measurement and Technology Conf.*, Vail, Co, USA, May 2003, pp. 20-22.

[155] T. Miksch, "Modelling and implementation of experimental plant," Diploma thesis, Mannheim University, Mannheim, Germany, July 2003

[156] A. Gambier, T. Miksch, and E. Badreddin, "A control laboratory plant to experiment with hybrid system," in *Proc. American Control Conf.*, Denver, USA, 2003.

[157] S. Abdelwahed, G. Karsai, and G. Biswas, "Online safety control of a class of hybrid systems," in *Proc. 41st IEEE Conf. Decision and Control*, Las Vegas, USA, December 2002, pp. 1988-1990.

[158] E. G. Gilbert and K. Tim Tan, "Linear systems with state and control constraints: the theory and application of maximal output admissible sets," *IEEE Trans. Automatic Control*, vol. 36, pp. 1008-1020, 1991.

[159] M. V. Kothare, "Control of system subject to constraints," Ph.D. dissertation, California institute of technology, Pasadena, California, 1997.

[160] C.T. E. Dorea and B. E. Milani, "Design of L-Q regulators for state constrained continuous-time system," *IEEE Trans. Automatic Control,* vol. 40, no. 3, pp. 544-548, 1995.

[161] G. A. Murad, I. Posthethwaite, and D. W. Gu, "A robust design approach to integrated control and diagnostics," in *Proc. 13th IFAC Word Congress Automatic Control, San Francisco*, CA, 1996, pp. 199-204.

[162] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, no. 6, pp. 601-611, 1976.

[163] R. N. Clark, "Instrument fault detection," *IEEE Trans. Aero. Electronic System*, vol. 14, pp. 456-465, May 1978.

[164] G. Delmaire, P. Gasser, M. Staroswiecki, and C. Christophe, "Comparison of multivariable identification and parity space techniques for FDI purpose in MIMO systems," *European Control Conf.* (*ECC'99), Karlsruhe, Germany*, 1999.

[165] L. T. Lai, "Sequential multiple hypothesis testing and efficient fault detection isolation in stochastic systems," *IEEE Trans. Information Theory*, vol. 46, no. 2, pp. 695-700, 2000.

[166] X. Lou, A. Willsky, and G. Verghese, "Optimal robust redundancy relations for failure in uncertainty systems," *Automatica,* vol. 22, no. 3, pp. 333-344, 1986

[167] G. Duan, D. How, and J. R. Patton, "Robust fault detection in descriptor system via generalised unknown input observers," *Int. J. Systems Science,* vol. 33, no. 5, pp. 369-377, 2002.

[168] R. X. Li, "Hybrid estimation techniques," *Control and Dynamic Systems,* vol. 76, C. T. Leondes*, Ed.* New York: Academic Press, 1996, pp. 213-287.

[169] P. S. Maybeck and P.D. Hanlon, "Performance enhancement of a multiple model adaptive estimator," *IEEE Trans. Aerospace and Electronic Systems*, vo. 31, no. 4, pp. 1240-1254, Oct. 1995.

[170] T.E Menke and P.S. Maybeck, " Sensor/actuator failure detections in the Vista F-16 by multiple model adaptive estimation," *IEEE Trans. on Aerospace and Electronic Systems,* vol. 31, no. 4, pp. 1218-1229, Oct. 1995.

[171] Y. H. Zhang, X. R.Li G. Z. Dai, H. C. Zhang, and J. Chen, "Fault detection and identification of dynamic systems using multiple feed-forward neural networks," in *Proc. 13th IFAC World Congress on Automatic Control, San Francisco, C*A, June 30-July 5, 1996, pp. 241-246.

[172] I. D. Magill, "Optimal adaptive estimation of sampled stochastic processes," *IEEE Trans. Automatic Control*, vol. 10, no. 4, pp. 234-439, 1965.

[173] Karl J. Astrom and B. Wittenmark, *Adaptive Control,* 2nd ed. Addison-Wesley, 1995.

[174] C. C. Hang, K. J. Astrom, and Q. G. Wang, "Relay feedback auto-tuning of process controllers-a tutorial review," *Automatica,* vol. 12, no. 1, pp. 134-162, 2002.

[175] W. K. Ho, C. C. Hang, and L. S. Cao, "Tuning of PID controllers based on gain and phase margin specifications," *Automatica,* vol. 31, no. 3, pp. 497-502, 1995.

[176] W. D. Chang, R. C. Hwang, and J. G. Hsieh, "Auto-tuning PID controller for a class of non-linear system based on Lyapunov approach," *J. Process Control,* vol. 12, pp. 233-242, 2002.

[177] F. Radke and R. Isermann, "A parameter adaptive PID controller with stepwise parameter optimization," *Automatica*, vol. 23, no. 4, pp. 449-457, 1987.

[178] T. H. Lee, C. C. Hang, W. K. Ho, and P. K. Yue, "Implementation of a knowledge-base PID auto-tunner," *Automatica,* vol. 29, no. 4, pp. 1107-1113, 1993.

[179] E.F. Camacho and C. Bordons, *Model Predictive Control*. Springer, 1999.

[180] J.A Rossiter, *Model-Based Predictive Control: Practical Approach*. CRC, 2003.

[181] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2002.

[182] D. Q. Mayne, J. B. Rawlings, C. V. Rao and Po. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789-814, 2000.

[183] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming-the explicit solution," *IEEE Trans. Automatic Control*, vol. 47, no. 12, pp. 1974-1985, December 2002.

[184] F. Borrelli, "Constrained optimal control of linear and hybrid systems," Lecture Notes in Information Science, Springer, 2003.

[185] A. Bemporad, M Morari, V. Dua, and E.N. Pistikopoulos, "The explicit linear quadratic regulator for constrained system," *Automatica,* vol. 38, no. 1, pp. 3-20, March 2003.

[186] A. Bemporad, M. Morari, V. Due, and E.N. Pistikopoulos, "The explicit solution of model predictive control via multi-parametric quadratic programming," in *Proc. America Control Conference*, Chicago, vol. 2, June 2000, pp. 872-876.

[187] A. T. Johansen, I. Petersen, and O. Slupphaug, "Explicit suboptimal linear quadratic regulation with input and state constraints," *Automatica,* vol. 38, no. 7, pp. 1099-1111, 2002.

[188] A. Bemporad and C. Filippi, "Suboptimal explicit RHC via approximate multi-parametric quadratic Programming," *Journal of Optimization Theory and Application,* vol. 117, no. 1,2003

[189] P. Tondel, T. A. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree," *Automatica*, vol. 39, no. 5, 00. 945-950, 2003.

[190] P. Tondel, T. Arne, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MPC solutions," *Automatica,* vol. 39, no. 3, pp. 489-497, 2003.

[191] S. J. Qin and T. A. Badgewell, "An overview of industrial model predictive control technology," in *5th Int.Conf.Chemical Process*, AIChE symposium series 316, 1997, pp. 232-256.

[192] P. O. M. Scokaert and J. B. Rawlings, "Feasibility issues in model predictive control", *AIChE Journal,* vol. 45, no. 8, pp. 1645-59, 1999.

[193] P. O. M. Scokaert, "Constrained predictive control," Ph.D. dissertation, university of Oxford, UK, 1994.

[194] M.L. Tyler and M. Morari, "Proposition logic in control and monitoring problem," *Automatica*, vol. 35, pp. 565-582, 1999.

[195] E.C. Kerrigan, A. Bemporad, D. Mignone, M. Morari, and J. M. Macieowski, "Multi-objective prioritisation and reconfiguration for the control of constrained hybrid systems," in *Proc. American Control Conf., C*hicago, IL, USA, vol. 3, June 2000, pp. 1694-1698.

[196] J. Vada, O. Slupphaug, and B. A. Foss, "Infeasibility handling in linear MPC subject to prioritized constraints," *14th IFAC World Congress on Automatic Control,* Beijing, China, 1999, pp. 163-168,

[197] J. Vada, O. Slupphaug, and T.A. Johansen, "Efficient optimal prioritized infeasibility handling in model predictive control – a parametric preemptive _multi-objective linear programming approach," *Journal of Optimization theory & Application*, vol. 109, no. 2, pp. 385-413, 2001.

[198] E. Kerrigan and J. Maciejowski, "Designing model predictive controller with prioritised constraints and objectives," in *Proc. IEEE Conf. Computer Aided control System and Design*, Sept 2002, pp.33-38.

[199] J. Vada, O. Slupphaug, T.A. Johansen, and B.A. Foss, "Linear MPC with optimal prioritized infeasibility handling: Application computation issues and stability," *Automatica,* vol. 37, pp. 1835-1843, 2001.

[200] M. Diehl and J. Björnberg, "Robust dynamic programming for Min-Max model predictive control of constrained uncertain system," *IEEE Trans. Automatic Control*, vol. 49, no. 12, pp. 2253-2257, December 2004.

[201] E. Kerrigen and I. Maciejowski, "Feedback Min-Mx MPC using a single linear program: Robust stability and explicit solution," *International Journal of Robust and Nonlinear Control*, vol. 14, pp. 395-413, 2004.

[202] P.O. Scokaert and D. Q. Mayne, "Min-Max Feedback Model Predictive Control for Constrained Linear Systems," *IEEE Trans. Automatic Control*, vol. 43, no. 8, pp. 1136-1142, August 1999.

[203] G. Franklin, J. D. Powell, and M. Workman, *Digital control dynamic systems,* 3rd ed. Addison Wesley, 1998.

[204] J.M. Maciejowski, "Reconfiguring Control system by optimization," in *Proc. European Control Conference (ECC),* Brussels, 1997

[205] A. Gambier and H. Unbehauen, "Multivariable generalized state-space receding horizon control in a real-time environment," *Automatica,* vol. 35, no. 11, pp. 1787-1797, 1999.

[206] A. Gambier, "Sate-space design of predictive control for MIMO systems," Ph.D. dissertation, Bochum University, Germany: Cuvillier Verlag Göttingen, 1995.

[207] L. H. Tsoukalas and R. E. Uhrig, Fuz*zy and neural approaches in engineering.* New York: John Wiley and Sons, 1997.

[208] R. Keten, *Getting stated with Qnx Neutrino: a guide for real-time programmers*. Parse Software devices, 2001.

[209] RT-Lab documents. [online]. Available: http://www.opal-rt.com