Fault Diagnosis and Fault Tolerant Control of Hybrid Systems

Tabatabaeipour, Seyed Mojtaba

Publication date:
2010

Document Version
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

Citation for published version (APA):
Tabatabaeipour, S. M. (2010). *Fault Diagnosis and Fault Tolerant Control of Hybrid Systems*.

Seyedmojtaba Tabatabaeipour

Fault Diagnosis and Fault Tolerant Control of Hybrid Systems

AALBORG UNIVERSITY

Fault Diagnosis and Fault Tolerant Control of Hybrid Systems
Ph.D. thesis

ISBN: 978-87-92328-35-9
June 2010

Contents

Preface and Acknowledgements

This thesis is submitted in partial fulfillment of the requirements for the Doctor of Philosophy in Automation and Control at Aalborg University. The work has been carried out in the period from June 2007 to June 2010 under supervision of Professor Thomas Bak, Professor Anders. P. Ravn, and Lead Expert in Control Engineering Roozbeh Izadi-Zamanabadi.

I would like to thank my supervisors for excellent guidance and support. Their knowledge, fruitful discussions, and help, from the initial stage of the project until the last stage of the project, has been a great resource to my accomplishment.

I would like to thank CISS and MoDES and the section for Automation and Control for supporting my research.

I was a guest at the Institute för Automatic, ETHZ, Zürich from September 2009 to February 2010. I would like to show my gratitude to Professor Manfred Morari who made my stay possible and to all people there specially Colin Jones for his time and helpful discussions.

Furthermore, I would like to thank all staff at the department of Control and Automation for creating a friendly environment.

I am very grateful to my friends specially Hamidreza Shaker, and Mehdi Gholami, for their support during these years.

Finally, I owe my deepest gratitude to my parents who, with their attitude towards life, gave me the most valuable education I have ever had.

Abstract

Modern technological systems consist of many components with strong interactions between them. The functionality and performance of the overall system depends on the performance of each component. A fault in a component may decrease the overall performance of the system and it may even lead to an unacceptable loss of the system functionality or instability. Therefore, it is important to design control systems that can automatically detect and diagnose occurred faults, maintain the overall functionality of the system, and ensure an acceptable performance for the faulty system.

Most advanced technological systems, include subsystems with continuous behavior and subsystems with discrete behaviors and interactions between them. Hybrid systems are a useful class for modeling of these systems . A hybrid system is a dynamical system with both continuous and discrete behaviors and non-trivial interactions between continuous evolutions and discrete transitions. Hybrid systems arises in many real world applications such as manufacturing, chemical process, traffic control, robotics, etc.

This thesis develops methods for Fault Detection and Diagnosis (FDD) and Fault Tolerant Control (FTC) of hybrid systems. In the area of FDD, we propose two methods for active diagnosis of hybrid systems. The first approach uses reach set computation to find the shortest test signal that can diagnose a fault. The approach does not guarantee the stability of the system. This is an important issue in active diagnosis, because we are exciting the system and perturbing it from the operating point; moreover, because during the diagnosis the system is controlled with the nominal controller which might not keep the faulty system stable. Therefore, in the second method, we propose an optimization based active diagnosis method that guarantees stability of the system. Stability of the system is ensured by superimposing a model predictive controller on the active diagnoser and by imposing the constraint that the state of the system after diagnosis should be in the feasible set of the model predictive controller. Feasibility of this constraint means that the systems is diagnosable and stabilizable. After diagnosing the fault, system reconfiguration is performed by updating the model predictive controller constraints based on the model of the faulty system. The generated test signal can be used for sanity check of the system at the commissioning phase or for checking the system periodically during the normal operation.

In the area of FTC, we propose a new method for passive FTC (PFTC) of PieceWise Linear (PWL) systems. We use PieceWise Quadratic Lyapunov (PWQL) functions. The approach provides an upper bound on the performance of the closed loop system which can be minimized using an optimization problem with Linear Matrix Inequality (LMI) constraints. This method uses PWL state feedback for controller design. But, states of a system are not available usually. Therefore, we propose a method for PFTC of PWL systems using output feedback. For output feedback the problem is formulated in terms of

Bilinear Matrix Inequalities (BMIs) and an optimal upper bound on the performance can be found using optimization with BMI constraints which is solved using the V-K iteration algorithm.

Synopsis

Moderne teknologiske systemer består af mange komponenter i tæt samarbejde. Funktion og ydeevne af det samlede system afhænger af performance af hver komponent. En fejl i en komponent kan nedsætte det samlede systems performance, og det kan føre til et uacceptabelt tab af systemets funktionalitet eller ustabilitet. Derfor er det vigtigt at designe reguleringssystemer, der automatisk kan opdage og diagnosticere fejl, bevare den samlede funktionalitet af systemet, og sikre en acceptabel performance for det fejlbehæftede system.

De fleste avancerede teknologiske systemer, omfatter delsystemer med kontinuert adfæd og delsystemer med diskrete adfæd og et samspil mellem dem. Hybride systemer er en nyttig klasse for modellering af disse systemer. Et hybrid system er et dynamisk system med både kontinuert og diskret adfæd og et ikke-trivielt samspil mellem den kontinuerte udviklingen og diskrete overgange. Hybrid systemer opstår i mange applikationer såsom produktion, kemisk processer, trafik kontrol, robotteknologi, etc.

Denne afhandling omhandler metoder til fejlfinding og diagnosticering (FDD) og fejltolerant regulering (FTC) af hybride systemer. Inden for FDD, foreslår vi to metoder til aktiv diagnosticering af hybride systemer. Den første strategi benytter mængde beregninger for at finde det korteste test signal som kan diagnosticere en fejl. Den fremgangsmåde kan ikke garantere systemets stabilitet. Dette er et vigtigt spørgsmål i aktiv diagnose, fordi vi påvirker systemet og flytter det fra driftspunktet. Endvidere er systemet under diagnosen reguleret af den nominelle regulator, som måske ikke kan stabilisere det fejlbehæftede system. Derfor foreslår vi en anden aktiv diagnose metode baseret påoptimering, der sikrer stabilitet af systemet. Systemets stabilitet er sikret ved at overlejre en model prædiktiv regulator pådet aktive diagnosesystem og ved at pålægge den begrænsning, at tilstanden af systemet efter diagnose skal være i det tilladelige mængde for den model prædiktive regulator. Denne begrænsning indebærer, at systemerne er diagnosticerbare og stabiliserbare. Efter at have diagnosticere fejl, reconfigureres systemet ved at opdatere den model prædiktive regulators begrænsninger baseret påmodellen af det fejlbehæftede system. Det genererede test signal kan bruges til sanity check af systemet ved indkøring eller for at kontrollere systemet jævnligt under normal drift.

Inden for FTC, foreslår vi endvidere en ny metode til Passiv FTC (PFTC) af stykkevis lineære (PWL) systemer. Vi bruger stykkevis kvadratiske Lyapunov funktioner. Denne fremgangsmåde giver en øvre grænse for performance af det tilbagekoblede system, der kan minimeres ved at se pådet som et optimerings problem med linear matrix ulighed (LMI) begrænsninger. Denne metode bruger PWL tilstandstilbagekobling til regulator design. Da tilstandstilbagekobling normalt ikke er direkte tilgængeligt foreslår vi en metode til PFTC af PWL systemer, der anvender output tilbagekobling. Output tilbagekoblingsproblemet formuleres i form af Bilinear matrix uligheder (BMI'er) og en optimal øvre grænse

for performance kan findes ved hjælp af en optimering med BMI begrænsninger, der er løst ved hjælp af V-K iteration algoritmen.

1 | Introduction

There is an increasing demand for reliability, safety, and performance of modern technological systems. Therefore Fault Detection and Diagnosis (FDD) and Fault Tolerant Control (FTC) of them is very important. Most modern technological systems consist of both discrete and continuous behaviors and interactions between them. Hybrid systems are a useful modeling class to capture behavior of these systems. In this thesis, we investigate the problem of FDD and FTC for hybrid systems.

This chapter describes the motivation for studying the problem and provides basic concepts of fault detection and diagnosis and fault tolerant control and gives an overview of the state of art in this field.

1.1 Motivation

Every system is prone to fault. A change in a component of a system that changes its behavior from its nominal behavior is called a fault. A fault may decrease the performance of the whole system. It might lead to an unacceptable performance or in serious cases it might yield shut-down of the system or instability and damages. In modern technological systems, there is a high demand on performance, safety, and reliability of systems. It is desired that if a fault happens, the control system can automatically detect the fault and moderate its effect on the system such that it can continue working while providing an acceptable performance. If an acceptable performance is not possible, it should be able to preserve the overall functionality and stability of the systems while allowing some degradation in the performance of the system. In any case, it is important to avoid dangerous areas to prevent damages to the system. Therefore, FDD and FTC are very important for modern technological systems.

The initiative problem for the research in this thesis stems from the funding project which is concerned with developing self-validating reconfigurable control systems. An aim of the project is to develop a control system that can provide sanity checks at the commissioning phase of the operation by methodically checking involved instrumentation functionalities. In a large system there are many sensors, actuators and other components. Every measurement from a sensor or output to an actuator should be assigned correctly to its corresponding variable in the control algorithm. Yet, it happens that a technician connects components of a system wrongly. Wrong sensor or actuator assignment potentially results in malfunction of the overall system. Therefore, it is desirable to design a controller which provides sanity check in the commissioning phase for verifying sensor and actuator assignment by generating an appropriate test signal. A way to address this problem is to use active diagnosis methods for generating a test signal to check the sanity of components as well as sensor and actuator assignments during the commissioning. Although the behavior of some systems during the normal operation can be approximated by a linear system around the operating point, but most of the industrial systems present a hybrid behavior during the commissioning phase.

Faults might also happen during the operation phase of the system. One way to tackle the problem is to diagnose the fault, using active or passive diagnosis, and then re-design the controller for the faulty system. Control re-design could be performed online or it could be a switching between a bank of pre-designed controllers for a fault or a set of faults. This method is called Active Fault Tolerant Control (AFTC). Another way to address the problem is to design a fixed controller such that it can tolerate some faults in the system. This is called Passive Fault Tolerant Control (PFTC). There is always some delay associated with detection and diagnosis of a fault. During the time period in which a fault occurs and is diagnosed, the system is working with the nominal controller. The faulty system with the nominal controller might become unstable during this period. For safety critical systems, the time window in which the system remains stable is too small for accurate fault detection and diagnosis. In this cases a PFTC is preferable. In practice, a combination of both methods are required. Some non-severe faults should be handled with PFTC and severe faults with AFTC. Moreover, many times, AFTC consists of switching between a bank of pre-designed controllers where each one is designed to handle a set of faults. In this cases, each of these controllers is a PFTC which can tolerate a set of faults.

In last three decades a lot of research has been carried out in the area of FDD and FTC. Many methods for FDD and FTC of discrete event systems or continuous systems are proposed. But many systems include both discrete and continuous behaviors. These systems are called hybrid systems. Hybrid systems are systems which contain both continuous behaviors and discrete behaviors such that there is non-trivial interactions between continuous evolutions and discrete transitions. Generally speaking, a hybrid system consists of several modes. In each mode, the system has a continuous dynamic. Transition between these modes happens if the continuous state of the system satisfies some conditions such as entering a region or hitting a guard or it could be based on time or based on some conditions on the input or a combination of them. This transition is described by a switching logic. This is illustrated in Fig. 1.1. Hybrid systems appear in many engineering applications such as mechanical systems, circuits systems, chemical processes, embedded systems, manufacturing, traffic control, etc. Hybrid systems have attracted a lot of research in recent years. Many works have studied modeling, simulation, stability analysis, verification, reachability analysis and control design of hybrid systems, see [AK03] and references therein.

FDD and FTC of Hybrid systems have attracted some research in recent years. Classical methods on FDD and FTC can not be applied directly to hybrid systems because at one hand, discrete event methods abstract the continuous behavior of a hybrid systems to discrete events and ignore the continuous behavior of the system. On the other hand, continuous systems methods do not deal with switching between modes and discrete behavior of a hybrid system. Therefore, It is necessary to develop methods that consider the behavior of a hybrid systems in a proper way i.e. methods that consider its continuous behavior, discrete behavior, and their interactions.

In this thesis, we investigate the problem of fault detection and diagnosis and fault

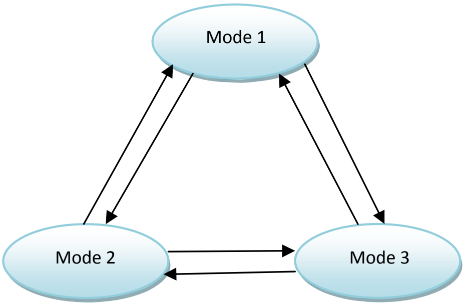

Figure 1.1: A hybrid system

tolerant control of hybrid systems, focusing on active fault diagnosis and passive fault tolerant control of this class of systems.

1.2 Background and State of the Art

1.2.1 Fault and System Behavior

Consider a dynamical system as depicted in Fig. 1.2. It has inputs and outputs, and the relation between the input and the output of the systems is described by some dynamical equations.

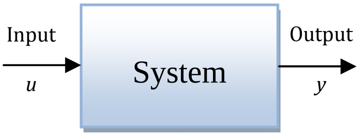

Figure 1.2: A System

A pair which consists of the input and the output of the system at a time instant is called the Input/Output (I/O) pair. For a given system, the set of all possible I/O pairs is called the system behavior. If $\mathcal{U}$ denotes the set of all inputs of the system and $\mathcal{Y}$ the set of all possible output of the system, then the behavior of the system is a set in $\mathcal{U} \times \mathcal{Y}$. This is shown in Fig. 1.3. The set $\mathbf{B}_0$ is the behavior of the systems and the point A is a possible or consistent I/O pair, whereas the point B shows an impossible or inconsistent I/O pair.

A fault, is defined as a change in the parameters or the structure of the systems. As it is shown in Fig. 1.4, three different kinds of faults can be considered for a plant: sensor faults, actuator, and internal faults.

The effect of the fault on the input-output behavior of the system is depicted in Fig. 1.5. The set $\mathbf{B}_0$, represents the normal behavior of the system, and the set $\mathbf{B}_1$, represents the system behavior subject to the fault f_1. The point A is consistent with the normal behavior of the systems and the point B is consistent with the faulty behavior of

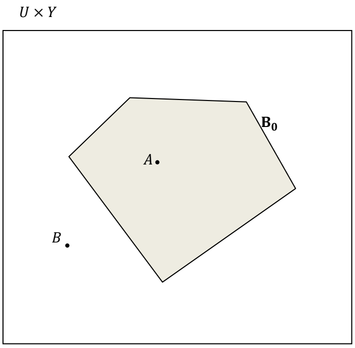

Figure 1.3: System behavior

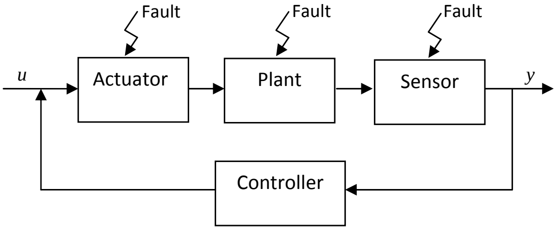

Figure 1.4: Different types of faults

the systems. The point C belongs to an area in which the faulty and the normal behavior of the system overlap.

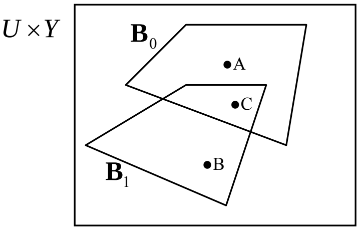

Figure 1.5: System input-output behavior subject to fault

1.2.2 Fault Detection and Diagnosis

Fault diagnosis consists of detection, isolation and identification of the occurred fault. It is a system that receives an I/O sequence from a system subject to faults and checks whether it is consistent with the behaviors of the system. Diagnosis methods can be divided into two main categories: model-free and model-based. A model-based diagnosis method uses a given model of the system to check the consistency of the I/O sequence with the behavior of the system.

From another perspective diagnosis methods can be divided into two classes: passive and active. In Passive Fault Diagnosis (PFD), the system observes the input and output of the system and based on the observation decides if a fault has occurred. In Active Fault Diagnosis (AFD), the diagnoser changes the input to the system and observes the input and output of the system to decide about the occurrence and type of the occurred fault. The change in the input of the systems could be adding a signal to the control input or changing the controller or some part of it which will result in changes in the input to the system.

1.2.2.1 Passive Fault Diagnosis

Fig. 1.6 depicts the structure of a passive fault diagnoser. It is a system that observes the input and output of the plant and based on the observations and by using the consistency principle gives a fault candidate as its output.

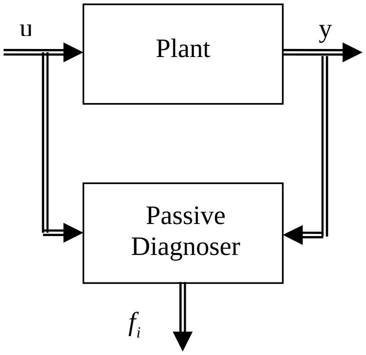

Figure 1.6: Structure of a passive fault diagnoser

PFD state of the art

In the last three decades, a lot of work has been done in the area of FDD, see books [PFC89], [PFC00], [Ise06], [BKLS06]. An excellent review on FDD methods can be found in the three-part review paper: [VRYK03], [VRK03], and [VRKY03]. In a broad view, the FDD methods are divided into two parts: model-based and model-free. Model-based methods use a given model of the system for detection and diagnosis of a fault. This model might be a mathematical description of the systems, quantitative model-based, or it could be a qualitative model of the system such as digraphs or fault trees, see [VRK03]. Model-free methods use a huge amount of data from the system and extract some features from it. These features are later used as a priori knowledge for diagnosis. The feature extraction could be either qualitative or quantitative. We will elaborate more on quantitative model-based methods. For reviews of other methods we refer the interested reader to the review papers [VRK03], [VRKY03], and books [Ise06], and [RCB00].

Quantitative model-based methods can be divided into three main categories :

- state estimation methods
- parameter estimation methods
- parity space methods

State estimation methods use state observers or Kalman filters for state estimation. Then based on the estimated states and system output, a residual signal is generated. The generated residual signal should be insensitive to noise, disturbance and model uncertainties, but sensitive to faults. This can be done by using further available knowledge about the system or by using robust fault detection techniques [FD97], [PC97]. The residual signal should be about zero when the model is fault-free and non-zero when the system is faulty such that a decision can be made based on the value or the pattern of the residual signal about the condition of the system.

The basic structure of an state estimation fault diagnosis is shown in Fig. 1.7. The system input and output are denoted respectively by u and y. The disturbance is denoted by d and $\hat{y}$ is the estimated output of the system. The estimated output is compared with the output of the system to generate the residual signal. A fault is detected when the residual signal is not zero or close to zero. To isolate and identify faults, these methods usually use a bank of state estimators where each one is sensitive to a fault or a set of faults and insensitive to other faults. Therefore, when a fault occurs the corresponding residual which is sensitive to this fault is non-zero i.e. $r_i(t) \neq 0$, while other residuals which are insensitive to this fault are close to zero. The bank of estimators should be designed such that by analyzing the resulting residual signals a complete isolation of faults is possible.

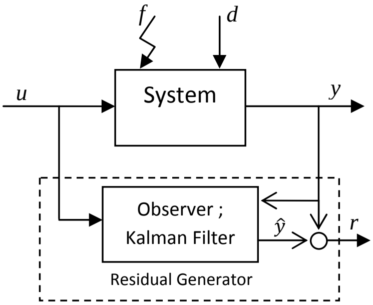

Figure 1.7: Fault diagnosis based on state estimation

Parameter estimation methods uses the input and the output of the system to estimate parameters of the systems. Usually it is assumed that the structure of the system is known a priori. Then, based on the I/O sequence parameters of the system are estimated. Faults are defined as the deviations of the system parameters from their nominal values. The estimated parameters are compared with the nominal values and based on the difference, decisions about occurrence of faults are made [IF91], [Ise93] and [Ise05].

Parity based methods use a transformed version of the state space model to generate parity equations. The value of the parity equation is then used for fault detection and isolation. A parity equation in the fault-free case is zero or close to zero [GS90], [Ger97], [CP99].

Here we review PFD methods for hybrid systems. In a hybrid system two main types of fault might occur: continuous faults: faults that affect the behavior of continuous subsystems, and discrete faults: faults that affect the switching or the transition between continuous subsystems. Since a hybrid system consists of both discrete and continuous systems, to tackle the FDD and FTC problems, researchers have used techniques from continuous systems, discrete systems, or a combination of them.

Many works have used discrete event systems to abstract and analyze hybrid systems behavior [AHLP00]. Discrete event modeling methods used for diagnosis of hybrid systems are finite state automata [Lun00], [FL01], [Lun08] , Petri-nets [ZKH$^+$05], and hybrid bond graphs [Nar02], [NB07].

[Lun00] proposes a discrete event model for diagnosis of quantized systems, i.e. continuous systems whose inputs and outputs are measured quantitatively. The approach does not use temporal information of the events and propose a non-deterministic discrete event model of the system which is suitable for diagnosis. The diagnoser uses consistency principle to decide about occurrence of faults. In other words, it checks the consistency of discrete I/O sequence with the un-timed discrete event model of the system. A necessary and sufficient condition for the model to be suitable for fault diagnosis is given. In [FL01], temporal informations about discrete events are used and therefore a timed discrete event model of the system suitable for fault diagnosis is given. The diagnosis is done using consistency-based diagnosis and a semi-Markov model of the quantized system.

[Lun08] considers the problem of fault diagnosis for a hybrid system which consists of some continuous subsystems where switching between these modes are controlled through a feedback controller. Based on the amount of the information used for abstraction, the paper proposes four models which can be used for diagnosis: embedded maps, semi-Markov processes, timed automata and non-deterministic automata. It is shown that if the obtained model is complete, then the diagnosis results are valid.

[ZKH$^+$05] uses a timed discrete event abstraction of the systems for diagnosis. A fault-symptom table is produced by simulating the hybrid automata of the system with abrupt and incipient faults. A decision tree is built based on the fault-symptom table. These steps are done offline. For online monitoring of the system a timed Petri net monitoring approach is used. Fault detection is based on the consistency of the observed events with its estimation by Petri net. After detecting the fault, the decision tree is used for online diagnosis. [Nar02], [NB07] deal with the problem of parametric faults in hybrid systems using hybrid bond graphs. [DKB09] extend this result such that the diagnoser can handle both parametric and discrete faults.

Other methods are based on continuous time diagnosis techniques. [CEMS04] proposes a method using structured parity relations for both continuous and discrete faults. A fault in a given mode is detected when a parity residual is not zero. They drive sufficient conditions for discernibility of two modes. When all modes are discernible, detection of discrete faults becomes possible.

Many works use state estimation of hybrid systems for FDD of hybrid systems. A discrete fault is usually modeled as a new mode and hence diagnosis of a discrete fault is equal to mode estimation for hybrid systems. [AC01] proposes a method based on a bank of Luenberger observers. The method assumes that the discrete state is known. [BBBSV02] proposes a hybrid observer which consists of a location observer and a continuous observer. The location observer is a finite state machine that observes the discrete input and output of the system and estimate the current location of the system. The continuous observer is a bank of Luenberger observers that receives the continuous input output of the system and the estimate of the current location of the system and estimates the continuous states of the system. A problem with using a bank of observers or filters is the high computational burden.

[HW02], [HW04] consider the problem of mode estimation. By combining hidden Markov models with continuous dynamics they propose a concurrent probabilistic automata framework for modeling of Hybrid systems. A set of possible modes are determined by a hidden Markov observer. A bank of Extended Kalman Filters (EKF) is used for tracking of continuous states of the most likely modes. Since the approach needs only tracking of most likely modes, it is more efficient than using an EKF for each mode. Using a bank of EKF is computationally expensive.

An alternative to using a bank of KF is to use particle filtering approaches because one can control the number of particles which are used for estimation. Moreover, in EKF it is assumed that noise is Gaussian while in particle filters these assumption is not necessary. In [KKZ03], particle filtering is used for state estimation. To improve robustness and efficiency of the algorithm for mode estimation, guard conditions are changed. Boundary of guards are increased by a small constant to prevent chattering in mode estimation.

Authors in [DC01] use particle filtering for fault detection of planetary rovers. The problem with particle filtering for FDD of hybrid systems is what is called sample impoverishment. The probability of transitions to a faulty state is low, hence there is not enough particle in a faulty mode to be considered as the most likely current mode. A remedy to this problem is to increase the number of particles as in [KKZ02], but this is to the cost of increasing computational complexity of the algorithm. A better solution is to use importance sampling [DC01]. [FTMM02] proposes an estimation method based on Mixed-Logical Dynamical (MLD) modeling framework using a moving horizon estimation technique. The problem is formulated as a mixed integer programming. The approach is used for FDD in [BMM99] where faults are introduce as unknown binary variables in the MLD model.

In [WLZL07] a method for state estimation and FDD of hybrid systems with unknown mode transition functions, unknown disturbances and model uncertainties is given. Faults are modeled as discrete modes. The approach consists of a continuous observer and a mode observer where both are composed of a bank of Unknown Input EKF (UIEKF). The UIEKFs of the mode observer run continuously to observe the mode. The continuous observer is a switching system which switches to the UIEKF of the corresponding mode detected by the mode observer. Only when a mode transition is detected, the mode observer is activated again to detect the new mode.

1.2.2.2 Active Fault Diagnosis

An active diagnoser generates a sequence of inputs which excites the system and observes the corresponding outputs. Then based on the observations decides whether a fault has occurred or not and if possible decide, which one has occurred. The structure of an active diagnoser is depicted in Fig. 1.8. It consists of an input generator and a passive diagnoser. The generator generate an input sequence which is applied to the system. Then the diagnoser diagnoses the system by observing the applied input sequence and the corresponding output sequence.

The main advantage of active diagnosis is when different behaviors of the system

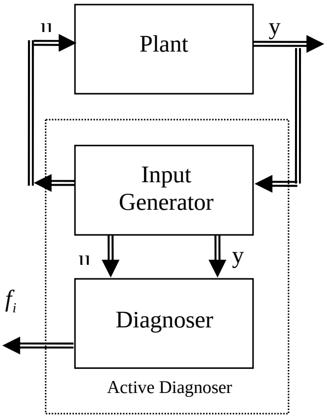

Figure 1.8: Structure of an active fault diagnoser

overlap, e.g. the point C in Fig. 1.5. In these areas, it is impossible to detect the fault by just observing the point. The active diagnoser moves the system from this area by injecting and input signal to the system to an area which uniquely belongs to the faulty or the normal system. ,

Applying the input to the system perturbs the system from its operating point. At one hand, the input should be big enough to make the detection possible and on the other hand it should not move the system from the operating point so much such that it leads to instability or to an unacceptable performance area.

Active diagnosis can be used in the following circumstances: (i) to generate a test signal in the commissioning phase for sanity check of the system. (ii) for faster detection and isolation of faults during the normal operation. (iii) for detection of hidden fault, where because of regulatory actions of the controller, the normal and the faulty behavior of the system exhibit the same behavior.

AFD State of the Art

For AFD, there is a few works for hybrid systems. We first review the literature on linear systems and then on hybrid systems. The problem of using a test signal is studied for a long time is system identification,but generating a test signal for fault detection is recently studied [CN04]. In [Nik98] a method for AFD of linear dynamical systems is introduced. The test signal with a given horizon is designed offline and then applied to the system online. Perturbations are assumed to be bounded in polyhedral sets. The problem of test signal design is formulated as solving a large linear program. Then, it is discussed how to construct a separating hyperplane as a filter for fault detection in real time. Conditions under which separation of polyhedral sets and hence the fault isolation is guaranteed is given.

In [NCD00] a robust method is proposed which is similar to the previous work but the test signal and uncertainties are assumed to be energy bounded. A test signal with minimum energy is designed such that it guarantees fault detection. It is assumed that the test signal is designed offline and is independent of the online measurements from the system. [CHN02] solves the problem for multi-model identification using optimization techniques with the assumption that there is no prior information on the state of the system. It is shown that the method can handle a large number of faults but the answer might not be optimal. But in many cases there is a priori information on the initial state of the system. [NC06] proposes a method for cases when it is known that the initial state is in a given region. This assumption can also be useful for modeling of some important classes of fault such as bias or a jump in the states of the systems. Previous methods could not deal with these faults.

Authors in [CDA$^+$06], extend the result of [CN04] to nonlinear systems. The nonlinear system is linearized and then based on the linearized model, a test signal is found using methods for linear systems. In [ASC08], bounds on nonlinearities are found such that the test signal obtained through linearization is proper for the nonlinear system. Another method is to use direct optimization techniques [ASC08], [And08]. In many application, a piecewise constants test signal is desirable. A method to find an optimal piecewise constant test signal is given in [CCN09] where the problem is cast as a constrained nonlinear optimization problem.

[CN04], [NC06], and [CHN02] model faults as an abrupt change in the system and use a Multi-Model framework for fault detection. [NCD10] considers the case of incipient faults where the fault is a drift in the parameters of the system.

The above approaches consider open-loop systems. In [Nie06], a setup for AFD is introduced that can be used for both open-loop and closed-loop configurations. Fault detection is performed by applying a periodic auxiliary input to the system. It is shown that using the proposed setup, in the nominal case there is no track of the periodic input signal in the residual signal, but in the faulty case a mark of the auxiliary input is present in the residual signal which is used for fault detection. It is shown that the transfer matrix from auxiliary input to the residual is equal to the dual Youla-Jabr-Bongiorno-Kucera (YJKB) transfer function. The method is used in [PN07] and [PN08] to improve the cumulative sum approach for stochastic change detection.

Authors in [SN10] propose two methods for AFD of linear systems by controller reconfiguration without adding an auxiliary signal. The first method deals with additive faults. It uses observer-based control such that the observer part switches periodically between a set of observers. Each observer is designed to be sensitive to a fault or a set of faults. Switching is performed such that the stability of the controlled system is guaranteed. The second method deals with parametric faults. In this approach, a fault is detected by temporarily destabilizing the system.

All of the aforementioned methods are for linear systems. There are few methods proposed for hybrid systems. [BTMO09] proposes a method that abstracts continuous dynamic of the system by discrete events and therefore the behavior of the hybrid system can be described by a hybrid language. When the system is in an ambiguous state, the algorithm looks for a configuration of the system in which the diagnosability properties are satisfied. Then the diagnoser finds a controllable path from the ambiguous state to the new configuration. This is formulated as a conditional planning problem. The diagnoser considers also the safety requirements by avoiding dangerous states. A qualitative event-based method is presented in [DB09]. While [BTMO09] just considers which controllable events should be executed to achieve fault diagnosis, [DB09] considers which controllable events should be blocked or executed.

1.2.3 Fault Tolerant Control

A control system that can tolerate occurrence of faults while maintaining the stability of the overall system and an acceptable degradation in the performance of the system is called a Fault Tolerant Control (FTC) system. In the past two decades, the area of FTC has attracted a significant amount of research, see review papers [BIZBL97], [Pat97], [BFK$^+$00], [BSW01], [Jia05] and books [Ise06] and [BKLS06].

FTC systems can be divided into two classes: Passive (PFTC) and active (AFTC). In AFTC systems, a fault is detected and diagnosed by a fault detection and diagnosis (FDD) scheme. Then the controller is redesigned or reconfigured in the case of severe faults. Control reconfiguration considers the problem of changing the control law or the controller structure by selecting a new set of inputs and outputs. After choosing the new configuration, new control parameters should be found such that the new controller can achieve the original system performance, if it is possible, or at least ensure a tolerable performance degradation in the faulty process, see [BKLS06].

In a PFTC system, the controller does not react to the occurrence of a fault. The structure and the parameters of the controller are designed such that the system can tolerate a set of faults without any change.

1.2.3.1 Active Fault Tolerant Control

A standard control problem can be stated as follows. We should choose a control law among specific class of control laws such as state feedback or output feedback, such that a control objective is met and the constraints on the system dynamics are satisfied [BKLS06]. The control objective is the objective that is to be achieved, for example, it could be the closed loop stability or a performance index which should be minimized.

An AFTC system consists of two main sections: fault detection and identification and controller redesign. The structure of an AFTC system is depicted in Fig. 1.9. The fault diagnosis block receives the input and output sequence from the system, and checks its consistency with the behavior of the system. If the I/O sequence is consistent with the normal behavior of the system, then the system is considered to be working in the normal condition by the FDD block and it will continue working with the nominal controller. If the I/O sequence is not consistent with the nominal behavior of the system, then the FDD block detects occurrence of a fault. Next, the FDD block tries to find out which fault has occurred by checking the consistency of the I/O sequence with the faulty behaviors of the system. The result is a fault candidate f_c. The controller re-design block is informed by the FDD block that the fault f_c has occurred. A new controller, should be designed online or be selected among pre-designed controllers such that the faulty system, can achieve the control objective. If such a controller exist, then the system is fault-tolerant with respect to the fault f_c and the control objective. But if the controller objective can not be achieved, the system is not fault tolerant. In this case, a possible solution is to change the control objective, e.g. by allowing some degradation in the performance of the system or by just considering the stability of the closed loop system.

AFTC state of the art

AFTC methods can be divided into two classes. In the first class, a bank of control laws is pre-computed and when a fault occurs based on a reconfiguration mechanism a control law is used. Among methods that use this approach are multiple-model method, gain

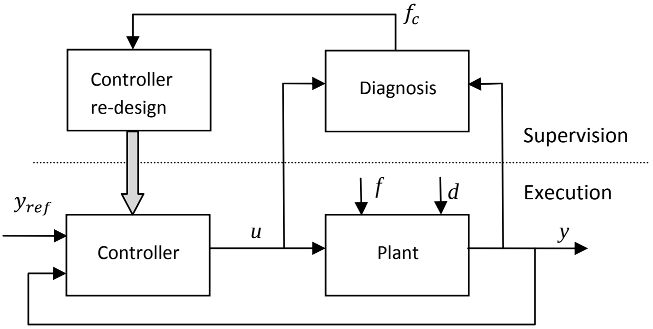

Figure 1.9: System behavior

scheduling and linear parameter varying methods, and general internal model control. In the second class, controller re-design is performed automatically on-line. Pseudo inverse methods, linear quadratic design, model matching, model predictive control are among methods that use this approach [ZJ08]. For a detailed review and classification of these method we refer the interested reader to [ZJ08] and [BKLS06]. Here, we review some methods proposed for hybrid systems.

AFTC for hybrid systems using the MLD framework and optimization techniques is proposed is [TMFTM01]. The problem of how to choose a redundant hardware for a faulty system, is formulated as an MPC problem and solved using mixed integer optimization techniques. [OMP08] proposed a method using MPC and MLD framework. It is discussed how to use the information about faults from the FDD module implicitly, by updating the internal model or dynamic constraints, or explicitly by introducing faults as states of the system. The method is applied to a sewer network.

For switched hybrid systems, [RTS06] proposes a FTC method using static output feedback against actuator faults. Assuming that the FDI provides an exact value of the parameter of the faulty actuator, on-line controller re-design is done such that the stability of the closed loop system with a LMI pole-placement is guaranteed. [YCJ08] proposes a method using passivity. A global passivity concept for switched systems is proposed. A FTC law should provide global passivity of the whole system and not necessarily of each mode. An observer-based method for periodic switched nonlinear systems is proposed in [YJC09].

In [NRZ09a] a method for fault detection, identification and reconfiguration of bimodal PWA systems is proposed. The authors consider actuator faults. Using a Luenberger-observer, fault parameters are estimated and then using informations provided by an observer, a fault tolerant state feedback controller is designed for the faulty system. The input-to-state stability of the overall system is studied and sufficient conditions are derived in terms of LMIs.

[RHvdWL08] develops a reconfigurable control method for PWA systems based on an extension of the idea of virtual actuators and virtual sensors for linear systems. The aim of the reconfiguration is to hide the fault from the controller and at the same time preserve the stability of the reconfigured system. The main feature of this approach is that the nominal controller is not changed but a reconfiguration block is inserted between the plant and the controller to achieve the fault-hiding goal. Sufficient conditions for existence of this reconfigurable controller are provided in terms of LMIs. In [RHvdWL10] the approach is extended such that it can also preserve tracking properties with regard to a constant input reference in presence of a constant disturbance.

1.2.3.2 Passive Fault Tolerant Control

In PFTC, there is no FDD scheme and occurrence of a fault is not detected. The controller is designed off-line and is fixed during the system operation. Therefore, it should be designed such that it can tolerate occurrence of a set of possible faults.

Since a passive fault tolerant controller is a common solution, when too many faults are considered, a solution may not exist or if it exists the performance of the controlled system would be very low. Therefore a passive fault tolerant controller can usually handle a few number of faults.

The advantage of the PFTC scheme can be explained as follows. When a fault occurs, it takes some time for the FDD module to detect the fault and to isolate and identify the fault. There may also be some delay due to the controller re-design. During this period, the system is working with the nominal controller. Performance of the system in this period is mainly dependent on the severity of the fault and the robustness of the nominal controller. It is clear that the controlled system may become unstable in this period, see [ZJ06]. For safety-critical systems, e.g. aircraft flight control or nuclear power plants, when a fault occurs, the time window in which the system remains stabilizable is too small to perform an accurate fault isolation and estimation. In these cases a PFTC system is preferable because it does not need a FDD scheme.

PFTC state of the art

The area of PFTC or reliable control systems has attracted considerable attention in recent years. [Vei95] presents a method for the design of a reliable linear quadratic state feedback control such that it can tolerate actuator outages. The method also provides a guaranteed upper bound on the performance index despite actuator outages. Reliable control using redundant controllers is addressed in [YYLW98]. [YWS01] considers a more general type of faults. Sensor and actuator faults are modeled by scaling factors with upper and lower bounds with a disturbance. The method guarantees the H_∞ performance of the normal systems as well as the faulty system to be less than a bound. [YWSL03] investigate the problem for a class of uncertain linear systems with norm bounded uncertainty. They consider actuator faults which are modeled by scaling factors. The approach provides an upper bound on the quadratic performance despite actuator faults. The problem is solved using LMIs.

Reliable H_∞ control for nonlinear systems using Hamilton-Jacobi inequality approach is presented in [YLW98]. In this paper, only actuator outage is considered. The authors in [YWS00], also consider the partial degradation of actuators. This approach provides an upper bound on the H_2 performance in presence of faults.

Among different classes of hybrid systems, PFTC is mainly studied for switched systems and piecewise linear systems. [WLZ07] proposes a method for a class of switched nonlinear systems. A sufficient condition for the controlled system with actuator failures to be stable with a H_∞ norm bound are derived in terms of partial differential inequalities which are very hard to solve.

PFTC for PWL continuous time systems using state feedback is presented in [NRZ09b].

The approach uses common Lyapunov functions. A common Lyapunov function may not always exist.

Another approach to address the PFTC problem is to use robust control techniques. In this case faults are modeled as uncertainties and a robust control is designed such that it can tolerate uncertainties and provide a guaranteed upper bound on a performance criterion. Robust control methods are also useful for AFTC. Because when a fault is detected and diagnosed, usually the parameters of the faulty system are not known exactly but are known with a bounded uncertainty.

[Fen02] studies robust control of uncertain PWL systems using state feedback and continuous piecewise Lyapunov functions. It is shown that the controller can be designed by solving a set of LMIs. [ZT08] propose a method for robust H_∞ output feedback control design for uncertain piecewise affine systems. The method uses Bilinear Matrix Inequalities to solve the problem. In [ZT09], a guaranteed cost control method using output feedback is proposed. The problem is reformulated as the feasibility of a set of BMIs. The non-convex optimization problem is solved using a method that combines genetic algorithms and semi definite programming. Both works, assume that switching of the controller is based on the real state of the system and not based on the estimated state of the system. In other words, the plant and the controller are always in the same region. This is not a realistic assumption. All of the aforementioned works are in the continuous time domain.

In the discrete time domain, the problem of robust stability of autonomous piecewise affine systems is studied in [Kan97], but the case of controller design is not addressed. [GLC08] propose a robust H_∞ control approach for uncertain discrete time piecewise affine systems. They consider time varying parameter uncertainties. The approach uses state feedback and formulates the problem as LMIs.

1.3 Outline of the Thesis

The reminder of this thesis is organized as follows. In the next chapter, we describe the methodology and preliminary definitions and theories. We will start with a definition of hybrid systems and then different classes of hybrid system such as Hybrid automaton, mixed logical dynamical systems and piecewise affine systems are described. In particular, some basic results on stability of PWA systems as well as model predictive methods for MLD systems are described. A summary of contributions of this PhD thesis will be given in Chapter 3. Finally, in chapter 4, conclusions and possible future works and research are discussed. In the addendum, contributions of the thesis are added which are listed in the following:

- Paper A[TRIZB09] In this paper a method for AFD of Hybrid system is presented. The method is based on reach set computation for the normal and faulty system. The method finds the shortest input sequence to detect the fault.

- Paper B[TIZBR09]: In this paper the method of [TRIZB09] is used for the problem of automatic sensor assignment during the commissioning phase. The method is tested on a supermarket refrigeration system.

- Paper C[TIZRB10]: In AFD we are perturbing the system from its operating point and therefore it is important to make sure that the system remains stabilizable.

In this paper a stabilizable AFD method for hybrid systems is presented. Using the MLD framework, the problem is formulated as a mixed integer programming problem. To ensure stabilizability of the system despite excitation, constraints are imposed on the optimization problem to make sure that there exists a stabilizing MPC for the normal and faulty models of the system.

- Paper D[TRIZB10]: This paper also deals with the problem of stability in AFD. In this paper, distinguishable steady outputs are used for AFD. The diagnoser looks for steady outputs of different models of the system such that they are distinguishable from each other. Then the system is excited to reach this steady output. Fault diagnosis is done by observing the steady outputs.

- Paper E[TIZBR10]: In this paper we deal with the problem of FTC for piecewise affine systems. PFTC design for PWL system using PWQ Lyapunov function is formulated in terms of LMIs. The method uses piecewise linear state feedback and provides us with an upper bound on a given performance index. Optimal upper bound can be found by solving a convex optimization problem with LMI constraints.

- Paper F[TBIZ10]: Since in many applications states of a system are not available, in this paper we extend the result of Paper E to output feedback control. This time the problem is formulated in terms of BMIs and the optimal upper bound on the performance cost is found by solving and optimization problem with BMI constraints.

2 | Methodology

In this chapter, the fault diagnosis and fault tolerant control problem are formulated and then an overview of different classes of hybrid systems and an equivalence result between these classes are given.

2.1 Fault diagnosis

Consider a dynamical system described by:

$$\mathcal{G}_0 : \begin{cases} x(k+1) = f(x(k), u(k)), \;\; x(0) = x_0, \\ y(k) = h(x(k), u(k)), \end{cases} \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input and $y \in \mathbb{R}^p$ is the output of the system. The pair $(u(k), y(k))$ which consists of the input and output of the system at time k is an I/O pair. A fault f is a change in the parameters or the the structure of the f or g. We assume the system is working in the nominal condition during the time period $[0, k_f[$. At time k_f, the fault f occurs. The dynamic of the system subject to the fault f, is described by:

$$\mathcal{G}_f : \begin{cases} x(k+1) = f_f(x(k), u(k)), \;\; x(k_f) = x_f, \\ y(k) = h_f(x(k), u(k)), \end{cases} \tag{2.2}$$

Input-output behavior of a system is the set of all possible input output of the system. We show the behavior of the normal system with $\mathcal{B}_0$ and the behavior of the system subject to faults $f_1, \cdots, f_n$ by $\mathcal{B}_1, \cdots, \mathcal{B}_n$.

The aim of fault diagnosis is to detect if a fault has occurred and if it has occurred to determine which fault is occurred. A fault diagnoser is a system that receives an I/O sequence from a system and tests the consistency of it with the behaviors of the system. The input and output sequence are respectively shown by: $U = \langle u(0), \ldots, u(T_d) \rangle$ and $Y = \langle y(0), \ldots, y(T_d) \rangle$. The diagnosis problem is stated as:

Problem 2.1 (Diagnosis problem)**.** *: Given the set $\mathcal{B} = \{\mathcal{B}_0, \cdots, \mathcal{B}_0\}$ describing behavior of the system with no faults and subject to faults $\{f_1, \cdots, f_n\}$, and the I/O sequence(U, Y), find a fault candidate f_c.*

What is stated above is the passive diagnosis problem where the fault is diagnosed by observing the system. In active diagnosis the the diagnoser generates an input sequence U, applies it to the system and then based on the output sequence, Y, makes a decision about the condition of the system. The active diagnosis problem can be stated as follows:

Problem 2.2 (Active diagnosis problem)**.** *Given the set $\mathcal{B} = \{\mathcal{B}_0, \ldots, \mathcal{B}_n\}$ describing behaviors of the system with no fault and subject to faults $\{f_1, \ldots, f_n\}$, find a sequence of inputs U such that (U, Y) belongs only to a unique $\mathcal{B}_i$.*

2.2 Fault Tolerant Control

A standard control problem can be shown by the following triple:

$$\langle \mathcal{O}, \mathcal{G}, \mathcal{K} \rangle \tag{2.3}$$

A control law $\mathbf{K}$ among a given set $\mathcal{K}$ is chosen such that the control objective $\mathcal{O}$ is met and the constraints on the system dynamics, $\mathcal{G}$ are satisfied [BKLS06].

In AFTC, a controller is designed for the nominal system by solving the control problem for the nominal system. The controller is changed when a fault is diagnosed. The FDD block observes the input and output of the system to see if any fault has occurred. The system works with the nominal controller as long as no fault is detected by the FDD block. When the FDD block detects occurrence of a fault, it tries to find out a fault candidate, f_c, for the fault that has occurred by checking the consistency of the I/O sequence with the faulty behaviors of the system. The FDD block informs the controller re-design block that the fault f_c has occurred. A new controller $\mathbf{K}_{rf}$, should be designed online or be selected among pre-designed controllers such that the faulty system, $\mathcal{G}_{f_c}$, can achieve the control objective $\mathcal{O}$. In other words, the following control problem should be solved:

$$\langle \mathcal{O}, \mathcal{G}_{f_c}, \mathcal{K}_f \rangle \tag{2.4}$$

The system is fault-tolerant with respect to the fault f_c and the control objective $\mathcal{O}$, if there exist a solution to the above problem. If the system is not fault-tolerant, a possible solution is to change the control objective, e.g. by allowing some degradation in the performance of the system or by just considering the stability of the closed loop system.

In PFTC, the controller is designed off-line such that it can tolerate occurrence of a set of possible faults and is fixed during the system operation. In other words, a passive fault tolerant controller, $\mathbf{K}_{PFTC}$ is a common solution to the following control problems.

$$\begin{cases} \langle \mathcal{O}, \mathcal{G}_0, \mathcal{K}_0 \rangle \\ \qquad \vdots \\ \langle \mathcal{O}, \mathcal{G}_n, \mathcal{K}_n \rangle \end{cases} \tag{2.5}$$

Since $\mathbf{K}_{PFTC}$ is a common solution to more than one control problem, it can usually handle a few number of faults. Because when too many faults are considered, a solution may not exist or if it exists the performance of the controlled system would be very low.

The advantage of PFTC scheme can be explained as follows. Assume a fault has occurred at k_f. It takes some time for the FDD module to detect, isolate and identify the fault. We assume that at k_D the fault is detected isolated and identified. There is also some delay due to controller re-design. Let k_{rf} be the time at which the new control law is implemented. During the period $[k_f, k_{rf}]$ the system is working with the nominal controller. Performance of the system in this period depends on the severity of the fault and the robustness of the nominal controller. If the time window in which the system remains stabilizable after occurrence of a fault is too small to isolate and estimate the fault accurately, a PFTC is preferred because it does not need a FDD scheme. In practice, usually both methods are used. Non-severe faults are handled with PFTC and severe faults with AFTC.

2.3 Hybrid systems

In control and systems theory, systems are traditionally divided into two categories: continuous or discrete. A continuous system is a system whose states take values in $\mathbb{R}^n$. A discrete system is a system whose states take discrete values, for example from the set of $\{ON, OFF\}$. These two classes of systems have been studied separately in control and systems theory and computer science. But many systems contain both behaviors, they contain states that take both continuous values and discrete values and evolution of these states are not independent but there is a non-trivial interaction between rules that govern evolutions of these states. Hybrid systems is a class of systems that is introduced to capture behaviors of this types of systems. A hybrid system consists of several modes as depicted in Fig. 2.1. The behavior of the system in each mode is described by a differential equation or a difference equation. Transition between modes may happen if the state of a system enter an area or pass a threshold or if a period of time is passed or by an external input. Hybrid systems arise in many applications such as mechanical systems, electrical circuits with switching components, chemical process, or embedded systems.

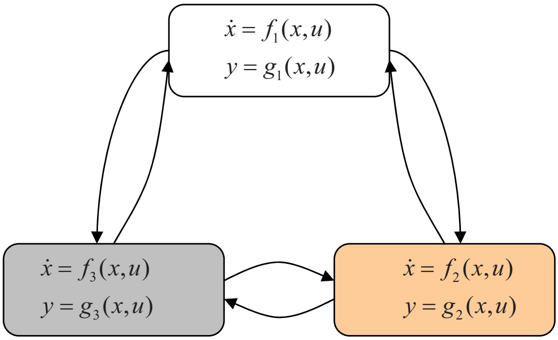

Figure 2.1: A Hybrid systems with 3 modes of behavior

There are many different modeling frameworks proposed for modeling of hybrid systems and each one has its own advantages and drawbacks. Here we introduce Hybrid automata, Mixed Logical Dynamical Systems (MLD), and PieceWise Affine (PWA) systems.

2.3.1 Hybrid Automaton

A Hybrid automaton can be briefly described as follows. It is a graph whose vertices represent discrete modes, q_i, and in each mode the dynamic of the system is characterized by a dynamical system with vector field f and the output map h. An edge between two vertices represents a transition between the corresponding discrete modes. The discrete dynamic and continuous dynamic interact with each other through guards and transition relations. Each discrete mode, q, has an invariant, $Inv(q)$, which characterizes the conditions that the continuous system must satisfy to stay in this mode. If a continuous state hits a guard then based on the transition relation the system jumps to a new mode. The continuous state in the new mode is determined bases on a jump function. We introduce the following description of a hybrid automaton [LSVW96].

Definition 2.1 (Hybrid Automaton)**.** A *hybrid automaton,* $\mathcal{H}$ is a collection $\mathcal{H} = (Q, X, U, Y, Init, f, h, Inv, E, G, J)$ where,

- Q is a set of finite discrete modes, $Q = \{q_1, q_2, \dots, q_m\}$,
- X is a finite set of continuous state variables,
- U is a finite collection of input variables,
- Y is a finite collection of output variables,
- $Init \subset Q \times X$ is a set of initial states,
- $f : Q \times X \times U \to \mathbb{R}^n$ is a vector field,
- $h : Q \times X \times U \to Y$ is an output map,
- $Inv : Q \to 2^{X \times U}$ assigns to each $q \in Q$ an invariant set $Inv(q) \subseteq X \times U$,
- $E \subset Q \times Q$ is a set of discrete transitions,
- $G : E \to 2^{X \times U}$ assigns to each $e = (q, q') \in E$ a guard $g(e) \subset X \times U$,
- $J : E \times X \times U \to 2^X$ is a jump function that assigns a jump set $J(e, x, u) \subseteq X \times U$ to each pair $e \in E$ and $x \in g(e)$.

The initial state, (q_0, x_0) of a hybrid automaton is in the set $Init$. The continuous state of the systems evolves based on the relation f and the input u. $x(k)$ stays in mode q_0 as long as it satisfies the conditions in $Inv(q_0)$. When the continuous state hits the guard $g(q_0, q')$, the transition $e = (q_0, q')$ is enabled and the discrete state of the system switches from q_0 to q' and the continuous state of the system jumps from $x(k)$ to $x(k+1) = J(e, x(k), u)$. From there, the continuous state evolves and the evolution of the discrete mode has the same procedure as before.

In the case of linear hybrid systems the vector field f_q is represented by a linear difference equation: $x(k+1) = A_{q(k)}x(k) + B_{q(k)}u(k)$ and and the output map is of the form $y(k) = C_{q(k)}x(k) + D_{q(k)}u(k)$.

The tuple $(q, x, u, y) \in Q \times X \times U \times Y$ is called a point of $\mathcal{H}$, $(q, x) \in Q \times X$ is called the state of $\mathcal{H}$, $u \in U$ is the input and $y \in Y$ is the output of $\mathcal{H}$. Also we refer to $(u, y) \in U \times Y$ as an observation of $\mathcal{H}$.

Definition 2.2 (Execution)**.** An execution of a hybrid automaton is a sequence $\chi = (\sigma(0), \dots, \sigma(k), \sigma(k+1), \dots)$ where $\sigma(0) = (q_0), x_0, u(0), y(0))$, $\sigma(k) = (q(k), x(k), u(k), y(k))$ such that:

- Initial condition $(q_0, x_0) \in Init$,

- Continuous evolution: for all k, $q(k) = q(k+1)$, $(x(k+1), u(k+1)) \in Inv(q(k))$:

$$x(k+1) = A_{q(k)}x(k) + B_{q(k)}u(k)$$

$$y(k+1) = C_{q(k)}x(k) + D_{q(k)}u(k)$$

- Transition: for all i, $e = (q(k), q(k+1)) \in E$, $(x(k), u(k)) \in G(e) : x(k+1) \in J(e, x(k), u(k))$, $(x(k+1), u(k+1)) \in Inv(q(k+1))$

A graphical representation of a hybrid automaton with four discrete modes is depicted if Fig. 2.2.

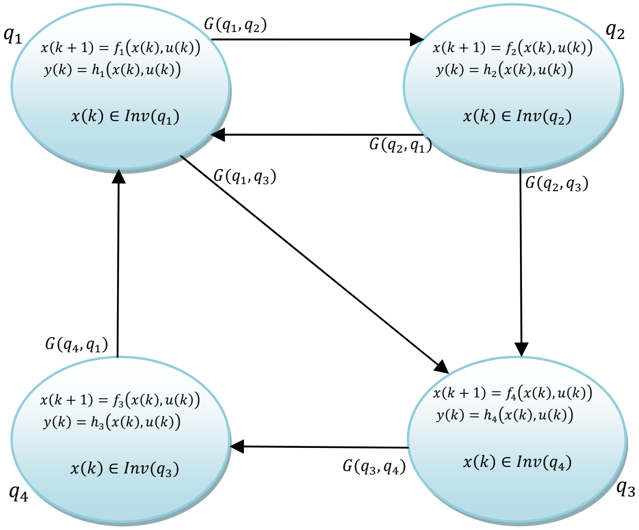

Figure 2.2: Graphical representation of a Hybrid automaton

2.4 Mixed Logical Dynamical System

Mixed Logical Dynamical (MLD) systems are proposed for modeling of systems that contains interaction between 'physical laws and logical rules and operating constraints', [BM99]. Borrowing techniques from propositional calculus, the logical rules and operating constraints of the system are translated into linear inequalities with both continuous and integer variables. MLD systems are capable of modeling many classes of systems such as linear hybrid systems, piecewise affine systems, constrained linear systems, and some classes of discrete event systems.

In the following the procedure for modeling a system in MLD framework is sketched briefly. The main idea is to transform the logical rules of the systems into mixed-integer inequalities such that the overall dynamic of the system can be described by a set of difference or differential equations which contain real and binary states and input and a set of mixed-integer inequalities.

A boolean variable X in the logical rules of the system is associated with a binary variable δ:

$$X = True \Leftrightarrow \delta = 1, \tag{2.6}$$

$$X = False \Leftrightarrow \delta = 0. \tag{2.7}$$

Then basic logical propositions are transformed into linear integer inequalities. As an example the boolean expression

$$X_1 \vee X_2,$$

is first represented by

$$[\delta_1 = 1] \vee [\delta_2 = 1],$$

and is then transformed to the following integer equality:

$$\delta_1 + \delta_2 \geq 1.$$

Or the relation $X_3 \to (X_1 \wedge X_2)$ is represented by $[\delta_3 = 1] \to [\delta_1 = 1] \vee [\delta_2 = 1]$ which is equal to the following integer inequalities:

$$\begin{cases} \delta_1 - \delta_3 \geq 0 \\ \delta_2 - \delta_3 \geq 0 \\ -\delta_1 - \delta_2 + \delta_3 \geq 1 \end{cases} \tag{2.8}$$

A detailed list for basic logical propositions is given in [Mig02]. The dynamics of continuous systems are expressed as before using difference equations.

There are some relations which contain both logical propositions, i.e. implication or If-then-else rules, and continuous variables or dynamical relations. These relations are translated to mixed integer inequalities. For example consider the relation:

$$IF\ X\ THEN\ z = f(X)\ ELSE\ z = 0,$$

which is equal to the logic proposition:

$$z = \delta f(x).$$

This is equivalent to the following set of mixed integer inequalities:

$$\begin{cases} z \geq M\delta, \\ -z \geq -m\delta, \\ z \geq f(x) - m(1-\delta), \\ -z \geq -f(x) + M(1+\delta), \end{cases} \tag{2.9}$$

where M and m are respectively the upper and lower bound of the function f on a bounded set $\mathcal{X}$:

$$M \triangleq \max_{x \in \mathcal{X}} f(x) \tag{2.10}$$

$$m \triangleq \min_{x \in \mathcal{X}} f(x) \tag{2.11}$$

Using this technique we can describe continuous dynamics, logical rules, operating constraint and interactions between them by a set of difference equations containing real and integer states, real and integer inputs, and auxiliary real and integer variables and a set of mixed-integer inequalities. The equations describing an MLD system are as follows:

$$x(t+1) = Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \tag{2.12}$$

$$y(t) = Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) \tag{2.13}$$

$$E_2\delta(t) + E_3z(t) \leq E_1u(t) + E_4z(t) + E_5 \tag{2.14}$$

where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_l}$ are states, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}$ are the inputs, $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_l}$ are the outputs. $\delta \in \{0,1\}^{r_l}$ and $z \in \mathbb{R}^{r_c}$ are auxiliary binary and continuous variables.

A trajectory of MLD system, starting from initial state $x(t_0) = x_0$, when the input sequence $\{u\}_{t_0}^{t-1} = \{u(t_0), u(t_0+1), \cdots, u(t-1))\}$ is applied to the system, is denoted by $x(t, t_0, x_0, \{u\}_{t_0}^{t-1})$. It is assumed that the system in (2.12)-(2.14) is completely well posed, see [BM99], which means given an initial state $x(t_0)$ and an input sequence $\{u\}_{t_0}^{t-1}$, the trajectory $x(t, t_0, x_0, \{u\}_{t_0}^{t-1})$ is unique. This requires the inequalities (2.14) to have a unique solution for $\delta(k)$ and $z(k)$ for a given state and input $x(k)$ and $u(k)$.

An equilibrium state of an MLD system is defined as follows.

Definition 2.3 (Equilibrium state). $x_e \in \mathbb{R}^{n_c} \times \{0,1\}^{n_l}$ is an equilibrium state of the MLD system (2.12)-(2.14) with input $u_e \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}$ if $x(t, t_0, x_e, u_e) = x_e \; \forall t \geq t_0, \forall t_0 \in \mathbb{Z}$. The corresponding output y_e is called the equilibrium output and the pair (x_e, u_e) is called the equilibrium pair.

The MLD framework is capable of modeling various classes of hybrid systems such as PieceWise Affine (PWA) systems, linear systems with piecewise linear output functions, linear systems with discrete inputs or with qualitative outputs, bilinear systems, and finite state machines in which an LTI system generates the events, see[BM99].

Equivalence of MLD systems with other classes of hybrid systems such as PWA systems, linear complementary (LC) systems, extended linear complementary (ELC) systems, and max-min-plus-scaling (MMPS) systems under some assumptions is shown in [HSB01].

Using the MLD framework, different problems such as optimal control, state estimation, etc. can be reformulated as mixed-integer programing problems and be solved using mixed integer programming techniques. To translate a description of a hybrid system into mixed integer equalities and inequalities (2.12)-(2.14) a modeling language called HYSDEL(HYbrid System DEscription Language) is proposed in [TB04]. HYSDEL receives a textual description of the hybrid systems as input and then translates it automatically to different classes of hybrid systems, in particular it returns a model of the system in PWA and MLD form.

2.4.1 Model Predictive Control of MLD systems

Model Predictive Control (MPC) is an attractive control method because of its capability to deal with constraints and to deal with multi-variable systems [MRRS00]. MPC consists of solving an optimal control problem over a finite horizon repeatedly. At each time step, given the current state of the system, an optimal control problem over a finite horizon is solved. The optimal input sequence is found and only the first element of the sequence is applied to the system. At the next time step, based on the new measurements from the system, a new optimal control problem is solved and the same procedure is repeated. The structure of a MPC is depicted in Fig. 2.3.

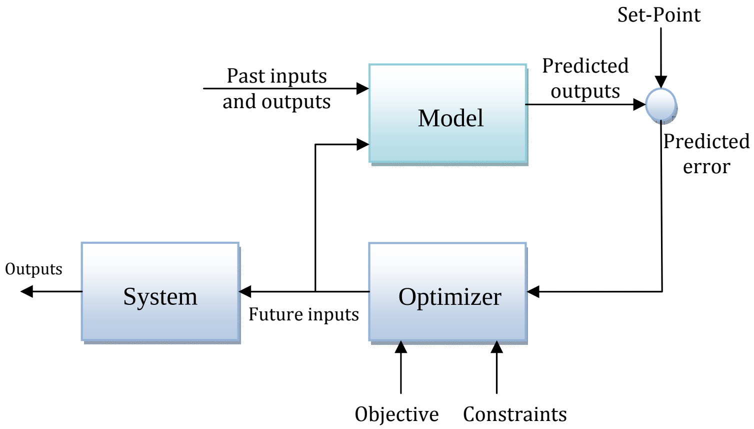

Figure 2.3: Structure of a Model Predictive Controller

The optimal control problem is computed over a short horizon and it can include constraints on states, inputs, or outputs. At each time step, a constrained optimization problem is solved. The type of the optimization problem depends on the class of the given model, the performance function to be minimized, and the type of constraints. For example, for a linear system with linear constraints on states and inputs, it is usual to choose a linear or a quadratic performance function and therefore the resulting optimization problem is a Linear Programming (LP) or a Quadratic Programming (QP) problem. The length of the prediction horizon plays an important role in MPC. Because at one hand the size of the optimization problem, and hence the computational complexity, depends crucially on the prediction horizon length: the smaller the prediction horizon, the lower the computational complexity. At the other hand, the performance of the controlled system, specially the stability of the closed loop system, depends on it. A longer prediction horizon results in a better performance and may be required to guarantee stability.

Here, we formulate the MPC for MLD systems. In this case the resulting optimization problem is an MILP or MIQP. Consider the MLD system (2.12)-(2.14) with constraints on input and states:

$$x(k) \in \mathbb{X} \times \{0,1\}^{n_l}, \tag{2.15}$$

$$u(k) \in \mathbb{U} \times \{0,1\}^{m_l}, \tag{2.16}$$

where $\mathbb{X} \subseteq \mathbb{R}^{n_c}$ and $\mathbb{U} \subseteq \mathbb{R}^{m_c}$ are compact polyhedral sets that contain the equilibrium pair (x_{c_e}, u_{c_e}) in their interior.

Define $x(k|t) \triangleq x(t+k, t, x(t), \{u\}_t^{k-1})$ and let $\delta(k|t), z(k|t), y(k|t)$ be similarly defined . The sequence generated from the initial state $x(k|k) = x(k)$ by applying the input sequence $\{u\}_k^{k+T-1} \triangleq \{u(k+1|k, \cdots, u(k+T-1|k))\}$ is denoted by

$$\mathbf{x}_k(x(k), \{u\}_k^{k+T-1}) \triangleq \{x(k+1|k), \cdots, x(k+T|k)\} \tag{2.17}$$

Assuming the equilibrium pair (x_e, u_e) as the desired target point, then

$$\mathcal{U}_T(x(k)) \triangleq \{(u) \in \mathbb{U}^T \times \{0,1\}^{Tm_l} | \mathbf{x}_k(x(k), \{u\}_k^{k+T-1}) \in \mathbb{X}^T \times \{0,1\}^{Tn_l}, x(T|k) = x_e\} \tag{2.18}$$

is the class of admissible input sequences with respect to x_e and $x(k)$. The cost function $J(x(k), \mathbf{u}_k)$ is defined as:

$$J(x(k), \mathbf{u}_k) \triangleq \Sigma_{k=0}^{T-1} \|Q_1(y(k|t) - y_e)\|_p + \|Q_2(x(k|t) - x_e)\|_p + \|Q_3(u(k|t) - u_e)\|_p + \|Q_4(\delta(k|t) - \delta_e)\|_p + \|Q_5(z(k|t) - z_e)\|_p, \tag{2.19}$$

where T is the prediction horizon, and $\|Qx\|_p = x^T Qx$ for $p = 2$ and $\|Qx\|_p = \|Qx\|_\infty$ for $p = \infty$, and Q_1, Q_2, Q_3, Q_4, Q_5 are symmetric positive definitive matrices for $p = 2$ and nonsingular matrices if $p = \infty$.

At time instance t, for a given $x(t)$, the optimal MPC minimizes, the objective function J subject to the dynamic constraints of the systems and input and state constraints:

$$\begin{cases} x(T|t) = x_e \\ x(t|t) = x(t) \\ x(k+1|t) = Ax(k|t) + B_1u(k) + B_2\delta(k|t) + B_3z(k|t) \\ y(k|t) = Cx(k|t) + D_1u(k) + D_2\delta(k|t) + D_3z(k|t) \\ E_2\delta(k|t) + E_3z(k|t) \leq E_1u(t) + E_4z(k|t) + E_5 \\ x(k) \in \mathbb{X} \times \{0,1\}^{n_l} \\ u(k) \in \mathbb{U} \times \{0,1\}^{m_l} \end{cases} \tag{2.20}$$

It is assume that there exists an optimal sequence for this problem, which is denoted by:

$$\mathbf{u}_k^* \triangleq \{u^*(k|k), \cdots, u^*(k+N-1|k))\} \tag{2.21}$$

The MPC control law is defined as the first element of this sequence:

$$u^{MPC}x(k) \triangleq u^*(k|k) \tag{2.22}$$

The input is applied to the system and the whole procedure is repeated at the next time instance.

In (2.20), the constraint $x(T|t) = x_e$ is imposed to guarantee the stability of the closed loop systems and is called the stability constraint. $x(k) \in \mathbb{X} \times \{0,1\}^{n_l}, u(k) \in \mathbb{U} \times \{0,1\}^{m_l}$ are state and input constraints.

The set of states for which the constraints (2.20) are feasible is called the feasible set which is defined as follows.

Definition 2.4 (Feasible set)**.** The feasible set $\mathbb{X}_F(T)$ is defined as

$$\mathbb{X}_F(T) = \{x \in \mathbb{X} \times \{0,1\}^{n_l} | \mathcal{U}_T(x) \neq \emptyset\} \tag{2.23}$$

The following theorem shows that in the MPC problem, feasibility is preserved over time and that feasibility implies stability.

Theorem 2.1. *Assume that (x_e, u_e) is an equilibrium pair. Fix $T \in \mathbb{Z}_{\geq 1}$. If the optimization problem (3.2) with constraints (2.20) is feasible for $x(t)$ at time t, then it is feasible at time $t+1$ for state $x(t+1)$ which is evolving based on the MLD system equations in (2.12)-(2.14) with input $u^{MPC}x(k)$. Moreover, the MPC law (2.22) stabilizes the system.*

Proof. For the proof of the theorem we refer the interested reader to [BM99]. □

The optimization problem that should be solved at each time is a MILP for $p = \infty$ or a MIQP problem for $p = 2$. There are many efficient solvers available for solving MILP or MIQP problems, but it should be pointed out that the worst-case computational complexity of these problems grows exponentially as the number of integer variable increases. Hence the application of this method is restricted to systems with small size or system with slow dynamics. However, a suboptimal solution also guarantees the stability of the system and therefore for a real time application we can stop the solver when a suboptimal solution is available.

Available methods for solving a Mixed Integer Program (MIP) are:

- Cutting and plane method
- Decomposition methods
- Logic-based methods
- Branch and Bound methods

See [RH05] and references therein for a review of these methods. There are commercial packages available such as CPLEX [Cpl03] as well as some noncommercial packages, for a review see [LR05].

2.5 Piecewise Affine systems

Piecewise affine systems approximate nonlinear system efficiently, and they arise in any practical system that contains PWA components such as dead-zones, saturation, hysteresis, relays, etc. In a PWA system the state space is partitioned into polyhedral regions and in each region the dynamic of the system is described by a difference equation or a differential equation. Switching between systems is based on crossing a guard.

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) + f_i \quad for \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \mathcal{X}_i \qquad (2.24) \\ y(k) &= C_i x(k) + D_i u(k) + g_i, \end{aligned}$$

where $x(k) \in \mathbb{R}^n$ is the state, $u(k) \in \mathbb{R}^m$ is the control input, and $y(k) \in \mathbb{R}^p$ is the output. f_i and g_i are affine terms. When the affine terms are zero the system is a PieceWise Linear (PWL) system. $\{\mathcal{X}_i\}_{i=1}^{s} \subseteq \mathbb{R}^n$ denotes a partition of the state into a number of polyhedral regions $\mathcal{X}_i, i \in \mathcal{I} = \{1, \cdots, s\}$. Each polyhedral region is represented by:

$$\mathcal{X}_i = \{ \begin{bmatrix} x \\ u \end{bmatrix} | \mathcal{H}_i x + \mathcal{F}_i u \leq h_i \}, \qquad (2.25)$$

We assume that there is a unique solution for the PWA system (2.24). Therefore the partition of the input-state space should not have overlapping regions, *i.e.* $\mathcal{X}_i \bigcap \mathcal{X}_j = 0$ for $i \neq j$. It should be noted that when the PWA system is continuous over a facet, then the regions who share this facet are both defined on the facet and still the solution of the system is uniquely defined.

2.5.1 Stability and Stabilization of PWA systems

In this section we explain results on the stability of PWA systems and state-feedback stabilization of discrete time PWA systems using Piecewise Quadratic (PWQ) Lyapunov functions.

Checking stability of a PWA system is not an easy task. Results on the stability of PWA systems usually consider a special class of PWA systems or give sufficient conditions. Many approaches use PWQ Lyapunov functions to provide sufficient conditions for stability of PWA or PWL systems which are formulated in terms of LMIs, see [JR98] for continuous time and [MFTM00], and [CM01] for discrete time.

Definition 2.5. The origin $x = 0$ is stable for a system of the form $x(k+1) = f(x(k))$, if for any $\epsilon > 0$ there exist a $\delta > 0$, such that if $\|x(0)\| < \delta$ then for all $k > 0$ we have $\|x(k)\| < \epsilon$. Moreover, the origin is asymptotically stable if $lim_{k\rightarrow\infty} \|x(k)\| = 0$

Theorem 2.2. *The origin $x = 0$ of the PWA system (2.24) is asymptotically stable if there exist a positive definite function $V(x(k))$ such that $V(x(k+1)) - V(x(k)) < 0, \forall x(k) \in \mathcal{X}_i, \forall i \in \mathcal{I}$.*

2.5.1.1 Quadratic Lyapunov function

A quadratic Lyapunov function for a PWA function is defined as:

$$V(x(k)) = x(k)^T P x(k), \tag{2.26}$$

where P is a positive definite matrix of appropriate dimension. Note that by switching of the system for $\mathcal{X}_i$ to $\mathcal{X}_j$, the Lyapunov function remains the same. In other words it is a common Lyapunov function between all regions. The condition for stability of a PWA system is derived by requiring the difference of Lyapunov function along every trajectory to be negative:

$$x(k+1)^T P x(k+1) - x(k)^T P x(k) < 0.$$

Assuming that $f_i = 0$, then the following LMI conditions for asyptotic stability are derived:

$$P > 0, \tag{2.27}$$

$$A_i^T P A_i - P < 0. \quad \forall i \in \mathcal{I} \tag{2.28}$$

The problem of state feedback control that we consider is that of designing a piecewise linear state feedback of the form:

$$u(k) = K_i x(k) \ for \ x(k) \in \mathcal{X}_i \tag{2.29}$$

such that the closed loop system

$$x(k+1) = \mathcal{A}_i x(k), \tag{2.30}$$

is stable where $\mathcal{A}_i = A_i + B_i K_i$ is exponentially stable. In this case the conditions for the closed loop stability are:

$$P > 0 \tag{2.31}$$
$$(A_i + B_i K_i)P(A_i + B_i K_i) - P < 0 \;\; \forall i \in \mathcal{I} \tag{2.32}$$

Since both P and K_i are unknown, the above inequality is nonlinear, but it can be transformed to LMIs using Schur complement.

Lemma 2.1. *The PWA system (2.24) with piecewise linear state feedback (2.29) is asymptotically stable if there exist $X = X^T > 0$ and matrices Y_i such that:*

$$\begin{bmatrix} X & (A_i X + B_i Y_i) \\ (A_i X + B_i Y_i) & X \end{bmatrix} < 0 \;\; \forall i \in \mathcal{I} \tag{2.33}$$

Then the piecewise linear state feedback gains are given by:

$$K_i = Y_i Q^{-1}. \tag{2.34}$$

2.5.1.2 Piecewise Quadratic Lyapunov Function

A PWQ Lyapunov function is defined as:

$$V(x(k)) = x(k)^T P_i x(k) \;\; if \; x(k) \in \mathcal{X}_i \tag{2.35}$$

In the case of discrete time systems, contrary to the continuous time case, the Lyapunov function need not to be continuous across facets. To have stability we must have:

$$V(x(k)) - V(x(k+1)) < 0. \tag{2.36}$$

The general case is when $x(k) \in \mathcal{X}_j$ and $x(k+1) \in \mathcal{X}_i$, then the conditions for stability are:

$$P_i > 0 \;\; \forall i \in \mathcal{I} \tag{2.37}$$
$$A_j^T P_i A_j - P_j < 0 \;\; \forall (i,j) \in \mathcal{S}, \tag{2.38}$$

where $\mathcal{S}$ is the set of all possible switching from region $\mathcal{X}_i$ to $\mathcal{X}_j$ which is:

$$\mathcal{S} = \{(i,j) | \exists k \in \mathbb{N}_0 \; such \; that \; x(k) \in \mathcal{X}_i, x(k+1) \in \mathcal{X}_j\} \tag{2.39}$$

The set $\mathcal{S}$ can be computed using reachability analysis for MLD systems. But another conservative alternative is to consider all switchings i.e. $\mathcal{S} = \mathcal{I} \times \mathcal{I}$. In this case the number of LMIs to be satisfied increases quadratically with respect to the increase in the number of polyhedral regions.

Using the same method as before we have the following lemma:

Lemma 2.2. *The PWA system (2.24) with piecewise linear state feedback (2.29) is asymptotically stable if there exist $X_i = X_i^T > 0$ and matrices Y_i such that:*

$$\begin{bmatrix} X_i & (A_jX_j + B_jY_j) \\ (A_jX_j + B_jY_j) & X_j \end{bmatrix} < 0 \ \forall (i,j) \in \mathcal{S}. \tag{2.40}$$

Then, the piecewise linear state feedback gains are given by:

$$K_i = Y_i {Q_i}^{-1}. \tag{2.41}$$

2.6 Equivalence between different classes

In [HSB01], different model classes of hybrid systems are studied and it is shown that under some assumptions they are equal. Under the assumption that the set of states, and input are bounded and that PWA, MLD, LC, ELS, and MMPS models are well-posed, then they are equivalent. Fig. 2.4 summarizes the equivalence between different classes by means of a graph. If A is connected by an edge to B, then the class A is a subset of the class B. A star on an edge means that there are some restrictive conditions for inclusion of A in B.

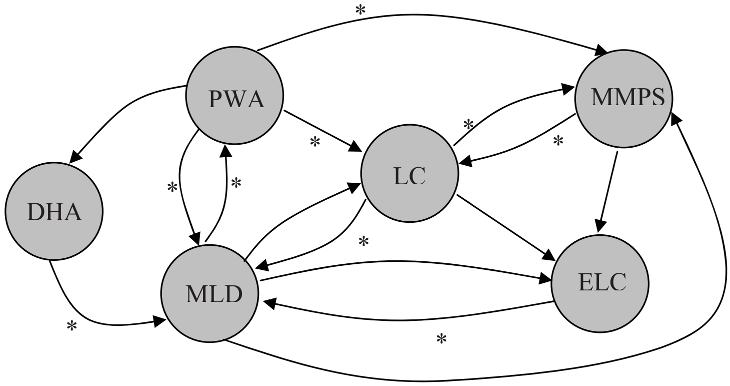

Figure 2.4: Equivalence between different classes of hybrid systems[HSB01]

3 | Summary of Contributions

The contributions of this thesis area in the form of several publications in the are of fault diagnosis and fault tolerant control of hybrid systems. In this chapter, the contributions are highlighted.

3.1 Active Fault Diagnosis of Hybrid systems

The first four papers A[TRIZB09], B[TIZBR09], C[TIZRB10], D[TRIZB10] deal with the problem of AFD of hybrid systems. In Paper A, a method for AFD of hybrid systems using reach set computation is developed. We have used the hybrid automaton modeling framework. Discrete faults and continuous faults are modeled as discrete modes. The system can be in a normal condition, N, or in a faulty condition, F, and the corresponding models are respectively denoted by Σ_N and Σ_F. The aim of the diagnoser is to find the condition of the system. It looks for two distinguishable trajectories χ_1 and χ_2 from the normal model , Σ_N, and the faulty model, Σ_F, of the system. This is done by computing the reach set of both models starting from the initial state and using all possible inputs, denoted respectively by $\mathcal{R}_{N_k}, \mathcal{R}_{F_k}$. If the corresponding outputs of these sets, $Y(\mathcal{R}_{N_k}), Y(\mathcal{R}_{F_k})$ are equal, then the algorithm computes reach sets for future steps until they become different, i.e. $\Delta_k = (Y(\mathcal{R}_{N_k}) \cup Y(\mathcal{R}_{F_k})) \backslash (Y(\mathcal{R}_{N_k}) \cap Y(\mathcal{R}_{F_k})) \neq \emptyset$. Then the algorithm choose a target which uniquely belongs to $\mathcal{R}_{N_k}$ or $\mathcal{R}_{F_k}$. The optimal input to reach this target is computed and applied to the system. The condition of the system is determined based on observing whether it reaches its target or not. The algorithm finds the minimum length required for diagnosis. It is also explained how we can extend the algorithm for more than one faulty mode. The method is tested on a two tank example for sanity check as well as for periodic diagnosis during nominal operation.

In Paper B, the AFD method proposed in paper A is used for automatic sensor assignment. For a system with n sensors, the problem of sensor assignment is to find among all permutations, the one that conforms to the dynamic behavior of the system. There are $n!$ candidates for the sensor assignment and if wrong assignments be considered as faults, then there are $n! - 1$ fault hypothesis. Applying the algorithm from Paper A directly, is computationally very expensive. In this paper the algorithm is modified such that it needs only one model of the system. The method is tested for sensor assignment of a super market refrigeration system.

The method proposed in Paper A is based on finding distinguishable reach sets, but we are interested in distinguishable trajectories. Therefore it is possible to miss some solutions of the problem. Moreover, the problem is solved in an open loop manner. Since in AFD we are exciting the system with the aim of fault detection and not with the aim of control, it is also possible that we drive the system to an area which is unstable. In Paper C, the problem is formulated as an optimization problem which searches for distinguishable output trajectories and also guarantees stabilizability of the system. In this paper we have used the MLD framework and the problem is formulated as a mixed integer optimization problem.

The aim of diagnosis is to find a sequence of input such that the outputs of the normal system and the faulty system are observably different:

$$|y_i(T_d)) - y_j(T_d)| \geq d \ \forall i,j \in \{0,\cdots,n\}, i \neq j,$$

where y_i denotes the output of the normal system when $i = 0$ and the output of the system subject to fault f_i when $i \neq 0$. T_d is the length required for diagnosis and d is a separation distance which is chosen based on the level of noise. This is formulated as a mixed integer feasibility test by translating the above constraint into mixed integer inequalities using techniques described in (2.4). The above constraint is equivalent to fault isolation for every fault. Formulations for fault detection and fault isolation for a set of fault are also given. Minimum length of the input sequence required for diagnosis is found by finding an upper bound and a lower bound for it and then running the bisection algorithm.

To guarantee the stability we use MPC. Assume that we want to make sure that the system is stabilizable in T sampling times. It must be ensured that the final state of the diagnosis for each system, $x_i(T_d)$, is steerable to the equilibrium of the system, x_{i_er}, in T steps. Then, $x_i(T_d)$ must be in the feasible set $\mathbb{X}_F^i(T)$, which is the feasible set of the MPC contoller designed for the system subject to fault i with a prediction horizon T. In other words, for all $x_i(T_d)$ there must exist a $\{u_i\}_{T_d}^{T_d+T-1} \in \mathbb{U}^T \times \{0,1\}^{Tm_l}$ such that $\mathbf{x}_{i_{T_d}}(x_i(T_d), \{u_i\}_{T_d}^{T_d+T-1}) \in \mathbb{X}^T \times \{0,1\}^{Tn_l}, x(T|T_d) = x_{i_{er}}$. This is added as a stability constraint to the diagnosis optimization problem. The solution is applied to the system and at $t = T_d$ the fault is diagnosed. After fault diagnosis, the system reconfiguration is performed by updating the dynamic constraints of MPC to that of the identified faulty system.

The method is tested on the two tank system. The simulation result shows that T_d and T is dependent on d: the bigger the d, the longer the length of input sequence.

Paper D addresses the stability issue in AFD but using another approach. To avoid instability due to perturbation, the system is moved from its current state to another steady state. The diagnoser looks for steady states from the normal and faulty systems such that the corresponding steady inputs are equal but their steady outputs are different. If such steady states exist, then the system is diagnosable by this method. Fault diagnosis is then performed by applying the steady input and observing the steady output. Note that distinguishable steady output might not exist or it might take a long time to reach them. In this case the method proposed in Paper C is preferable.

3.2 Fault-Tolerant Control of Hybrid System

The last two papers are in the area of PFTC for Hybrid systems. We have used the PWL modeling framework to formulate stability conditions despite faults in terms of LMI inequality constraints.

In Paper E[TIZBR10], we consider the problem of PFTC for PWL systems against actuator faults. A faults is modeled as a partial loss of the actuator gains:

$$u_j^F = (1-\alpha_j)u_j, \;\; 0 \leq \alpha_j \leq \alpha_{M_j}, \tag{3.1}$$

where u_j is the $j'th$ actuator and u_j^F is the $j'th$ failed actuator. α_j denotes the percentage of failure in the $j'th$ actuator, α_{M_j} is the maximum loss in the $j'th$ actuator. $\alpha_j = 0$ presents the case of no fault in the jth actuator, $0 < \alpha_j < 1$ corresponds to the partial loss of it, and $\alpha_j = 1$ corresponds to complete loss of it. The problem is to design a piecewise linear state feedback control such that the closed loop system remains stable despite the faults in actuators. We consider the following quadratic performance index:

$$J = \sum_{k=0}^{\infty} x^T(k)Qx(k) + u^T(k)Ru(k). \tag{3.2}$$

Using PWQ Lyapunov functions a PWL state feedback control is designed such that it ensures the stability of the closed loop system and also provides an upper bound on the performance index as:

$$J \leq x(0)^T P_{i_0} x(0), \tag{3.3}$$

with $x(0) \in \mathcal{X}_{i_0}$, i.e. i_0 is the index of the initial region. The problem is cast as feasibility of a set of LMIs. The upper bound on (3.2) is minimized independent of the initial condition by considering $x(0)$ as a random variable uniformly distributed on a bounded region. This problem is solved as a convex optimization problem with LMI constraints. In the simulation results it is discussed how one can perform a trade off between the performance of the system and the degree of the system tolerance to the partial loss of actuator gains.

In Paper F[TIZBR10], we have extended the previous results using output feedback because states of a system are not always available. In this paper, we use the framework of uncertain PWL systems. Faults can be modeled as uncertainties and then an output feedback control is designed that can stabilize the closed loop system and provide an upper bound on the quadratic performance index. The following PWL system is considered:

$$\begin{aligned} x(k+1) &= (A_i + \Delta A_i)x(k) + (B_i + \Delta B_i)u(k) \quad for\; x \in \mathcal{X}_i \\ y(k) &= (C_i + \Delta C_i)x(k), \end{aligned} \tag{3.4}$$

where $x(k) \in \mathbb{R}^n$ is the state, $u(k) \in \mathbb{R}^m$ is the control input, and $y(k) \in \mathbb{R}^p$ is the output. $\{\mathcal{X}_i\}_{i=1}^s \subseteq \mathbb{R}^n$ denotes a partition of the state into a number of polyhedral regions $\mathcal{X}_i, i \in \mathcal{I} = \{1, \cdots, s\}$. Each polyhedral region is represented by:

$$\mathcal{X}_i = \{x | \mathcal{H}_i x \leq h_i\}, \tag{3.5}$$

and $\Delta A_i, \Delta B_i, \Delta C_i$ are parameter uncertainties in the parameters of the subsystem i of the following form:

$$\begin{bmatrix}\Delta A_i, & \Delta B_i\end{bmatrix} = M_{1_i} H \begin{bmatrix} N_{A_i}, & N_{B_i}\end{bmatrix}, \tag{3.6}$$

$$\Delta C_i = M_{2_i} H N_{C_i}, \tag{3.7}$$

where H is an uncertain matrix bounded by:

$$HH^T \leq I. \tag{3.8}$$

and $M_{1_i}, M_{2_i}, N_{A_i}, N_{B_i}, N_{C_i}$ are known constant matrices of appropriate dimensions. The problem is to design an output feedback of the form:

$$\begin{aligned} x_c(k+1) &= A_{c_i} x_c(k) + B_{c_i} y(k) \quad for\ x_c \in \mathcal{X}_i \\ u(k) &= C_{c_i} x_c(k) + D_{c_i} y(k). \end{aligned} \tag{3.9}$$

The novelty of the approach is that we do not make the common unrealistic assumption that the switching of the controller is based on the state of the system, i.e. $x(t)$, but we assume that it is based on the estimated state $x_c(t)$. Here, the problem is formulated in terms of BMIs. Using a similar procedure as before, the upper bound on the performance is minimized by solving an optimization problem with BMI constraints. The optimization problem is solved using the V-K iteration algorithm.

4 | Conclusions and Future Work

Due to the increasing safety, reliability, and performance requirements of modern technological systems, Fault Detection and Diagnosis and Fault Tolerant Control of them is very important. Most of modern technological systems consist of both discrete and continuous behaviors and interactions between them. Hybrid systems are a useful modeling class to capture behavior of these systems. In this thesis, we investigated the problem of FDD and FTC for hybrid systems. The accomplished results provide contributions to both areas of fault diagnosis and fault-tolerant control of hybrid systems. Contributions of the thesis in the area of fault diagnosis are:

- A new methods for active fault diagnosis of hybrid systems based on the reach set computation. This method performs AFD in an open loop manner and instead of searching for distinguishable trajectories it looks for distinguishable reach sets. As a result, it might conclude that a fault is not diagnosable while it is indeed diagnosable. However, this algorithm finds automatically the minimum length for the input sequence required for active diagnosis.

- An extension of the above method for sensor assignment during the commissioning phase with application to a supermarket refrigeration system.

- A stabilizable AFD method for hybrid systems which formulates the problem as an optimization problem. This method guarantees stabilizability despite the excitation signal that is used for fault diagnosis and it looks for distinguishable trajectories. To find the minimum input sequence length for the diagnosis, a bisection algorithm is used by finding upper and lower feasible bounds for the input sequence length.

- A method for AFD of hybrid systems using distinguishable steady outputs. This method uses the steady state properties of the system. It looks for distinguishable steady states from the normal and the faulty system such that the corresponding outputs are distinguishable. Because the system is moved to a steady state, the method preserves stability but distinguishable steady outputs does not always exist.

In the area of FTC of hybrid system, we consider the problem of passive fault-tolerant control for PWL systems. Contribution of the thesis in this area are:

- A guaranteed cost method for PFTC of PWL systems against actuator faults. Piecewise state feedback is used to design a PFTC for PWL systems such that it can tolerate a partial loss of actuator gains. The problem is formulated in terms of LMIs.

The optimal guaranteed cost is found by solving a convex optimization problem with LMI constraints.

- A guaranteed cost method for output feedback control of uncertain PWL systems. Because most of the time, states of a system are not measurable, the previous method is extended to dynamic output feedback control. The problem is formulated in the more general framework of uncertain PWL systems, hence sensor faults and internal faults may also be considered. The output feedback problem is cast as the feasibility of a set of BMIs and it is solved using the V-K iteration algorithm.

Some possible future directions for this research are listed in the following:

- The reach set computation based method can be extended by considering polyhedron bounded disturbance and noise explicitly as well as assuming the initial state to be given by a polyhedron. It is possible to extend the approach by using a moving horizon scheme such that at each time instant, the initial state is updated by using set-based state estimation methods. A challenge here is that the set-based estimation methods for hybrid systems are not well developed. An alternative is to assume that states are estimated with a given upper bound on the estimation error.

- The output feedback method is formulated in terms of BMIs. BMI problems are in general non-convex and $\mathcal{NP}$-hard. The iterative algorithms used for solving them provide a sub-optimal solution. It would be useful to formulated the problem in terms of LMIs, though it is not clear how to do that for uncertain PWL systems.

- A PFTC cannot handle many faults, especially when severe faults might happen it is either infeasible to design a controller that can handle all faults or the resulting controller presents a very low performance. AFTC systems have this advantage that they can switch among different controllers designed for each faulty situation, but there is a risk of instability due to the delays associated with detection and isolation of faults or due to the wrong decision by FDD system. A possibility for future work is to combine AFTC with PFTC to design a reconfigurable controller for PWL systems that guarantees stability of the closed loop system during the delay associated with FDD. The idea is to find minimal recoverable configurations and find a common solution to these configurations. Then, update the controllers based on the new information from the FDD module. It is desirable to be able to update the controllers such that the instability which might happen because of the incorrect information from the FDD module is avoided.

References

[AC01] A. Alessandri and P. Coletta. Design of Luenberger observers for a class of hybrid linear systems. *Hybrid systems: computation and control*, pages 7–18, 2001.

[AHLP00] R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, 2000.

[AK03] P. Antsaklis and X. Koutsoukos. Hybrid systems: Review and recent progress. In T. Samad and G. Balas, editors, *Software-Enabled Control*, pages 271–298. IEEE Press, 2003.

[And08] I.V. Andjelkovic. *Auxiliary Signal Design for Fault Detection for Nonlinear Systems: Direct Approach*. PhD thesis, North Carolina State University, 2008.

[ASC08] I. Andjelkovic, K. Sweetingham, and S.L. Campbell. Active fault detection in nonlinear systems using auxiliary signals. In *American Control Conference*, pages 2142–2147, Seattle, WA, 2008.

[BBBSV02] A. Balluchi, L. Benvenuti, M. D. D Benedetto, and A.L. Sangiovanni-Vincentelli. Design of observers for hybrid systems. In *5th International Workshop on Hybrid Systems: Computation and Control*, pages 76–89, London, UK, 2002. Springer-Verlag.

[BFK$^+$00] M. Blanke, C. Frei, F. Kraus, R. J. Patton, and M. Staroswiecki. What is fault tolerant control? In *4th IFAC symposium on fault detection, supervision and safety for technical processes*, pages 40–51, 2000.

[BIZBL97] M. Blanke, R. Izadi-Zamanabadi, S. A. Bogh, and Z. P. Lunau. Fault-tolerant control systems-a holistic view. *Control Engineering Practice*, 5(5):693–702, 1997.

[BKLS06] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag, 2006.

[BM99] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–428, 1999.

[BMM99] A. Bemporad, D. Mignone, and M. Morari. Moving horizon estimation for hybrid systems and fault detection. In *the Proceedings of American Control Conference*, volume 4, pages 2471 –2475, 1999.

[BSW01] M. Blanke, M. Staroswiecki, and N. E. Wu. Concepts and methods in fault-tolerant control. In *American Control Conference*, volume 4, pages 2606–2620, 2001.

[BTMO09] M. Bayoudh, L. Travé-Massuyes, and X. Olive. Active diagnosis of hybrid systems guided by diagnosability properties. In *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, 2009.

[CCN09] D. Choe, S. L. Campbell, and R. Nikoukhah. Optimal piecewise-constant signal design for active fault detection. *International Journal of Control*, 82(1):130–146, 2009.

[CDA$^+$06] S. L. Campbell, K. J. Drake, I. Andjelkovic, K. Sweetingham, and D. Choe. Model based failure detection using test signals from linearizations: A case study. In *IEEE International Conference on Control Applications*, pages 2659–2664, 2006.

[CEMS04] V. Cocquempot, T. El Mezyani, and M. Staroswiecki. Fault detection and isolation for hybrid systems using structured parity residuals. In *5th Asian Control Conference,*, volume 2, 2004.

[CHN02] S. L. Campbell, K. G. Horton, and R. Nikoukhah. Auxiliary signal design for rapid multi-model identification using optimization. *Automatica*, 38(8):1313–1325, 2002.

[CM01] F.A. Cuzzola and M. Morari. A generalized approach for analysis and control of discrete-time piecewise affine and hybrid systems. *Hybrid Systems: Computation and Control*, 2034:189–203, 2001.

[CN04] S.L. Campbell and R. Nikoukhah. *Auxiliary Signal Design for Failure Detection*. Princeton University Press, 2004.

[CP99] J. Chen and R.J. Patton. *Robust model-based fault diagnosis for dynamic systems*. Kluwer Academic Publishers Norwell, MA, USA, 1999.

[Cpl03] I. Cplex. 9.0 Users Manual. *ILOG Inc., Gentilly, France*, 2003.

[DB09] M. J. Daigle and G. Biswas. Improving diagnosability of hybrid systems through active diagnosis. In *Safeprocess09*, pages 217–222, 2009.

[DC01] R. Dearden and D. Clancy. Particle filters for real-time fault detection in planetary rovers. In *12th International Workshop on Principles of Diagnosis*, pages 1–6, 2001.

[DKB09] M. J. Daigle, X. D. Koutsoukos, and G. Biswas. An event-based approach to integrated parametric and discrete fault diagnosis in hybrid systems. *Transactions of the Institute of Measurement and Control*, pages 1–24, 2009.

[FD97] P. M. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of process control*, 7(6):403–424, 1997.

[Fen02] G. Feng. Controller design and analysis of uncertain piecewise-linear systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 49(2):224 –232, 2002.

[FL01] D. Forstner and J. Lunze. Discrete-event models of quantized systems for diagnosis. *International Journal of Control*, 74(7):690–700, 2001.

[FTMM02] G. Ferrari-Trecate, D. Mignone, and M. Morari. Moving horizon estimation for hybrid systems. *IEEE Transactions on Automatic Control*, 47(10):1663–1676, 2002.

[Ger97] J. Gertler. Fault detection and isolation using parity relations. *Control Engineering Practice*, 5(5):653–661, 1997.

[GLC08] Y. Goa, Z. Liu, and H. Chen. Robust H_∞ control for constrained discrete-time piecewise affine systems with time-varying parameter uncertainties. *IET Control Theory and Application*, 3:1132–1144, 2008.

[GS90] J. Gertler and D. Singer. A new structural framework for parity equation-based failure detection and isolation. *Automatica*, 26(2):381–388, 1990.

[HSB01] W. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, 2001.

[HW02] M. Hofbaur and B. Williams. Mode estimation of probabilistic hybrid systems. In *Hybrid systems: Computation and Control*, volume 3927 of *Lecture Notes in Computer Science*, pages 253–266. Springer, 2002.

[HW04] M. W. Hofbaur and B. C. Williams. Hybrid estimation of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(5):2178–2191, 2004.

[IF91] R. Isermann and B. Freyermuth. Process fault diagnosis based on process model knowledge: Part I-principles for fault diagnosis with parameter estimation. *Journal of Dynamic Systems, Measurement, and Control*, 113:620, 1991.

[Ise93] R. Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing. *Automatica*, 29(4):815–835, 1993.

[Ise05] R. Isermann. Model-based fault-detection and diagnosis-status and applications. *Annual Reviews in control*, 29(1):71–85, 2005.

[Ise06] R. Isermann. *Fault-diagnosis systems*. Springer Verlag, 2006.

[Jia05] J. Jiang. Fault-tolerant control systems-an introductory overview. *Acta Automatica Sinica*, 31(1):161–174, 2005.

[JR98] J. Johansson and A. Rantzer. Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):555–559, 1998.

[Kan97] M. Kantner. Robust stability of piecewise linear discrete time systems. In *Proceedings of American Control Conference*, volume 2, pages 1241 –1245 vol.2, jun 1997.

[KKZ02] X. Koutsoukos, J. Kurien, and F. Zhao. Monitoring and diagnosis of hybrid systems using particle filtering methods. In *International Symposium on Mathematical Theory of Networks and Systems*, 2002.

[KKZ03] X. Koutsoukos, J. Kurien, and F. Zhao. Estimation of distributed hybrid systems using particle filtering methods. In *Hybrid systems: Computation and Control*, volume 2623 of *Lecture Notes in Computer Science*, pages 298–313. Springer, 2003.

[LR05] J.T. Linderoth and T.K. Ralphs. Noncommercial software for mixed-integer linear programming. *Integer Programming: Theory and Practice*, pages 253–303, 2005.

[LSVW96] N. Lynch, R. Segala, F. Vaandrager, and H. Weinberg. *Hybrid Systems III*, chapter Hybrid I/O automata, pages 496–510. Lecture Notes in Computer Science. Springer, 1996.

[Lun00] J. Lunze. Diagnosis of quantized systems based on a timed discrete-event mode. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 30(3):322–335, 2000.

[Lun08] J. Lunze. Fault diagnosis of discretely controlled continuous systems by means of discrete-event models. *Discrete Event Dynamic Systems*, 18(2):181–210, 2008.

[MFTM00] D. Mignone, G. Ferrari-Trecate, and M. Morari. Stability and stabilization of piecewise affine and hybrid systems: an LMI approach. In *Proceedings of the 39th IEEE Conference on Decision and Control*, volume 1, pages 504 –509 vol.1, 2000.

[Mig02] D. Mignone. *Control and estimation of hybrid systems with mathematical optimization*. PhD thesis, Swiss Federal Institute of Technology, 2002.

[MRRS00] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 36:789–814, 2000.

[Nar02] S. Narasimhan. *Model-based diagnosis of hybrid systems*. PhD thesis, Vanderbilt University, 2002.

[NB07] S. Narasimhan and G. Biswas. Model-based diagnosis of hybrid systems. *IEEE transactions on man and cybernetics*, 37(3):347–361, 2007.

[NC06] R. Nikoukhah and S. L. Campbell. Auxiliary signal design for active failure detection in uncertain linear systems with a priori information. *Automatica*, 42(2):219–228, 2006.

[NCD00] R. Nikoukhah, S. L. Campbell, and F. Delebecque. Detection signal design for failure detection: a robust approach. *International Journal of Adaptive Control and Signal Processing*, 14(7):701–724, 2000.

[NCD10] R. Nikoukhah, S. L. Campbell, and K. Drake. An active approach for detection of incipient faults. *International Journal of Systems Science*, 41(2):241–257, 2010.

[Nie06] H. H. Niemann. A setup for active fault diagnosis. *IEEE Transactions on Automatic Control*, 51(9):1572–1578, 2006.

[Nik98] R. Nikoukhah. Guaranteed active failure detection and isolation for linear dynamical systems. *Automatica*, 34(11):1345–1358, 1998.

[NRZ09a] N. Nayebpanah, L. Rodrigues, and Y. Zhang. Fault identification and reconfigurable control for bimodal piecewise affine systems. In *European Control Conference*, pages 2694–99, 2009.

[NRZ09b] N. Nayebpanah, L. Rodrigues, and Y. Zhang. Fault-tolerant controller synthesis for piecewise-affine systems. In *IEEE American Control Conference*, pages 222–226, 2009.

[OMP08] C. Ocampo-Martinez and V. Puig. Fault-tolerant model predictive control within the hybrid systems framework: Application to sewer networks. *International Journal of Adaptive Control and Signal Processing*, 23(8):757–787, 2008.

[Pat97] R. J. Patton. Fault-tolerant control systems: The 1997 situation. In *3rd IFAC symposium on fault detection supervision and safety for technical processes*, volume 3, pages 1033–1054, 1997.

[PC97] R. J. Patton and J. Chen. Observer-based fault detection and isolation: robustness and applications. *Control Engineering Practice*, 5(5):671–682, 1997.

[PFC89] R.J. Patton, P.M. Frank, and R.N. Clarke. *Fault diagnosis in dynamic systems: theory and application*. Prentice Hall, 1989.

[PFC00] R. Patton, P.M. Frank, and R. Clark. *Issues of fault diagnosis for dynamic systems*. Springer Verlag, 2000.

[PN07] N. K. Poulsen and H. H. Niemann. Stochastic change detection based on an active fault diagnosis approach. In *46th IEEE Conference on Decision and Control*, pages 346–351, 2007.

[PN08] N. K. Poulsen and H. H. Niemann. Active fault diagnosis based on stochastic tests. *International Journal of Applied Mathematics and Computer Science*, 18(4):487–496, 2008.

[RCB00] E. Russell, L. H. Chiang, and R. D. Braatz. *Data-driven methods for fault detection and diagnosis in chemical processes*. Springer Verlag, 2000.

[RH05] A. Richards and J. How. Mixed-integer programming for control. In *American Control Conference*, pages 2676 – 2683, 2005.

[RHvdWL08] J. H. Richter, W. Heemels, N. van de Wouw, and J. Lunze. Reconfigurable control of PWA systems with actuator and sensor faults: stability. In *47th IEEE Conference on Decision and Control*, pages 1060–1065, 2008.

[RHvdWL10] J. H. Richter, W. Heemels, N. van de Wouw, and J. Lunze. Reconfigurable control of piecewise affine systems with actuator and sensor faults: stability and tracking1. Technical report, Institute of Automation and Computer Control, Ruhr-Universität Bochum, 2010.

[RTS06] M. Rodrigues, D. Theilliol, and D. Sauter. Fault tolerant control design for switched systems. In *2nd IFAC Conference on Analysis and design of hybrid systems*, 2006.

[SN10] J Stoustrup and H. H Niemann. Active fualt diagnosis by controller modification. *International Journal of Systems Science*, 2010.

[TB04] F. D. Torrisi and A. Bemporad. HYSDEL-a tool for generating computational hybrid models for analysis and synthesis problems. *IEEE Transactions on Control Systems Technology*, 12(2):235–249, 2004.

[TBIZ10] S. Tabatabaeipour, T. Bak, and R. Izadi-Zamanabadi. Output feedback guaranteed cost control of discrete-time piecewise linear systems. *submitted to IET Control Theory and Application*, 2010.

[TIZBR09] S. Tabatabaeipour, R. Izadi-Zamanabadi, T. Bak, and A. P. Ravn. Automatic sensor assignment of a supermarket refrigeration system. In *IEEE Multi-Conference on Control Applications, (CCA) & Intelligent Control, (ISIC),*, pages 1319–1324, July 2009.

[TIZBR10] S. Tabatabaeipour, R. Izadi-Zamanabadi, T. Bak, and A. P. Ravn. Passive fault-tolerant control of piecewise linear systems against actuator faults. *submitted to Internationl Journal of Systems Scinece*, 2010.

[TIZRB10] S. Tabatabaeipour, R. Izadi-Zamanbadi, A. P. Ravn, and T. Bak. Stabilizable active diagnosis of hybrid systems. *submitted to International Journal of Control*, 2010.

[TMFTM01] K. Tsuda, D. Mignone, G. Ferrari-Trecate, and M. Morari. Reconfiguration strategies for hybrid systems. In *American Control Conference*, volume 2, pages 867–873, 2001.

[TRIZB09] S. Tabatabaeipour, A. P. Ravn, R. Izadi-Zamanabadi, and T. Bak. Active fault diagnosis of linear hybrid systems. In *Safeprocess09*, pages 211–216, 2009.

[TRIZB10] S. Tabatabaeipour, A. P. Ravn, R. Izadi-Zamnabadi, and T. Bak. Active diagnosis of MLD systems using distinguishable steady outputs. In *IEEE International Symposium on Industrial Electronics*, 2010.

[Vei95] R.J. Veillette. Reliable linear-quadratic state-feedback control. *Automatica*, 31(1):137–144, 1995.

[VRK03] V. Venkatasubramanian, R. Rengaswamy, and S.N. Kavuri. A review of process fault detection and diagnosis:: Part II: Qualitative models and search strategies. *Computers & Chemical Engineering*, 27(3):313–326, 2003.

[VRKY03] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, and K. Yin. A review of process fault detection and diagnosis:: Part III: Process history based methods. *Computers & Chemical Engineering*, 27(3):327–346, 2003.

[VRYK03] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S.N. Kavuri. A review of process fault detection and diagnosis:: Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, 27(3):293–311, 2003.

[WLZ07] R. Wang, M. Liu, and J. Zhao. Reliable H_∞ control for a class of switched nonlinear systems with actuator failures. *Nonlinear Analysis: Hybrid Systems*, 1(3):317–325, 2007.

[WLZL07] W. Wang, L. Li, D. Zhou, and K. Liu. Robust state estimation and fault diagnosis for uncertain hybrid nonlinear systems. *Nonlinear Analysis: Hybrid Systems*, 1(1):2–15, 2007.

[YCJ08] H. Yang, V. Cocquempot, and B. Jiang. Fault tolerance analysis for switched systems via global passivity. *IEEE transaction on Circuits and Systems II*, 55:1279 –1283, 2008.

[YJC09] H. Yang, B. Jiang, and V. Cocquempot. A fault tolerant control framework for periodic switched non-linear systems. *International Journal of Control*, 82(1):117–129, 2009.

[YLW98] G. H. Yang, J. Lam, and J.L. Wang. Reliable H_∞ control for affine nonlinear systems. *IEEE Transactions on Automatic Control*, 43(8):1112–1117, Aug 1998.

[YWS00] G.H. Yang, J.L. Wang, and Y.C. Soh. Reliable guaranteed cost control for uncertain nonlinear systems. *IEEE Transactions on Automatic Control*, 45(11):2188–2192, 2000.

[YWS01] G.H. Yang, J.L. Wang, and Y.C. Soh. Reliable H_∞ controller design for linear systems. *Automatica*, 37(5):717–725, 2001.

[YWSL03] G.H. Yang, J.L. Wang, Y.C. Soh, and K.Y. Lou. Reliable state feedback control synthesis for uncertain linear systems. *Asian Journal of Control*, 5(2):301–308, 2003.

[YYLW98] G. H. Yang, Zhang S. Y., J. Lam, and J.L. Wang. Reliable control using redundant controllers. *IEEE Transactions on Automatic Control*, 43(11):1588–1593, Nov 1998.

[ZJ06] Y. M. Zhang and J. Jiang. Issues on integration of fault diagnosis and reconfigurable control in active fault-tolerant control systems. In *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processed*, pages 1513–1524, 2006.

[ZJ08] Y. Zhang and J. Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2):229–252, 2008.

[ZKH$^+$05] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, and P. Cheung. Monitoring and fault diagnosis of hybrid systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 6:1225–1240, 2005.

[ZT08] J. Zhang and W. Tang. Output feedback H_∞ control for uncertain piecewise linear systems. *Journal of Dynamical and Control Systems*, 14(1):121–144, 2008.

[ZT09] J. Zhang and W. Tang. Output feedback optimal guaranteed cost control of uncertain piecewise linear systems. *International Journal of Robust and Nonlinear Control*, 19:596–590, 2009.

Articles

Paper A

Active Fault Diagnosis of Linear Hybrid Systems

Seyedmojtaba Tabatabaeipour, Anders P. Ravn, Roozbeh Izadi-Zamanabadi, and Thomas Bak

This paper is published in :
the Proceeding of 7th IFAC International Symposium on Fault Detection, Supervision, and Safety of Technical Processes, pp. 211-216

Abstract

A method for active fault diagnosis of linear discrete time hybrid systems is presented. The algorithm generates appropriate test signals that can be used for sanity check during system commissioning or later in the normal phase to detect faults which are impossible to detect by means of passive diagnosis methods because of regulatory actions of the controller. The algorithm is illustrated on a two tank benchmark example.

1 Introduction

In complex large control systems there are many components with strong interaction between them. Hence the overall system depends crucially on the individual performance of the components. Therefore a fault in a single component may degrade the overall performance of the system and may even leads to unacceptable loss of system functionality. Thus fault diagnosis is of crucial importance in automatic control of complex large systems.

There are two main approaches to fault diagnosis: active and passive. In the passive approach the diagnoser observes the input and output of the system and based on the measured I/O decides whether a fault has occurred or not. Most of the available methods for fault diagnosis are of this kind.

In active fault diagnosis the diagnoser generates a test signal which excites the system to decide whether the system represents the normal behaviour or the faulty behaviour and if possible decides which faulty behaviour occurs. The test signal should be designed such that it affects the overall system as little as possible although enough to make fault diagnosis possible. The advantage of the active approach is in the operating points where the normal system and faulty system represents the same behaviour. Under such circumstances it is possible to detect faults faster by active diagnosis. Active fault diagnosis can also be used to provide sanity check in the commissioning phase by generating an appropriate test signal.

Modelling of complex systems are captured by hybrid system theory, which has been subject of intensive research in recent years, for an overview see [1]. Generally speaking, a hybrid system is a dynamical system with both continuous and discrete behaviours and non-trivial interaction between continuous evolutions and discrete transitions.

Fault diagnosis of hybrid systems has been investigated recently, for a survey one can look at [2], [3], [4]. A class of approaches for diagnosis of hybrid systems uses discrete/temporal abstraction of the continuous dynamics [5]. In [2], the diagnoser uses a discrete event abstraction of the system and the continuous dynamics information is used when it becomes necessary. In [4], the authors use a Petri net abstraction for dealing with continuous behaviour of hybrid systems. In [3] a model based diagnosis method based on a hybrid bond graph modelling framework is proposed. Particle filtering methods are another class of methods for diagnosis of hybrid systems; [6], [7].

All of the aforementioned approaches are in the area of passive diagnosis. In [8], [9] the problem of active diagnosis for linear system using an auxillary signal for fault detection is investigated. The results of [8] is extended for nonlinear systems in [10] using linearization and also a direct optimization approach. In the field of discrete event systems, some approaches have been proposed for active diagnosis. Active diagnosis of

DES is studied in [11] and input sequence for diagnosis is computed. [12] studied the active diagnosis problem of DES as a supervisory control problem.

To the knowledge of the authors there is no research considering directly active fault diagnosis of hybrid systems. In this paper an active fault diagnosis method for diagnosis of linear hybrid system in discrete time is proposed. The idea is based on reach set computations for the faulty and the normal system. For both systems, those states that the system can reach in forthcoming steps considering all possible excitations are considered. Reach sets are computed as long as the faulty system and the normal system have the same reach sets. But as soon as they represent different sets the algorithm terminates and selects a point which uniquely belongs to one of the sets. Then the optimal input for reaching the selected point is calculated and injected to the system. If the system reaches the selected point then it is in the corresponding mode, otherwise it is in the other mode.

This paper is organized as follows. An outline of the approach and some preliminaries are given in Section 2. Section 3 describes the algorithm. In Section 4 the proposed algorithm is applied to the two tank benchmark example. Section 5 provides conclusions and topics for future investigation.

2 Outline of the Method

Most model-based diagnostic methods follow the same principle [13]. They observe a sequence of measured input and output of the system and decide whether the measured I/O pair is consistent with the model that describes the behaviour of the system. If the consistency is not confirmed a fault is detected.

Suppose that the current observed I/O pair is A or B as depicted in Fig. 5.1. The set $\mathbf{B}_0$ represents the normal behaviour of the system and the set $\mathbf{B}_1$ represents the behaviour of the system subject to the fault f_1. As long as A or B belong uniquely to the sets $\mathbf{B}_0$ or $\mathbf{B}_1$, the diagnoser can decide whether the system is in its normal operation or subject to a fault. The ambiguity arises when the observed data is the I/O pair C, which belongs to the area where the normal behaviour and the faulty behaviour of the system overlap. In this case, the diagnoser can not distinguish if the system is subject to the fault f_1 or in the normal operation. The main idea of active fault diagnosis is to exert an input signal to the system to move C to an area which belongs uniquely either to the set $\mathbf{B}_0$ or $\mathbf{B}_1$.

The active diagnosis algorithm in this paper assumes that we have a model of the normal and the faulty system. From current state, we predict the behaviour of the system at future time steps considering all possible inputs and using both models. We then find the first time step that the faulty and the normal system have different behaviours. Now consider the set holding these different behaviours. We choose one of them, e.g. belonging to future behaviour of the normal system. Then we find an optimal input sequence that will drive the system to a state corresponding to the selected behaviour and apply it to the system. If the output of the system reaches the corresponding output of the selected behaviour, then the system is in the normal mode otherwise it is faulty.

In order to make this idea precise, we define following terms.

Definition 5.1 (Hybrid Automaton)**.** A *hybrid automaton,* $\mathcal{H}$ is a collection $\mathcal{H} = (Q, X, U, Y, Init, f, h, Inv, E, G, J)$ where,

- Q is a set of finite discrete modes, $Q = \{q_1, q_2, \ldots, q_m\}$,

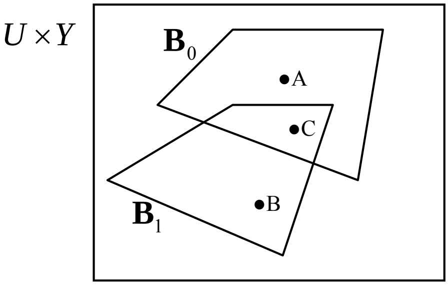

Figure 5.1: System behaviour

- X is a finite set of continuous state variables,
- U is a finite collection of input variables,
- Y is a finite collection of output variables,
- $Init \subset Q \times X$ is a set of initial states,
- $f : Q \times X \times U \rightarrow \mathbb{R}^n$ is a vector field,
- $h : Q \times X \times U \rightarrow Y$ is an output map,
- $Inv : Q \rightarrow 2^{X \times U}$ assigns to each $q \in Q$ an invariant set $Inv(q) \subseteq X \times U$,
- $E \subset Q \times Q$ is a set of discrete transitions,
- $G : E \rightarrow 2^{X \times U}$ assigns to each $e = (q, q') \in E$ a guard $g(e) \subset X \times U$,
- $J : E \times X \times U \rightarrow 2^X$ is a jump function that assigns a jump set $J(e, x, u) \subseteq X \times U$ to each pair $e \in E$ and $x \in g(e)$.

In the case of linear hybrid systems the vector field f_q is represented by a linear difference equation: $x_{i+1} = A_{q_i} x_i + B_{q_i} u_i$ and and the output map is of the form $y_{i+1} = C_{q_i} x_i + D_{q_i} u_i$.

The tuple $(q, x, u, y) \in Q \times X \times U \times Y$ is called a point of $\mathcal{H}$, $(q, x) \in Q \times X$ is called the state of $\mathcal{H}$, $u \in U$ the input and $y \in Y$ is called the output of $\mathcal{H}$. Also we refer to $(u, y) \in U \times Y$ as an observation of $\mathcal{H}$.

Definition 5.2 (Execution)**.** An execution of a hybrid automaton is a sequence $\chi = (\sigma_0, \ldots,$
$\sigma_i, \sigma_{i+1}, \ldots)$ where $\sigma_0 = (q_0, x_0, u_0, y_0)$, $\sigma_i = (q_i, x_i, u_i, y_i)$ and $\sigma_{i+1} =$
$(q_{i+1}, x_{i+1}, u_{i+1}, y_{i+1})$ such that:

- Initial condition $(q_0, x_0) \in Init$,
- Continuous evolution: for all i, $q_i = q_{i+1}$, $(x_{i+1}, u_{i+1}) \in Inv(q_i)$:

$$x_{i+1} = A_{q_i} x_i + B_{q_i} u_i$$

$$y_{i+1} = C_{q_i} x_i + D_{q_i} u_i$$

- Transition: for all i, $e = (q_i, q_{i+1}) \in E, (x_i, u_i) \in G(e)$:
  $x_{i+1} \in J(e, x_i, u_i), (x_{i+1}, u_{i+1}) \in Inv(q_{i+1})$

For modelling of faults in hybrid systems two types of faults can be considered: discrete faults and continuous faults. Discrete faults can be considered as a new mode or location in a hybrid system. Here continuous faults are also modelled as a new mode as in [2]. It is supposed that events that describe transitions from a normal location to a faulty location are unobservable events. The system can be in a normal condition N or a faulty condition F where each condition is a subset of Q. A condition set $K = \{N, F_1, \ldots, F_p\}, p > 1$ is a set of conditions that is a complete partition of the mode set Q.

For every condition $\kappa \in K$, the corresponding dynamical system, Σ_κ, is denoted by:

$$\Sigma_\kappa = \{\kappa, X, U, Y, Init, f, Inv, E_\kappa, G, J\}$$

where $E_\kappa = \{e = (q, q') \,|q \in \kappa, q' \in \kappa\}$ and $Init_\kappa \subset \kappa \times X$.

3 The proposed algorithm

The diagnoser is a system that gives us an estimate $\hat{\kappa}(k)$ of the current system condition $\kappa(k)$. A passive diagnoser recieves a sequence of observations $\langle(u(k-m), y(k-m)), \ldots, (u(k), y(k))\rangle$ as input and generetes an estimate of the current condition $\hat{\kappa}(k)$ as output. The excitation signal or the input comes from the controller.

In active diagnosis the diagnoser generates an input sequence $\langle u(k+1), \ldots, u(k+m')\rangle$, applies it to the system and observes the output sequence $\langle y(k+1), \ldots, y(k+m')\rangle$ to determine the system condition. The output of the diagnoser could be an estimate of the current condition of the system $\hat{\kappa}(m')$ or the condition of the system for some finite transition into the past $\hat{\kappa}(m' - k')$. So, the active diagnosis problem can be defined as follows:

Problem 5.1 (Active diagnosis problem)**.** *Given a hybrid automaton $\mathcal{H}$, find a sequence of input $\langle u(0), \ldots, u(m)\rangle$ such that the condition $\kappa(0)$ is determined by observing the sequence $\langle y(0), \ldots, y(m)\rangle$.*

If the input sequence exists, then we can ask for the optimal one, where optimality can be interpreted in different senses. The algorithm that is proposed in this paper looks for the shortest sequence of the inputs that can diagnose the system.

Here, it is supposed that an observer-based passive diagnoser for the hybrid system is designed which gives us the initial state of the system. For detailed description of this diagnoser, the interested reader is referred to [14] and [2]. But briefly, the diagnoser consists of a bank of observers, each one designed for a discrete mode q_i of the hybrid system. The inputs of the observers are a sequence of observations (u, y). Based on the output of the observers, a residual vector $\rho = \{r_1, \ldots r_m\}$ is generated. A zero residual r_i shows that the corresponding mode, q_i, is consistent with the input and output sequence. If the current state of the system, $(q(k), x(k))$, is determined uniquely then the condition is also determined . A problem arises when both the faulty mode and the normal mode are recognized as consistent modes with the I/O sequence. This is because these two modes have indistinguishable executions, where indistinguishable executions are defined as follows.

Definition 5.3 (Indistinguishability)**.** Given a hybrid system $\mathcal{H}$ and $\delta \in \mathbb{N}$, modes q and q' are indistinguishable on the time interval $[i, i+\delta]$ if there exist executions $\chi = (\sigma_i, \dots, \sigma_{i+\delta})$ and $\chi' = (\sigma'_i, \dots, \sigma'_{i+\delta})$, where the corresponding continuous outputs are equal.

This problem may happen very often. Consider a simple hybrid system with two discrete modes q_1 an q_2 and a switch between them (like an on/off valve) which forces the system to switch between these two modes. If the switch is stuck in one position, say, such that the mode q_1 is active, then the faulty system has exactly the same properties as the mode q_1. Therefore, the faulty mode and q_1 are indistinguishable. An advantage of our method is that there is no need for modelling efforts to make these two modes distinguishable.

3.1 The Algorithm for a system with one faulty mode

In this subsection, the proposed algorithm is described for a system with one faulty mode. Therefore, the condition set is $\{N, F\}$. The possibility for expansion of the method for more than one faulty mode is discussed in the next subsection.

The idea of the proposed method is to find two executions χ_1 and χ_2 respectively from the system in normal conditon, Σ_N, and the faulty system, Σ_F, which are distinguishable. This task is done by finding all possible outputs that both systems could reach in the future time steps considering all admissible inputs and starting from the given initial state. As soon as they represent different outputs then the required executions are found and the algorithm terminates.

To find all possible outputs that a system could reach in the future, reach set of the system and the corresponding output should be computed.

Definition 5.4 (Reach Set)**.** *Reach Set* of a hybrid automata $\mathcal{H}$ at time k denoted by $Reach_k(\mathcal{H}, \mathcal{X}(0), \mathcal{U})$ is the set of all states $(q, x) \in Q \times X$ that are reachable by a given hybrid automata $\mathcal{H}$ at time step k, starting from any initial state $x(0) \in \mathcal{X}(0)$ and with all possible inputs $u \in \mathcal{U}$.

As described in Algorithm 1, the reach set of both the normal system, $\mathcal{R}_{N_k}$, and the faulty system, $\mathcal{R}_{F_k}$, are calculated for time k. It is assumed that the area of tolerable performance is given by the set $\mathcal{T}$. The area of untolerable performance is excluded from the reach sets. The corresponding outputs are denoted by $Y(\mathcal{R}_{N_k})$ and $Y(\mathcal{R}_{F_k})$. If the set $\Delta_k = (Y(\mathcal{R}_{N_k}) \cup Y(\mathcal{R}_{F_k})) \backslash (Y(\mathcal{R}_{N_k}) \cap Y(\mathcal{R}_{F_k}))$ is not empty then there exist distinguishable executions in the time interval $[0, k]$. The set Δ_k is called the *discriminating set*. Now, there are two possible ways to determine the condition of the system. Assume that at time $k = K_{max}$ the discriminating set $\Delta_{K_{max}} \neq \emptyset$. It can be assumed that the system at time 0 is in the Normal condition. We choose a point which uniquely belongs to the future behaviour of the normal system i.e $\tilde{y}(K_{max}) \in (\Delta_{K_{max}} \cap Y(\mathcal{R}_{N_{K_{max}}}))$. After choosing the point, the optimal input to reach $\tilde{y}(K_{max})$ is computed and applied to the system. If $y(K_{max}) = \tilde{y}(K_{max})$ then system is in the normal condition otherwise it is in the faulty condition. Since the termination of the algorithm is not guaranteed, K_{max} may not exist. For practical applications a bound β on K_{max} is set. If the algorithm does not terminate after β steps, it is recognized as indiagnosable by this method.

Table 5.1: Active Fault Diagnosis

```
Algorithm 1
Given $x_0, \beta, \Sigma_N, \Sigma_F, (\Sigma_N \neq \Sigma_F)$
Find condition $\kappa$
$k = 0, I = x_0, \mathcal{R}_{N_0} = \mathcal{R}_{F_0} = x_0$
Repeat
  $\mathcal{R}_{N_k} = Reach(\Sigma_N, \mathcal{R}_{N_{k-1}}, U)$
  $\mathcal{R}_{F_k} = Reach(\Sigma_F, \mathcal{R}_{F_{k-1}}, U)$
  $\mathcal{R}_{N_k} = \mathcal{R}_{N_k} \cap \mathcal{T}$
  $\mathcal{R}_{F_k} = \mathcal{R}_{F_k} \cap \mathcal{T}$
  $I = Y(\mathcal{R}_{N_k}) \backslash Y(\mathcal{R}_{F_k})$
  $k = k + 1$
Until $(I \neq \emptyset \vee k > \beta - 1)$
$K_{max} = k$
IF $I = \emptyset$
  The fault $F$ is undiagnosable
Else
  Solve the optimization problem
  $\min_{\mathbf{u}_{K_{max}}} J(\mathbf{x}_{K_{max}}, \mathbf{u}_{K_{max}}, \mathbf{y}_{K_{max}})$
  s.t. $\begin{cases} \Sigma_N \\ x_o = x_0, y_f \in Y(I) \end{cases}$
  Apply $\mathbf{u}_{K_{max}}$ to the system
  IF $y_{K_{max}} \in Y(I)$ Then $\kappa = N$ Else $\kappa = F$
```

Another strategy is to assume that $\kappa(0) = F$ and choose $\tilde{y}(K_{max}) \in (\Delta_{K_{max}} \cap Y(\mathcal{R}_{F_{K_{max}}}))$. If $y(K_{max}) = \tilde{y}(K_{max})$ then the system is in faulty condition otherwise it is the normal condition. In Algorithm 1, the first strategy is chosen.

In the case of linear systems, having the convex polyhedral of $\mathcal{X}(0), \mathcal{U}$, the reach set can be computed as:

$$Reach(\Sigma, \mathcal{X}(0), \mathcal{U}) = A\mathcal{X}(0) \oplus B\mathcal{U}, \tag{5.1}$$

where $\oplus$ is the geometric or Minkowski sum. The first part considers the effect of the autonomous part of the system, $x(k+1) = A_q x(k)$, which is computed as mapping of the convex set $\mathcal{X}(0)$ through the matrix A. Because the mapping of a convex set by a linear transformation yields a convex set, the resulting set is also convex. Similarly, in the second part of (5.1) the effect of input is computed by mapping the set $\mathcal{U}$ by matrix B which again results in a convex set. Finally, the reach set is computed as the Minkowski sum of these two sets. For computational reasons, the representation used for the reach set and input set consists of sets which are closed under linear transformation and Minkowski sum such as polytopes, ellipsoids or zonotopes [15].

In the case of hybrid systems, as is shown in Algorithm 2 in Table. 5.2, enabled transitions should also be considered and the corresponding jump functions should be applied. Note that in general the reach set could be nonconvex and disconnected *i.e.* a finite union of p disconnected convex polytopes. In this case it is enough to apply the above algorithm to each polytope separately and at the end calculate the union of results.

The cost function $J(\mathbf{x}_k, \mathbf{u}_k, \mathbf{y}_k)$ is the same as the cost function for the controller, which can have the following from:

$$\Sigma_{k=0}^{K_{max}} \|y(t+k) - r(k)\| + \|u(t+k) - u_r(k)\| + \|x(t+k) - x_f\|,$$

where $r(k)$ is the output reference signal, $u_r(k)$ the input reference signal and x_f is the final desired state.

In the above formulation we have assumed that the system is in the normal condition and therefore the optimization problem is solved by constraining the variables to the hybrid dynamic of the normal system Σ_N. In the case which the system is in the faulty condition and it is not possible to remain in the area of required performance, it is required that the system will remain in a region of tolerable performance. Suppose that the area of tolerable performance is described by the polytope $\mathcal{T} = \{x \in \mathbb{R}^n | \mathcal{P}x \leq \mathcal{M}\}$.To ensure that system states will still remain in the polytope of the tolerable performance, the following constraints should be added to the optimization problem: $\{\mathcal{P}x(i) \leq \mathcal{M}\}_{i=k+1}^{k+K_{max}}$.

Since we have supposed that there exist an observer that gives us the current state at each time, this new information can be used in the algorithm. Suppose that at time $k-1$ the algorithm starts with x_{k-1}. At time k, the information that the diagnoser is using for predicting the behaviour of the system at time $k+1$ is the polytope $Reach_k(\Sigma, x_{k-1}, U)$. While the information from the observer for the current state is more exact. So based on this information, the diagnoser can predict the future behaviour of the system more precisely. Therefore, the overall algorithm can be described as follows. At each time step the output of the observer is given as the input to the main algorithm as described in Algorithm 1. When the optimal input sequence $u(k), \ldots, u(k+K_{max})$ is computed only the first element,$u(k)$, is applied to the system. The overall procedure repeats until $K_{max} = 1$, which means that only in one step it is possible to find the point that uniquely

Table 5.2: Reach Set Computation

```
Algorithm 2
Given $\mathcal{H}, \mathcal{R}_k, \mathcal{U}$,
$\mathcal{R} = \emptyset, \mathcal{R}_G = \emptyset$
$Q_R = \{q | (q, x) \in \mathcal{R}_k \}$
For all $q \in Q_R$
  $X = \{x | (q, x) \in \mathcal{R}_k\}$
  $X_q = X \cap Inv(q)$
  $\mathcal{R}_X = A_q X_q \oplus B_q \mathcal{U}$
  $E_q = \{e | e = (q, q') \in E\}$
  For all $e \in E_q$
    $G_e = X \cap g(e)$
    $G_{e_{trans}} = execute\_transition(G_e)$
    $\mathcal{R}_{G_e} = A_{q'} G_{e_{trans}} \oplus B_{q'} \mathcal{U}$
    $\mathcal{R}_G = \mathcal{R}_G \cup \mathcal{R}_{G_e}$
  End
  $\mathcal{R} = \mathcal{R} \cup \mathcal{R}_G \cup \mathcal{R}_{G_e}$
End
$Reach_{k+1}(\mathcal{H}, \mathcal{R}_k, \mathcal{U}) = \mathcal{R}$
```

belongs to the normal predicted behaviour of the system. The diagnoser applies the optimal input to the system and then the status of the system can be determined. The modified version of the algorithm is more computationally demanding but it can diagnose the fault faster because it also uses available information from the observer.

3.2 Extension for more than one faulty mode

The algorithm can be extended as follows when there is more than one faulty mode. At first, the algorithm tries to choose a state that its corresponding output uniquely belongs to one of the sets $Y(\mathcal{R}_{\Sigma_\kappa k}), \kappa \in K$. Then the optimal input for driving the system to the chosen state is applied to the system. If the system could reach the target state then it is in the condition κ. Otherwise if the current output is consistent with just one of the modes then the corresponding condition is the system condition. But if it is consistent with more than one mode, then the same procedure should be repeated for these modes starting from the new initial condition.

4 Example

The proposed method is tested on the two tank system depicted in Fig. 5.2. The system consists of two cylindrical tanks with cross sectional area A. These two tanks are connected together by two pipes at the bottom and at level h_v. The flows through the pipes, denoted by $Q_{12}V_{12}$ and $Q_{12}V_1$, are controlled using two on/off valves V_{12} and V_1. There is a flow Q_1 through a pump to tank 1 which is a continuous input. Dynamical equations

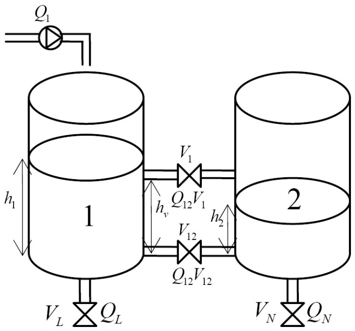

Figure 5.2: Two-tank system

of the system is described as follows.

$$\dot{h}_1 = \frac{1}{A}(Q_1 - Q_{12}V_{12} - Q_{12}V_1 - Q_L), \tag{5.2}$$

$$\dot{h}_2 = \frac{1}{A}(Q_{12}V_{12} + Q_{12}V_1 - Q_N), \tag{5.3}$$

where h_1 and h_2 denote the levels of tanks 1 and 2 respectively. The flow $Q_{12}V_{12}$ is described as:

$$Q_{12}V_{12} = V_{12}k_{12}sign(h_1 - h_2)\sqrt{2g\,|h_1 - h_2|},$$

where g is the gravity constant and k_{12} is a constant. similarly $Q_L = V_L k_L \sqrt{2gh_1}$ and $Q_N = V_N k_N \sqrt{2gh_2}$. The flow trough valve V_1 is described by:

$$\begin{aligned} Q_{12}V_1 &= V_1 k_1 sign(max\{h_v, h_1\} - max\{h_v, h_2\}) \\ &\sqrt{|2g(max\{h_v, h_1\} - max\{h_v, h_2\})|} \end{aligned}$$

In order to apply the reach set computation algorithm to the above system, the dynamic of the system should be described as a discrete time linear hybrid system. This task is done in three steps. First, four discrete modes corresponding to four combinations of binary inputs are generated. In each of these modes the governing equations are obtained by substituting the corresponding values of binary inputs. The system switches between these four discrete modes based on the binary input vector $V = [V_{12}, V_1]$. Then, the nonlinear relation $\sqrt{x}$ is approximated by a straight line x. The resulting equations are piecewise affine. Finally, differential equations 5.2, 5.3 are discretized in time by Euler approximation $\dot{h}_i(t) \approx \frac{h_i(t+1)-h_i(t)}{Ts}$, where T_s is sample time.

To compute $\mathcal{R}_k$ from $\mathcal{R}_{k-1}$, all possible binary and continuous inputs must be considered. Algorithms 2 considers all possible continuous inputs. To consider the effect of all possible binary inputs, for every corresponding discrete mode, the reach set is computed via algorithm 2 and $\mathcal{R}_k$ is obtained by calculating the union of the results.

The proposed active fault diagnosis algorithm is used for sanity check of the valve V_1. A stuck ON fault is considered in V_1 and the algorithm is used to generate the shortest test signal sequence to diagnose this fault. Nine different scenarios as shown in Table. 5.3 are considered. In each scenario, a binary input is used or fixed during the diagnosis to 0 or 1.

Fig. 5.3 and 5.4 show the results for scenario 1 where both discrete inputs are used. In order to make the difference between $Y(\mathcal{R}_{N_k})$ and $Y(\mathcal{R}_{F_k})$ observable, the discriminating set in algorithm 1 is changed to $I = Y(\mathcal{R}_{N_k}) \backslash (Y(\mathcal{R}_{F_k}) \oplus \mathcal{B}(0,d))$, where $\mathcal{B}(0,d)$ is a box defined as $\mathcal{B}(0,d) = \{x \in \mathbb{R}^2 | 0 \leq x_i \leq d\}$. The algorithm terminates after $k = 5$ steps. $Y(\mathcal{R}_{N_5})$ and $Y(\mathcal{R}_{F_5}) \oplus \mathcal{B}(0, 0.01)$ are shown in Fig. 5.3. The set I consists of two polytopes shown in Fig. 5.4. One of these polytopes(the grey one here) is considered as the target set and then the input to reach the target set is computed and applied to the system. The resulting output and the expected output of the system are depicted in Fig. 5.4. As it can be seen the result of the diagnosis algorithm is that the system is faulty.

As it is shown in Table 5.3, scenarios 4, 5, 6 are not applicable. Because V_1 is fixed as 1 and therefore the model of the normal system becomes exactly the same as the model of the faulty system.Moreover, it shows that using a valve as a free input variable causes more computational complexity than fixing it. The reason is that the main source for the computational complexity of the algorithm is nonconvexity of the reach set which is casued by either crossing a gaurd (h_v here) or a switching input. It should be noted that however using both valves is the most computationally demanding scenario but for this scenario the algorithm will find the shortest input sequence for diagnosis while by fixing valves it may not find the shortest sequence e.g. as it is the case in scenarios 2, 9.

Fig. 5.5 demonstrates the case where the faulty and the normal system exhibit same dynamic behaviour. In this example a model predictive controller is designed for the two tank system. Fig. 5.5 shows the simulation of the closed loop system. As one can see the control variable V_1 is manipulated such that the output of the system in the normal condition and in the faulty one is exactly the same. In this situation if a stuck ON fault

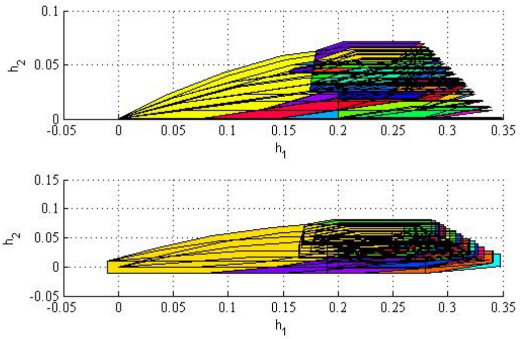

Figure 5.3: Top:Reach set of the normal system at $k = 5$:($\mathcal{R}_{N_5}$), Bottom:Reach set of the faulty system at $k = 5$ added by $\mathcal{B}(0, 0.01)$:($\mathcal{R}_{F_5} \oplus \mathcal{B}(0, 0.01)$).

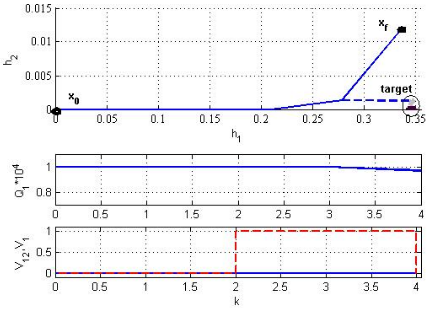

Figure 5.4: Top:Output of the system(solid), expected output of the system (dashed) and discriminating set (target), Middle:continuous input Q_1, Bottom:discrete inputs:V_1(dashed), V_{12}(solid)

happens, no passive diagnoser would be able to diagnose it, while the active diagnoser proposed here is capable of detecting this fault. Our active diagnoser was started at $t =$ 200 sec. and the result is shown in Fig. 5.6.

5 Conclusion and future works

This paper presented an approach for active fault diagnosis of hybrid systems based on reach sets computation of both the normal and the faulty modes. The proposed method can be used for sanity check of the system at the commissioning phase and also periodically during normal operation for faster detection of faults or detection of faults when it is impossible to detect them by a passive diagnoser.

During the diagnosis it is assumed that the system is in the normal mode of the operation. To ensure that if the system is faulty, it will remain in the tolerable performance region, the optimization problem is solved subject to constraints describing polytope of the tolerable area. It might happen that the optimization become infeasible by these con-

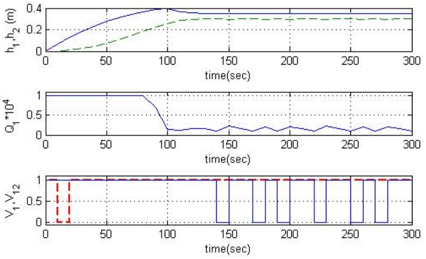

Figure 5.5: Top:output of the closed loop system for both faulty and normal system:h_1(solid) h_2(dashed), Middle:continuous input Q_1, Bottom:discrete inputs: V_1(dashed line), V_{12}(solid line)

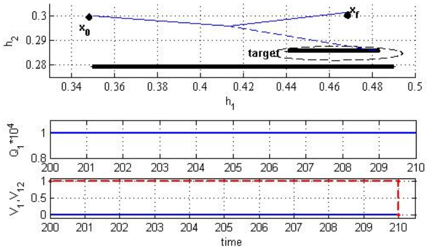

Figure 5.6: Top:Output of the system(solid), expected output of the system and target set, Middle:continuous input Q_1, Bottom:discrete inputs:V_1(dashed line), V_{12}(solid line)

straints. This issue is subject to future investigations.

References

[1] P. Antsaklis and X. Koutsoukos, “Hybrid systems: Review and recent progress,” in *Software-Enabled Control*, T. Samad and G. Balas, Eds. IEEE Press, 2003, pp.

Table 5.3

Scenario	V_1	V_{12}	K_{max}	CPU time (sec)
1	x	x	5	8.9
2	x	1	6	0.87
3	x	0	5	0.71
4	1	x	NA	-
5	1	1	NA	-
6	1	0	NA	-
7	0	x	5	4.5
8	0	0	5	0.37
9	0	1	6	0.54

271–298.

[2] R. Mohammadi, S. Hashtrudi-Zad, and K. Khorasani, “A hybrid architecture for diagnosis in hybrid systems with applications to spacecraft propulsion system,” in *IEEE International Conference on Systems, Man and Cybernetics*, 2007, pp. 3184–3190.

[3] S. Narasimhan and G. Biswas, “Model-based diagnosis of hybrid systems,” *IEEE transactions on man and cybernetics*, vol. 37, no. 3, pp. 347–361, 2007.

[4] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, and P. Cheung, “Monitoring and fault diagnosis of hybrid systems,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, pp. 1225–1240, 2005.

[5] J. Lunze, “Diagnosis of quantized systems by means of timed discrete-event representations,” in *Hybrid systems: Computation and Control*, ser. Lecture Notes in Computer Science, N. Lynch and B. Krogh, Eds., vol. 1790. new york: Springer, 2000, pp. 258–271.

[6] X. Koutsoukos, J. Kurien, and F. Zhao, “Estimation of distributed hybrid systems using particle filtering methods,” in *Hybrid systems: Computation and Control*, ser. Lecture Notes in Computer Science, vol. 2623. Springer, 2003, pp. 298–313.

[7] M. Hofbaur and B. Williams, “Mode estimation of probabilistic hybrid systems,” in *Hybrid systems: Computation and Control*, ser. Lecture Notes in Computer Science, vol. 3927. Springer, 2002, pp. 253–266.

[8] S. Campbell and R. Nikoukhah, *Auxiliary Signal Design for Failure Detection*. Princeton University Press, 2004.

[9] H. H. Niemann, “A setup for active fault diagnosis,” *IEEE Transactions on Automatic Control*, vol. 51, no. 9, pp. 1572–1578, 2006.

[10] I. Andjelkovic, K. Sweetingham, and S. Campbell, “Active fault detection in nonlinear systems using auxiliary signals,” in *American Control Conference*, Seattle, WA, 2008, pp. 2142–2147.

[11] F. Lin, “Diagnosability of discrete events systems and its application,” *Discrete event systems*, vol. 4, pp. 197–212, 1994.

[12] M. Sampath, S. Lafortune, and D. Teneketzis, “Active diagnosis of discrete-event systems,” *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 908–929, 1998.

[13] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*. Springer-Verlag, 2006.

[14] A. Balluchi, L. Benvenuti, M. D. D. Benedetto, and A. Sangiovanni-Vincentelli, “Design of observers for hybrid systems,” in *5th International Workshop on Hybrid Systems: Computation and Control*. London, UK: Springer-Verlag, 2002, pp. 76–89.

[15] A. Girard, C. L. Guernic, and O. Maler, “Efficient computation of reachable sets of linear time-invariant systems with inputs,” in *Hybrid systems: Computation and Control*, ser. Lecture Notes in Computer Science, vol. 3927. Springer, 2006, pp. 257–271.

Paper B

Automatic Sensor Assignment of a Supermarket Refrigeration System

Seyedmojtaba Tabatabaeipour, Roozbeh Izadi-Zamanabadi, Thomas Bak, and Anders P. Ravn

This paper is published in:
the Proceeding of IEEE Multi-Conference on Control Applications, (CCA) & Intelligent Control, (ISIC), pp. 1319-1324

Abstract

Wrong sensor assignment is a major source of faults in industrial systems during the commissioning phase. In this paper a method for automatic sensor assignment based on active diagnosis is proposed. The active diagnosis method is developed for diagnosis of linear hybrid systems. It generates the appropriate test signal which can be used for sanity check at the commissioning phase. It could also be used for faster detection of faults during the normal phase of operation or for detection of faults which are impossible to detect by passive methods because of regulatory actions of the controller. The method is tested on a supermarket refrigeration system.

1 INTRODUCTION

In a large system there are many sensors, actuators and other components. Every measurement from a sensor or output to an actuator should be assigned correctly to its corresponding variable in the control algorithm. Yest, it happens that a technician connects components of a system wrongly. Wrong sensor or actuator assignment potentially results in malfunction of the overall system. Therefore, it is desirable to design a controller which provides a sanity check in the commissioning phase for verifying sensor and actuator assignment by generating an appropriate test signal.

A way to tackle this problem is to consider wrong connections as faults and use fault diagnosis methods. Diagnosis methods can be divided into two main categories: active and passive. In passive diagnosis the diagnoser observes the system and based on the observation decides about the occurance of faults. In active fault diagnosis the diagnoser generates a test signal which excites the system to decide whether the observed system dynamics exhibits the normal behaviour or the faulty behaviour and if feasible decide which faulty behaviour occurs.

Industrial systems typically include both discrete and continuous components and a hybrid system formulation is therefore natural to adopt. Generally speaking, a hybrid system is a dynamical system with both continuous and discrete behaviours and non-trivial interaction between continuous evolutions and discrete transitions. Fault diagnosis of hybrid systems has been investigated recently, for a survey see [1], [2], [3]. Most of the available methods are in the area of passive diagnosis. [4] propose a method for active diagnosis of linear systems using an auxiliary signal for fault detection. The results of [4] are extended to nonlinear systems in [5] using linearization and also a direct optimization approach. A setup for active diagnosis of linear system for parametric faults is proposed by [6]. In [7] and [8] the problem for discrete event systems is investigated. In our previous work [9], we proposed an active fault diagnosis method for diagnosis of linear hybrid systems in discrete time. The method is based on prediction of the behaviour of the system in the future by means of reach set computations based on a faulty and a normal model of the system. If we apply the method directly to the sensor assignment problem, in other words, if we consider all possible assignments as a fault, we need a model for each possible assignments which yields a high computational effort. In this paper we extend the active diagnosis algorithm to the sensor assignment problem such that only one model of the system is necessary. To illustrate the method a supermarket refrigeration system is considered.

2 Preliminaries and Problem formulation

To make our ideas precise we first define the problem and give some preliminary definitions.

2.1 Problem Formulation

Consider a system with n sensors *i.e.* $[S_1, \cdots, S_n]$. The sensor assignment problem is to find among all permutations the one that conforms to the dynamic behaviour of the undelying system. The problem is defined as follows.

Problem 6.1 (Sensor assignment problem)**.** *Given a set of measurements* $\boldsymbol{y} = [y_1, \cdots, y_n]$ *representing measurements from* $[S_1, \cdots, S_n]$ *and a model of the system as* $\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k), \boldsymbol{u}(k))$, $[\hat{y}_1(k), \cdots, \hat{y}_n(k)]' = h(\boldsymbol{x}(k), \boldsymbol{u}(k))$. *Find a permutation of* $\boldsymbol{y}$ *namely* $\boldsymbol{V}$ *such that for a large* N*, for all* i

$$\Sigma_{k=1}^{N} |V_i(k) - \hat{y}_i(k)| < \Sigma_{k=1}^{N} |V_j(k) - \hat{y}_i(k)| \tag{6.1}$$

for all $j \in 1, \cdots, n, j \neq i.\square$

2.2 Preliminaries

Definition 6.1 (Hybrid Automaton)**.** A *hybrid automaton,* $\mathcal{H}$ is a collection $\mathcal{H} = (Q, X, U, Y, Init, f, h, Inv, E, G, J)$ where,

- Q is a set of finite discrete modes, $Q = \{q_1, q_2, \ldots, q_m\}$,
- X is a finite set of continuous state variables,
- U is a finite collection of input variables,
- Y is a finite collection of output variables,
- $Init \subset Q \times X$ is a set of initial states,
- $f : Q \times X \times U \to \mathbb{R}^n$ is a vector field,
- $h : Q \times X \times U \to Y$ is an output map,
- $Inv : Q \to 2^{X \times U}$ assigns to each $q \in Q$ an invariant set $Inv(q) \subseteq X \times U$,
- $E \subset Q \times Q$ is a set of discrete transitions,
- $G : E \to 2^{X \times U}$ assigns to each $e = (q, q') \in E$ a guard $g(e) \subset X \times U$,
- $J : E \times X \times U \to 2^X$ is a jump function that assigns a jump set $J(e, x, u) \subseteq X \times U$ to each pair $e \in E$ and $x \in g(e).\square$

In the case of discrete time linear hybrid systems the vector field f_q is represented by a linear difference equation: $x_{i+1} = A_{q_i} x_i + B_{q_i} u_i$ and the output map is of the form $y_{i+1} = C_{q_i} x_i + D_{q_i} u_i$. We refer to $(u, y) \in U \times Y$ as an observation of $\mathcal{H}$.

An execution of a hybrid automaton is a sequence $\chi = (\sigma_0, \dots, \sigma_i, \sigma_{i+1}, \dots)$ where $\sigma_0 = (q_0, x_0, u_0, y_0)$, $\sigma_i = (q_i, x_i, u_i, y_i)$ and $\sigma_{i+1} = (q_{i+1}, x_{i+1}, u_{i+1}, y_{i+1})$ which satisfies the discrete and continuous evolution constraints imposed by hybrid automata and σ_0 satisfies the initial condition [9].

Both discrete faults and continuous faults are modeled as a mode in hybrid automata as in [1]. It is supposed that events that describe transitions from a normal mode to a faulty mode are unobservable. The system can be in a normal condition N or a faulty condition F where each condition is a subset of Q. A condition set $K = \{N, F_1, \dots, F_p\}, p \geq 1$ is a set of conditions that constitutes a complete partition of the mode set Q. For every condition $\kappa \in K$, the corresponding dynamical system, Σ_κ, is denoted by:

$$\Sigma_\kappa = \{\kappa, X, U, Y, Init, f, Inv, E_\kappa, G, J\}$$

where $E_\kappa = \{e = (q, q') \,|q \in \kappa, q' \in \kappa\}$ and $Init_\kappa \subset \kappa \times X$.

3 The proposed algorithm

In this section active fault diagnosis is described firstly and then it is explained that how the sensor assignment problem can be solved by extending the proposed algorithm.

3.1 Active Diagnosis

We are going to solve Problem 1 by means of an active diagnoser which generates a test signal in the commissioning phase for finding the true sensor assignment. A diagnoser is a system that gives us an estimate $\hat{\kappa}(k)$ of the current system condition $\kappa(k)$. A passive diagnoser recieves a sequence of observations as input and generates an estimate of the current condition $\hat{\kappa}(k)$ as output. In active diagnosis an input sequence $\langle u(k+1), \dots, u(k+m) \rangle$ is generated by the diagnoser and applied to the system. The resulting output sequence $\langle y(k+1), \dots, y(k+m) \rangle$ is observed by the diagnoser to determine the system condition. The active diagnosis problem is defined as follows:

Problem 6.2 (Active diagnosis problem)**.** *Given a hybrid automaton $\mathcal{H}$, Find a sequence of inputs $\langle u(0), \dots, u(m) \rangle$ such that the condition $\kappa(0)$ is determined by observing the sequence $\langle y(0), \dots, y(m) \rangle$.*

If the input sequence exists, *i.e.* if the system is diagnosable, we can look for the optimal solution, where optimality can be interpreted in different senses. The proposed algorithm looks for the shortest sequence of inputs that can diagnose the system.

A model-based passive diagnoser usually checks the consistency of the I/O pair with the expected behaviour of system based on a given model. If the consistency is verified, the system is in the normal mode otherwise it is faulty. Now consider Fig. 6.1. The set $\mathbf{B}_0$ represents the normal behaviour of the system and the set $\mathbf{B}_1$ represents the behaviour of the system subject to the fault f_1. As long as the observed I/O pair is uniquely in the set $\mathbf{B}_0$ or $\mathbf{B}_1$, such as point A or B, the diagnoser can detect whether the system is faulty or not. But for a point such as C which belongs to the intersection of $\mathbf{B}_0$ and $\mathbf{B}_1$ it is impossible to detect the mode of the system. The idea here is to exert an input signal to the system to move C to an area which belongs uniquely either to the set $\mathbf{B}_0$ or $\mathbf{B}_1$.

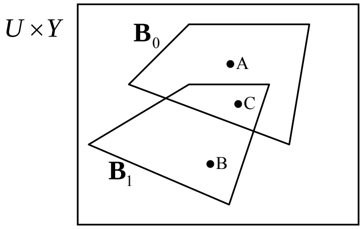

Figure 6.1: System behaviour

Given a model of the normal and the faulty system, from the current state we predict all possible behaviour that each model of the system can present in the next step considering all possible inputs. This task is repeated as long as the predicted behaviour of the fautly and the normal model are the same. As soon as they become different, we find the set holding these different behaviours. We choose one of them, e.g. belonging to the future behaviour of the normal system. Then we find an optimal input sequence that will drive the system to a state corresponding to the selected behaviour and apply it to the system. If the output of the system reaches the corresponding output of the selected behaviour, then the system is in the normal mode otherwise it is faulty.

It is supposed that the initial state of the system is given by an observer-based passive diagnoser as proposed by [10]. The diagnoser consists of two parts: mode observer and countinuous observer. If the current state of the system, $(q(k), x(k))$, is determined uniquely then the condition is also determined. A problem arises when both the faulty mode and the normal mode are recognized as consistent with the I/O sequence. A mode is consistent with the I/O sequence when the corrresponding element in the residual vector $\rho = \{r_1, \dots r_m\}$ generated by the mode observer is zero. Consistency of two modes with the I/O sequence means that they have indistinguishable executions. Two executions are called indistinguishable in a time interval if their corresponding continuous output in that time interval are identical.

3.2 The proposed algorithm

In this subsection the proposed algorithm for one faulty mode is described. In [9] it is explained how to expand the algorithm to more than one faulty mode.

The algorithm looks for two distinguishable executions χ_1 and χ_2 respectively from the system in normal condition, Σ_N, and the faulty system, Σ_F. In order to accomplish this task, all possible outputs that both systems could reach in the future time steps considering all admissible inputs and starting from the given initial state is computed which is equal to reach set computation.

Definition 6.2 (Reach Set). *Reach Set* of a hybrid automata $\mathcal{H}$ at time k denoted by $Reach_k(\mathcal{H}, \mathcal{X}(0), \mathcal{U})$ is the set of all states $(q, x) \in Q \times X$ that are reachable by a given hybrid automata $\mathcal{H}$ at time step k, starting from any initial state $x(0) \in \mathcal{X}(0)$ and with all possible inputs $u \in \mathcal{U}$.

As soon as the corresponding outputs of the reach sets of the system based on the normal and the faulty model of the system becomes different the algorithm terminates.

Table 6.1: Active Fault Diagnosis

```
**Algorithm 1**
**Given** $x_0, \beta, \Sigma_N, \Sigma_F, (\Sigma_N \neq \Sigma_F)$
**Find** condition $\kappa$
$k = 0, I = x_0, \mathcal{R}_{N_0} = \mathcal{R}_{F_0} = x_0$
**Repeat**
  $\mathcal{R}_{N_k} = Reach(\Sigma_N, \mathcal{R}_{N_{k-1}}, U)$
  $\mathcal{R}_{F_k} = Reach(\Sigma_F, \mathcal{R}_{F_{k-1}}, U)$
  $\mathcal{R}_{N_k} = \mathcal{R}_{N_k} \cap \mathcal{T}$
  $\mathcal{R}_{F_k} = \mathcal{R}_{F_k} \cap \mathcal{T}$
  $I = Y(\mathcal{R}_{N_k}) \backslash Y(\mathcal{R}_{F_k})$
  $k = k + 1$
**Until** $(I \neq \emptyset \vee k > \beta - 1)$
$K_{max} = k$
**IF** $I = \emptyset$
  The fault $F$ is undiagnosable
**Else**
  **Solve the optimization problem**
  $\min_{\mathbf{u}_{K_{max}}} J(\mathbf{x}_{K_{max}}, \mathbf{u}_{K_{max}}, \mathbf{y}_{K_{max}})$
  s.t. $\begin{cases} \Sigma_N \\ x_o = x_0, y_f \in Y(I) \end{cases}$
  **Apply** $\mathbf{u}_{K_{max}}$ **to the system**
  **IF** $y_{K_{max}} \in Y(I)$ **Then** $\kappa = N$ **Else** $\kappa = F$
```

In Algorithm 1, reach sets of the normal system and the faulty system at time k are respectively denoted by $\mathcal{R}_{N_k}$ and $\mathcal{R}_{F_k}$ and the area of tolerable perfomance is denoted by the set $\mathcal{T}$. The area of tolerable performance is defined by the minimum level of control objectives and system constraints, which are required to maintain safe operation. At each time step $\mathcal{R}_{N_k}$ and $\mathcal{R}_{F_k}$ are computed. To ensure that the solution found by the algorithm does not include any intolerable performance, the area of intolerable performance is excluded from the reach sets. The corresponding outputs are denoted by $Y(\mathcal{R}_{N_k})$ and $Y(\mathcal{R}_{F_k})$. If these two sets are not exactly the same or in other words if the set $\Delta_k = (Y(\mathcal{R}_{N_k}) \cup Y(\mathcal{R}_{F_k})) \backslash (Y(\mathcal{R}_{N_k}) \cap Y(\mathcal{R}_{F_k}))$ is not empty then there exist distinguishable executions in the time interval $[0, k]$. The set Δ_k is called the *discriminating set*. As soon as the discriminating set becomes nonempty the algorithm proceeds to the next step which is determining the system condition.

To determine the system condition we need to make a hypothesis about it at time 0. If we assume that the system at time 0 is in the Normal condition, as it is assumed in algorithm 1, to test this hypothesis, the algorithm chooses a point which uniquely belongs to the future behaviour of the normal system i.e $\tilde{y}(K_{max}) \in Y(\mathcal{R}_{N_{K_{max}}}) \backslash Y(\mathcal{R}_{F_{K_{max}}})$ where $k = K_{max}$ shows the first time that the discriminating set becomes nonempty. After choosing the point, the optimal input to reach $\tilde{y}(K_{max})$ is computed and applied to the system. If $y(K_{max}) = \tilde{y}(K_{max})$ then the hypothesis is verified and the system is in the normal condition otherwise it is in the faulty condition. Fig. 6.2 illustrates the algorithm.

Since the termination of the algorithm is not guaranteed, for practical applications a bound β on K_{max} is set. If the algorithm does not terminate after β steps, it is recognized as indiagnosable by this method.

The above results are valid only if the reach set at time k is computed from the initial state without any uncertainty. But suppose that the initial state is given in the set $\mathcal{X}(0)$, then two different cases should be considered. In the first case, if all the states in the obtained reach sets $\mathcal{R}_{N_{K_{max}}}$ and $\mathcal{R}_{F_{K_{max}}}$ are reachable from the initial set $\mathcal{X}(0)$ within K_{max} sampling time then the previous result is hold, in other words, $\Delta_k = (Y(\mathcal{R}_{N_k}) \cup Y(\mathcal{R}_{F_k}))\backslash(Y(\mathcal{R}_{N_k}) \cap Y(\mathcal{R}_{F_k}))$. Checking the reachability condition for hybrid systems is not simple. In the second case, if the reachability condition does not hold then the conservative approach is to check when the two reach sets $Y(\mathcal{R}_{N_K})$ and $Y(\mathcal{R}_{F_K})$ are totally distinct from each other *i.e.* $Y(\mathcal{R}_{N_K}) \cap Y(\mathcal{R}_{F_K}) = \emptyset$. When this condition is satisfied the algorithm must terminate and $\Delta_k = Y(\mathcal{R}_{N_k}) \cup Y(\mathcal{R}_{F_k})$.

To find the optimal input, the following cost function is used:

$$J(\mathbf{x}_k, \mathbf{u}_k, \mathbf{y}_k) =$$

$$\Sigma_{k=0}^{K_{max}} \|y(t+k) - r(k)\| + \|u(t+k) - u_r(k)\| + \|x(t+k) - x_f\|,$$

where $r(k)$ is the output reference signal, $u_r(k)$ the input reference signal and x_f is the final desired state.

Two groups of constraints are applied in the optimization. The first one is that the state variables should evolve based on the dynamic of the system which is dependent on our hypothesis. The second group ensures that the system remains in the area of tolerable performance for the situation that our hypothesis was wrong and the system is actually faulty. Suppose that the area of tolerable performance is given by the polytope $\mathcal{T} = \{x \in \mathbb{R}^n | \mathcal{P}x \leq \mathcal{M}\}$. To ensure that system states will remain in $\mathcal{T}$, the following constraints should be added to the optimization problem: $\{\mathcal{P}x(i) \leq \mathcal{M}\}_{i=k+1}^{k+K_{max}}$.

For a linear systems the reach set can be computed as:

$$Reach(\Sigma, \mathcal{X}(0), \mathcal{U}) = A\mathcal{X}(0) \oplus B\mathcal{U}, \tag{6.2}$$

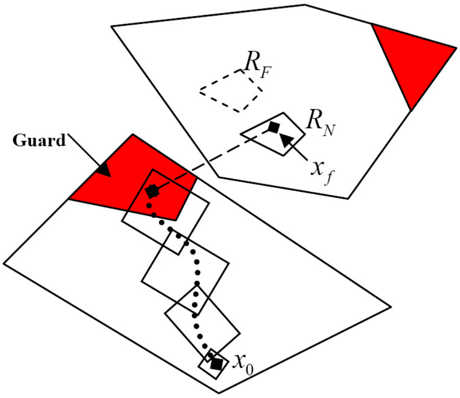

Figure 6.2: Active diagnosis method

where $\mathcal{X}(0), \mathcal{U}$ denote the convex polyhedra of the initial state and the input respictvely and $\oplus$ is the geometric or Minkowski sum. For computational effciency the representation used for the reach set and input set consists of sets which are closed under linear transformation and Minkowski sum such as polytopes, ellipsoids or zonotopes [11]. In the case of linear hybrid systems enabled transitions and the corresponding jump functions should be considered. The reach set computation is descibed with more details in [9].

3.3 Sensor Assignment

Wrong assignment of sensors can be considered as a fault, and it can be modelled as a permutation of the output vector. The problem is to find among all permutations the one that is consistent with the dynamic behaviour of system. Consider a system with n sensors. There are $n!$ candidate assignments or in another words $n! - 1$ fault hypothesis. If we use algorithm 1 directly, it will be computationally very expensive. The method proposed here only needs one model of the system. It is supposed that the initial state of the system is given such that the outputs are indistinguishable, i.e. $y_i = y_j, i, j \in 1, \cdots, n, i \neq j$. In order to simplify the explanation, the idea is described for a system with two sensors. We assume that as long as $|y_1 - y_2| < \epsilon$ outputs can not be distinguished. If we excite the system such that as its result $y_1 > y_2 + \epsilon$ or $y_2 > y_1 + \epsilon$ then they can be distinguished. Therefore as before we compute future reach sets of the system. As soon as the corresponding output set goes outside the region $|\hat{y}_1 - \hat{y}_2| < \epsilon$ the algorithm terminates. A state correspondent to a point in $Y(\mathcal{R}_{K_{max}}) \cap (|\hat{y}_1 - \hat{y}_2| > \epsilon)$ is chosen. Any point in this set exibits an order between its elements *i.e.* $\hat{y_1} > \hat{y_2}$ or $\hat{y_2} > \hat{y_1}$. A point in this set is chosen. We find the input for leading the system to the chosen point and apply it to the system. By comparing the order in the elements of the output vectors and the predicted orders between elements of $[\hat{y}_1, \hat{y}_2]$ we can find the correct assignment. For example, if a point in $\hat{y_2} > \hat{y_1}$ is chosen and the observed output presents the following order $y_1(K_{max}) > y_2(K_{max})$ then S_1 should be assigned to the variable $\hat{y}_2$ and S_2 to $\hat{y}_1$. If there are more than two sensors the strategy is the same. The algorithm looks for an area where the outputs present an order which is $Y(\mathcal{R}_{K_{max}}) \cap (|y_i - y_j| > \epsilon, 0 \leq i, j \leq n, i \neq j)$. The system is then driven to that area. By comparing the predicted order and the obseved order the assignment is accomplished.

4 System Description

In a supermarket, for customer's convenience, goods are usually placed in an open display case in a refrigerator. Fig. 6.3 shows a supermarket refrigeration system with two display cases. The system consists of five main parts, namely liquid manifold, display cases, suction manifold, compressor and condenser. The refrigerant in the liquid manifold is in the liquid phase. It is led into the evaporators inside the display cases through inlet valves. The compressor keeps the evaporator temperature at a certain level by keeping the pressure in the suction manifold at a constant pressure. The refrigerant removes heat from goods while evaporating in the evaporators and transforming into low pressure gas. The low pressure refrigerant is compressed in the compressor rack. The refrigerator circuit is closed by feeding back the liquid refrigerant from the condenser to the liquid manifold.

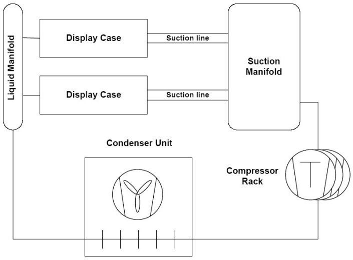

Figure 6.3: A Simplified Supermarket Refrigeration System

Fig. 6.4 shows an schematic illustration of the measurements and control instrumentation in a typical display case used in a supermarket refrigeration system. An air flow is circulating through the evaporator. The refrigerant is led into the evaporator through an on/off inlet valve and evaporates while absorbing the heat from the surrounding. The circulating air flow creates a cold air curtain at the front of the display case. Since the air curtain is colder than the goods and the surroundings, it absorbs the heat from the goods $(Q_{goods-air})$ and the surroundings $(Q_{airload})$. The absorbed heat is transferred through the evaporator wall to the evaporator $(Q_{air-wall})$.

5 The Hybrid Model of the System

The hybrid model we use is based on the model proposed in [12].

5.1 Evaporator

The dynamic of the evaporator is obtained by writing energy balance equations:

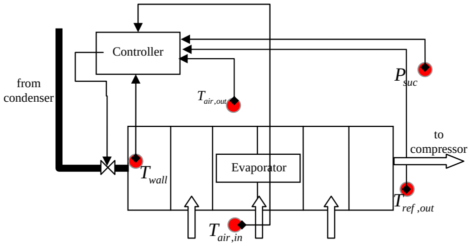

Figure 6.4: An evaporator and its instrumentation

$$\frac{dT_{air,in}}{dt} = \frac{\dot{Q}_{goods-air} + \dot{Q}_{airload} - \dot{Q}_{air-wall}}{M_{air}Cp_{air}} \tag{6.3}$$

$$\frac{dT_{wall}}{dt} = \frac{\dot{Q}_{air-wall} - \dot{Q}_{e}}{M_{wall}Cp_{wall}} \tag{6.4}$$

$$\frac{dT_{goods}}{dt} = \frac{-\dot{Q}_{goods-air}}{M_{goods}Cp_{goods}} \tag{6.5}$$

Moreover,

$$\dot{Q}_{air-wall} = UA_{air-wall}(T_{air} - T_{wall}) \tag{6.6}$$

$$\dot{Q}_{e} = UA_{wall-ref}(M_{ref})(T_{wall} - T_{e}) \tag{6.7}$$

$$\dot{Q}_{goods-air} = UA_{goods-air}(T_{goods} - T_{air}) \tag{6.8}$$

$$UA_{wall-ref} = UA_{wall-ref,max}\frac{M_{ref}}{M_{ref,max}} \tag{6.9}$$

$$T_{air,in} - T_{air,out} = \frac{\dot{Q}_{air-wall}}{\dot{m}_{air}Cp_{air}}, \tag{6.10}$$

where M denotes the mass, Cp the heat capacity and UA the overall heat transfer coefficient with the subcript denoting the media between which the heat is transferred. T_e shows the evaporation temperature which is a refrigerant dependant function of the evaporation pressure P_e. Here it is assumed that there is no pressure drop in the suction line and therefore the suction pressure P_{suc} is equal to the evaporation pressure.

It is assumed that the evaporator will be filled or emptied abruptly as the valve is opened or closed respectively. Consequently, the value of the mass of refrigerant ,M_{ref}, switches between 0 and $M_{ref,max}$.

5.2 The Suction Manifold

The dynamic of the suction pressure is described by

$$\frac{dP_{suc}}{dt} = \frac{\dot{m}_{in-suc} + \dot{m}_{ref-const} - \dot{V}_{comp} \cdot \rho_{suc}}{V_{suc}\frac{d\rho_{suc}}{dP_{suc}}} \tag{6.11}$$

where V_{suc} is the volume of the suction manifold, $\dot{V}_{comp}$ is the volume flow from the suction manifold to the compressor and $\dot{m}_{in-suc}$ is the total mass flow from the evaporator to the suction manifold which is given by

$$\dot{m}_{in-suc} = \Sigma_{i=1}^{n}\frac{\dot{Q}_{e},i}{\Delta h_{lg}}, \tag{6.12}$$

where n is the number of the display cases. $\dot{m}_{ref-const}$ is a constant disturbance representing mass flow from other unmodelled refrigerator entities. ρ_{suc} represents the density of the vapor in the suction manifold which is a nonlinear refrigerant-dependent function of P_{suc}.

5.3 The Compressor

A number of compressors working in parallel that can be switched on or off separately constitute the entire compressor capacity. The entire volume flow out of the suction manifold is described by $\dot{V}_{comp} = \Sigma_{i=1}^{q} \dot{V}_{comp,i}$, where $\dot{V}_{comp,i}$ is the volume flow created by one compressor which is given by

$$\dot{V}_{comp,i} = \frac{comp_i.\eta_{vol}.V_{sl}}{100} \qquad i = 1, \cdots, q, \tag{6.13}$$

where $comp_i$ denotes the capacity of the i'th compressor, q is the number of compressors, η_{vol} is the constant volumetric efficiency and V_{sl} is the total displacement volume.

5.4 The Overall Model

Putting together the above subsystems we get the overall dynamical model of the system. Each display case has three states, namely $T_{air,in}, T_{goods}, T_{wall}$ and the suction manifold has one state which is P_{suc}. Measured variables are $T_{air,in}, T_{wall}, T_{air,out}, T_e$. Inputs of the system are the evaporator inlet valves and comppressors valves. These valves are considered as on/off valves and therefore the overall model of the systems represents a hybrid dynamic.

In order to apply our method to this system we need a linear hybrid dynamical model of it. Therefore nonlinearities such as the dependency of T_e and ρ_{suc} on P_{suc} in equations 6.7, 6.11 are substituted by linear approximations of them.

6 Simulation Results

The sensor assignment algorithm is tested on the refrigeration system for assignment of the wall and input air temperature sensors to $[T_{air,in}, T_{wall}]$. Because we are considering the commissioning phase there is no goods inside the display case and therefore $\dot{Q}_{goods-air} = 0$ and T_{goods} is not a state. It is assumed that the initial states are in the polytope $\{14 \leq T_{air,in} \leq 16, 14 \leq T_{wall} \leq 16, 1 \leq P_{suc} \leq 3\}$. Also $\dot{Q}_{airload}$ is considered as a disturbance and is assumed to vary between 1500 and 4500. We have used $T_s = 2$ as sampling time for discretization and $\epsilon = 1$.

To consider the effect of all possible binary inputs, for every corresponding discrete mode the reach set is computed via algorithm 1 and $\mathcal{R}_k$ is obtained by calculating the union of the results. Because of uncertainties due to the initial states given as a polytope and $\dot{Q}_{airload}$ considered as disturbance, as explained in section III, we should either check the reachability condition or consider the conservative solution. Here the conservative solution is considered. Consider the reach set at time k. Because of the switching effect of the binary inputs it is a union of p polytopes $\mathcal{R}_k = \cup_{i=1}^{p} \mathcal{P}_i$. We can not say that every state in $\mathcal{R}_k$ is reachable but we know that a state in $\mathcal{P}_i$ is reachable by choosing the corresponding sequence of binary inputs. Therefore, for termination of the algorithm it is enough to check whether there exist a $\mathcal{P}_i$ in $\mathcal{R}_k$ such that its intersection with $|y_1 - y_2| < \epsilon$ is empty. It happens at $k = 4$ and the reach set is depicted in Fig. 6.5.

Fig. 6.6 shows the initial states, the target polytope and the observed and predicted output. Comparing the expected order and the observed order the assignment is
$(y_2, T_{air,in}), (y_1, T_{wall})$. The obtained input sequence for both valves is $[1, 1, 1, 1]$ which

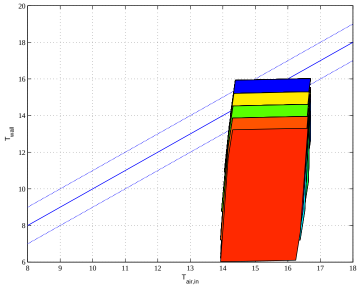

Figure 6.5: Reach set of the system at $k = 4$, $T_{wall} = T_{air,in} + 1$, $T_{wall} = T_{air,in} - 1$ and $T_{wall} = T_{air,in}$

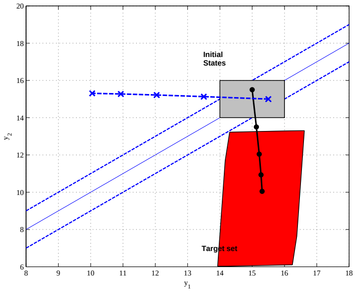

Figure 6.6: Initial states, the target polytope, the observed output (dashed) and the predicted output (solid).

means that both valves should be opened, that is we should cool down the system as soon as possible. It is shown in [9] that when there is both continuous and discrete inputs, the main complexity of the algorithm is due to discrete inputs which cause switching and therefore nonconvexity in the reach set. If the computational complexity is too high, it is possible to fix some of discrete inputs and diagnose the system at the cost of losing the optimal input. However, sensor assignment computations can be done offline.

A frequent fault in the refrigeration system is that the value of the pressure sensor is fixed at its value when the fault happens. Fig.6.7 shows a simulation of the refrigeration system controlled by a hysteresis controller for $T_{air,in}, P_{suc}$ where the upper and the lower values for $T_{air,in}$ are $0, 4$ and those of P_{suc} are $1, 1.5$. If this fault happens, for example at $t = 300$, no passive diagnosis method will be able to detect it until $t = 1162$. This is because the normal system and the faulty system in this period exhibit the same behaviour. Using the active diagnosis method helps us to diagnose the fault faster. We have applied the algorithm and the input sequence is to open V_{evap} for 3 sampling times. The reason for this can be easily seen if one looks at the behaviour of the system at $t = 1162$ when the controller opens V_{evap} and as its result P_{suc} increases.

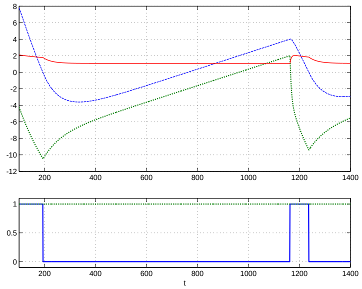

Figure 6.7: Top: $T_{air,in}$(dashed), T_{wall}(dotted), P_{suc}(solid), Bottom: V_{comp}(dotted)V_{evap}(solid).

7 Conclusion

In this paper an approach to the problem of sensor assignment based on active fault diagnosis is proposed and tested on a supermarket refrigeration system. The active diagnosis approach could also be used for sanity check at the commissioning phase or for faster detection of faults during the normal operation of the system.We extended the previous result on active diagnois for sensor assignment such that we do not need reach set computation for every possible assignment, but reach set computation is itself a computationally burdensome task. An algorithm that does not need reach set computation would be desirable. In our future work it will be investigated using a reformulation as an optimization problem.

References

[1] R. Mohammadi, S. Hashtrudi-Zad, and K. Khorasani, "A hybrid architecture for diagnosis in hybrid systems with applications to spacecraft propulsion system," in *IEEE International Conference on Systems, Man and Cybernetics*, 2007, pp. 3184–3190.

[2] S. Narasimhan and G. Biswas, "Model-based diagnosis of hybrid systems," *IEEE transactions on man and cybernetics*, vol. 37, no. 3, pp. 347–361, 2007.

[3] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, and P. Cheung, "Monitoring and fault diagnosis of hybrid systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, pp. 1225–1240, 2005.

[4] S. Campbell and R. Nikoukhah, *Auxiliary Signal Design for Failure Detection*. Princeton University Press, 2004.

[5] I. Andjelkovic, K. Sweetingham, and S. Campbell, "Active fault detection in nonlinear systems using auxiliary signals," in *American Control Conference*, Seattle, WA, 2008, pp. 2142–2147.

[6] H. H. Niemann, "A setup for active fault diagnosis," *IEEE Transactions on Automatic Control*, vol. 51, no. 9, pp. 1572–1578, 2006.

[7] F. Lin, "Diagnosability of discrete events systems and its application," *Discrete event systems*, vol. 4, pp. 197–212, 1994.

[8] M. Sampath, S. Lafortune, and D. Teneketzis, "Active diagnosis of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 908–929, 1998.

[9] S. Tabatabaeipour, A. P. Ravn, R. Izadi-Zamanabadi, and T. Bak, "Active fault diagnosis of linear hybrid systems," in *Safeprocess09*, 2009, pp. 211–216.

[10] A. Balluchi, L. Benvenuti, M. D. D. Benedetto, and A. Sangiovanni-Vincentelli, "Design of observers for hybrid systems," in *5th International Workshop on Hybrid Systems: Computation and Control*. London, UK: Springer-Verlag, 2002, pp. 76–89.

[11] A. Girard, C. L. Guernic, and O. Maler, "Efficient computation of reachable sets of linear time-invariant systems with inputs," in *Hybrid systems: Computation and Control*, ser. Lecture Notes in Computer Science, vol. 3927. Springer, 2006, pp. 257–271.

[12] L. Larsen, R. Izadi-Zamanabadi, R. Wisniewski, and C. Sonntag, "Supermarket refrigeration systems-a benchmark for the optimal control of hybrid systems," Technical report for the HYCON NoE, 2007. http://www. bci. tudortmund. de/ast/hycon4b/index. php, Tech. Rep., 2001.

Paper C

Active Diagnosis of MLD Systems using Distinguishable Steady Outputs

Seyedmojtaba Tabatabaeipour, Anders P. Ravn, Roozbeh Izadi-zamabadi, and Thomas Bak

This paper is accepted in:
IEEE International Symposium on Industrial Electronics, 2010

Abstract

In active diagnosis the system is excited by a signal that aims to uncover latent errors. However, the diagnosis signal may destabilize the system, in particular in an open-loop structure, but also in a closed-loop structure, because the nominal controller is designed to stabilize the nominal system. This paper presents a method for active diagnosis of MLD systems where instability is avoided: The diagnoser looks for steady states of both the normal and faulty system which are reachable by the same input such that the corresponding outputs are distinguishable from each other. The input is applied to the system and the condition of the system is determined based on the output. Thus this excitation preserves stability. The method can be useful in a design phase to find a sensor allocation which guarantees diagnosability. The method is tested on the two tank benchmark example.

1 Introduction

In a complex control system there are many components with strong interaction between them. Hence the overall system performance depends on the individual performance of components. A fault in a single component may, therefore, degrade the overall performance of the system and may even lead to unacceptable loss of system functionality. Thus fault diagnosis is of crucial importance in automatic control of complex systems.

There are two main categories of diagnosis methods : passive and active. In passive diagnosis, the the input and output of the system is observed by the diagnoser. Based on the observation the diagnoser decides whether a fault has occurred or not. The input is generated by an external input or by the controller.

In Active Fault Diagnosis (AFD) the diagnoser generates an input, which excites the system, to decide whether the output represents a normal or a faulty behavior and if possible decide which fault occurred. The generated input must perturb the system from the operation point but at the same time not lead the system to instability or to an unacceptable performance.

The area of active diagnosis has attracted a lot of attentions in recent years. See papers [1], [2], [3], [4], [5], [6], [7], [8], and books [9], [10]. Most of the available methods are in open-loop configuration and for linear systems. In [11] a method for active diagnosis of hybrid system based on reachabilty anaysis is proposed and extended for automatic sensor assignment in [12]. [13] proposes a model predictive method for active diagnosis of hybrid system using Mixed Logical Dynamical (MLD) framework. A qualitative event-based approach for active diagnosis of hybrid systems is presented in [14] where diagnosis is improved by executing or blocking controlable events. [6] and [7] present a method for active diagnosis of parametric faults in closed loop systems based on YJKB parameterization.

Stability is an important issue in the fault tolerant control systems. When a fault occurs, it takes time for the fault detection module to detect the fault and even when it is detected it needs some time to isolate and identify the fault. During this period the system is working in a faulty condition. For a closed-loop system, because the controller is designed for the nominal system the performance of the closed loop system in this period is dependent on the severity of the fault and the robustness of the nominal controller. The controlled system may become unstable in this period [15]. The faulty system may not be stabilizable with the nominal controller and the time window for an unstable system , *e.g.* double inverted pendulum, may be too small to detect and isolate the fault and then reconfigure the loop. [16].

For active diagnosis the stability issue is more critical because we are exciting the system with the aim of detecting the fault. When the AFD starts the diagnosis it is not known whether the system is in the normal or the faulty condition. A stability guaranteeing method for diagnosis of additive, parametric and multiplicative faults for linear systems based on observer parameterization is proposed in [17].

In [13] a model predictive method is proposed for active diagnosis of MLD system. The problem is reformulated as a mixed integer programming problem. The objective function of the optimization problem is to make an observable difference between predicted outputs of the normal system and the faulty systems fulfilling constraints imposed by required performance during fault detection. While the computed input sequence diagnoses the fault, it may destabilize the system.

In this work a different approach is used. The system is moved from its current states to other steady states. These steady states belong to either the normal system or a faulty system which are reachable by the same input and the corresponding steady outputs are distinguishable. The fault is diagnosed based on the output measurement. Because the system is moving to steady state, regardless of its condition, injected input does not destabilize the system. When it is not possible to find a diagnosis signal that separates the the output of the normal system from that of a faulty system, the diagnosis using separating output may not be possible. In this case this approach could be used as a pre-analysis for deciding which outputs must be measured to have the capability of diagnosis using steady outputs.

The structure of the paper is as follows. In Section 2 preliminaries and problem formulation are explained. Section 3 explains the proposed algorithm. In section 4, the method is tested on the two tank example. And finally conclusions and future investigation are discussed in Section 5.

2 Preliminaries and Problem formulation

In this section we first introduce the MLD framework and then the active diagnosis problem is formulated.

2.1 Mixed Logical Dynamical Systems

For modeling of hybrid systems, the mixed logical dynamical (MLD) framework proposed in [18] is used. The equations describing an MLD system are as follows:

$$x(t+1) = Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \tag{7.1}$$

$$y(t) = Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) \tag{7.2}$$

$$E_2\delta(t) + E_3z(t) \leq E_1u(t) + E_4z(t) + E_5 \tag{7.3}$$

where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_l}$ are states, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}$ are the inputs, $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_l}$ are the outputs. $\delta \in \{0,1\}^{r_l}$ and $z \in \mathbb{R}^{r_c}$ are auxiliary binary and continuous variables.

The MLD framework has the capability of modeling various classes of hybrid systems such as PieceWise Affine (PWA) systems, linear systems with piecewise linear output functions, linear systems with discrete inputs or with qualitative outputs, bilinear systems, and finite state machines in which a linear time invariant system generates the events [18].

Equivalence of MLD systems with other classes of hybrid systems such as PWA systems, linear complementary (LC) systems, extended linear complementary (ELC) systems, and max-min-plus-scaling (MMPS) systems under some assumptions is shown in [19].

Using the MLD framework different problems such as optimal control, state estimation, etc. can be reformulated as a mixed-integer programming problem and then can be solved using mixed integer programming techniques.

2.2 Problem Formulation

In model-based passive diagnosis, the diagnoser receives a sequence of input/output measurements. A model of the normal system $\mathcal{B}_0$ and different models of the system subject to different faults, namely $\mathcal{B}_1, \ldots, \mathcal{B}_n$. are given. Then, the diagnoser checks the consistency of the measured I/O sequence with given model. As explained in [20], the output of the diagnoser is a fault candidate index $f \in 1, \ldots, n$ such that the observed I/O sequence is consistent with the corresponding behavior $\mathcal{B}_f$[20]. In this case the input is given by an external system.

The structure of an active diagnoser in depicted in Fig. 7.1. It consists of a generator and a diagnoser. The generator generates an input sequence $U = \langle u(0), \ldots, u(m) \rangle$ which is applied to the system and then index f is determined by the diagnoser through the observation of the applied input sequence and the output sequence $Y = \langle y(0), \ldots, y(m) \rangle$.

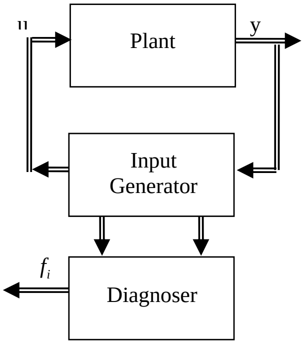

Figure 7.1: Structure of an Active fault diagnoser system

The active diagnosis problem can be stated as follows:

Problem 7.1 (Active diagnosis problem)**.** *Given the set* $\mathcal{B} = \{\mathcal{B}_0, \ldots, \mathcal{B}_n\}$ *describing behaviors of the system with no fault and subject to faults* $\{f_1, \ldots, f_n\}$*, find a sequence of inputs* U *and* $i \in \{0, \ldots, n\}$ *such that* (U, Y) *belongs only to* $\mathcal{B}_i$*.*

If the input sequence exists, *i.e.* if the system is diagnosable then we can look for the optimal solution, where optimality can be interpreted in different senses.

The main advantage of active diagnosis is when different behaviors of the system overlap, see Fig. 7.2. The faultless behavior and the behavior of the system subject to the fault f_1 are in the sets $\mathcal{B}_0$ and $\mathcal{B}_1$ repectively. As long as the observed I/O pair uniquely belongs to the set $\mathcal{B}_0$ or $\mathcal{B}_1$, such as point A or B, it can be decided whether the system is faulty or not. But if the observed pair belongs to the intersection of $\mathcal{B}_0$ and $\mathcal{B}_1$, like C, it is impossible to diagnose the fault. The main idea of the proposed algorithm is to generate an input signal to move the system from C to an area which belongs uniquely either to the set $\mathcal{B}_0$ or $\mathcal{B}_1$.

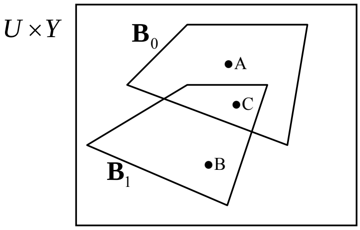

Figure 7.2: Input-Output Space

3 The Proposed Algorithm

It is assumed that the initial states of the system are in the area in which the faulty behavior and the normal behavior overlap. If this is not the case the fault could be diagnosed by means of a passive diagnoser. It is assumed that the model of the faulty system and the normal system is given in MLD form as in (7.1)-(7.3) with subscript 0 indicating the normal system and i indicating the system equation for the system subject to fault f_i.

The diagnosis aims at finding a sequence of inputs such that the outputs based on the different dynamics becomes distinguishable from each other. In another words:

$$Y_i \neq Y_j, \quad \forall\, i, j \in \{0, \ldots, n\}, i \neq j \tag{7.4}$$

This difference between $y_i(k)$ and $y_j(k)$ should be observable which means:

$$|y_i(T) - y_j(T)| \geq d \texttt{ for all } i, j \in \{0, \ldots, n\}, i \neq j \tag{7.5}$$

or if a relative separation is used: $|y_i(T) - y_j(T)| \geq d \cdot |y_i(T)|$, where T is the length of the sequence and d is a separation distance that is dependent on the level of noise.

Satisfaction of the above constraints, (7.4) and (7.5), is actually isolation for every single fault. Isolation for every single fault is very demanding and may not be necessary. One can consider the following scenarios which are less demanding:

- **Fault detection**: In this case, the aim is to find if the system is working normally or it is faulty. We are not interested to detect which fault has occurred. Therefore (7.4) can be relaxed as:

$$|y_0(T) - y_i(T)| \geq d, \quad \forall\, i \in \{1, \ldots, n\}, \tag{7.6}$$

- **Fault isolation for a set of faults**: It is possible that a set of faults have the same impact on the functionality of the system and also require the same fault accommodation or control reconfiguration actions. Therefore it is not required to isolate these faults. Moreover it could be the case that these faults cannot be isolated easily and therefore we just aim at isolation of the set. It is assumed that indices for these faults is given by the set $\mathcal{F}$, then (7.4) becomes:

$$|y_i(T) - y_j(T)| \geq d \;\; \forall\, i \in \mathcal{F}, j \notin \mathcal{F}, \tag{7.7}$$

Note that a practical approach is to first detect the fault. Then isolate a set and then isolate a fault in this set.

Due to rich behavior of a MLD system it may have different steady states. We use this property. In this work, we are looking for steady states from systems $i, j \in \{0, \cdots, n\}$ namely, x_{s_i}, such that the corresponding output are distinguishable i.e. :

$$\left|y_{s_i} - y_{s_j}\right| \geq d, \;\; \forall\, i, j \in \{0, \ldots, n\}, i \neq j \tag{7.8}$$

If these steady outputs exist then the fault is diagnosable.

A steady state value for an MLD system can be obtained by solving a mixed integer problem of the following form:

$$\begin{aligned} \min_{x_s, u_s, \delta_s, z_s} \;\; & \|Q_1(y_s - y_r)\|_p + \|Q_2(x_s - x_r)\|_p + \\ & \|Q_3(u_s - u_r)\|_p + \|Q_4(\delta_s - \delta_r)\|_p + \\ & \|Q_1(z_s - z_r)\|_p \end{aligned} \tag{7.9}$$

$$s.t. \begin{cases} x_s = Ax_s + B_1 u_s + B_2 \delta_s + B_3 z_s \\ y_s = Cx_s + D_1 u_s + D_2 \delta_s + D_3 z_s \\ E_2 \delta_s + E_3 z_s \leq E_1 u_s + E_4 x_s + E_5 \end{cases} \tag{7.10}$$

,where $\|.\|_p$ is p norm, Q_i are positive definite weighting matrices. y_r, x_r, u_r, d_r, z_r are given offset vectors.

It is possible that the resulting steady state $(x_s, u_s, \delta_s, z_s)$ is not reachable. It is also possible that a steady state does not exist but cycling-steady states exist [21]. Here we assume that the steady state is reachable and that we do not have cycling-steady state behavior.

Distinguishable steady outputs, if they exist, can be found by solving the the following problem:

$$\begin{aligned} \min_{x_{s_i}, u_s, \delta_{s_i}, z_{s_i}} \;\; & \Sigma_{i=0}^{n} \|Q_{1_i}(y_{s_i} - y_r)\|_p + \|Q_{2_i}(x_{s_i} - x_r)\|_p \\ & \|Q_{3_i}(u_s - u_r)\|_p + \|Q_{4_i}(\delta_{s_i} - \delta_r)\|_p + \\ & \|Q_{5_i}(z_{s_i} - z_r)\|_p \end{aligned} \tag{7.11}$$

$$s.t. \begin{cases} x_{s_i} = Ax_{s_i} + B_{1_i} u_s + B_{2_i} \delta_{s_i} + B_{3_i} z_{s_i} \\ y_{s_i} = C_i x_{s_i} + D_{1_i} u_s + D_{2_i} \delta_{s_i} + D_{3_i} z_{s_i} \\ E_{2_i} \delta_{s_i} + E_{3_i} z_{s_i} \leq E_{1_i} u_{s_i} + E_{4_i} x_{s_i} + E_{5_i} \\ \left|y_{s_i} - y_{s_j}\right| \geq d \texttt{ for all } i, j \in \{0, \ldots, n\}, i \neq j \end{cases} \tag{7.12}$$

, where $y_r, x_r, u_r, \delta_r, z_r$, is a reference vector. Selection of this reference vector is based on the phase in which we are doing the diagnosis. If we are in the operating phase, then they are chosen equal to the current operating values. In other words, we want to find those steady states which are the closest to the current operating point and at the same time are distinguishable. If we are in the commissioning phase, they are equal to the reference signals. In other words, we are looking for those steady states which are closest to the reference signals and are distinguishable. Note that additional constraint on states and outputs could be easily handled in this formulation by adding them to the optimization constraints.

In (7.11), the distinguishability constraint $\left|y_{s_i} - y_{s_j}\right| \geq d$ should be written in the appropriate form. To do that, the following auxiliary binary variables are introduced:

$$\begin{aligned} [s_{ij1} = 1] &\leftrightarrow [y_{s_i} - y_{s_j} \leq d] \\ [s_{ij2} = 1] &\leftrightarrow [y_{s_j} - y_{s_i} \leq d] \\ s_{ij} = s_{ij1} \wedge s_{ij2}, i, j &\in \{0, \dots, n\}, i \neq j \\ S &= \vee_{i=0}^{n} s_{ij} \end{aligned} \tag{7.13}$$

The constraints $\left|y_{s_i} - y_{s_j}\right| \geq d \texttt{ for all } i, j \in \{0, \dots, n\}, i \neq j$ can be transformed into the equality constraint $S = 0$ and a set of mixed integer linear inequalities obtained from transforming logical propositions in (7.13) to equivalent mixed integer inequalities using the technique introduced in [18].

The auxiliary binary variable S as it is formulated in (7.13) aims at isolation of every single fault. For other scenarios S is constructed as follows:

- **Fault detection**:

$$S = \vee_{i=1}^{n} s_{0i} \tag{7.14}$$

- **Fault isolation for a set of faults**:

$$S = \vee s_{ij}, \;\; \forall\, i \in \mathcal{F}, j \notin \mathcal{F} \tag{7.15}$$

Using the introduced auxiliary variable, the problem can be rewritten as:

$$\begin{aligned} \min_{x_{s_i}, u_s, \delta_{s_i}, z_{s_i}} \;\; & \Sigma_{i=0}^{n} \|Q_{1_i}(y_{s_i} - y_r)\|_p + \|Q_{2_i}(x_{s_i} - x_r)\|_p \\ & \|Q_{3_i}(u_s - u_r)\|_p + \|Q_{4_i}(\delta_{s_i} - \delta_r)\|_p + \\ & \|Q_{5_i}(z_{s_i} - z_r)\|_p \end{aligned} \tag{7.16}$$

$$s.t. \begin{cases} x_{s_i} = Ax_{s_i} + B_{1_i}u_s + B_{2_i}\delta_{s_i} + B_{3_i}z_{s_i} \\ y_{s_i} = C_i x_{s_i} + D_{1_i}u_s + D_{2_i}\delta_{s_i} + D_{3_i}z_{s_i} \\ E_{2_i}\delta_{s_i} + E_{3_i}z_{s_i} \leq E_{1_i}u_s + E_{4_i}x_{s_i} + E_{5_i} \\ S = 0 \end{cases} \tag{7.17}$$

If the above optimization problem is feasible then there are x_{s_i} for $i = 0, \cdots, n$ such that the corresponding outputs y_{s_i} are distinguishable. Otherwise if the optimization problem in (7.16), (7.17) is infeasible, then the system is not diagnosable by this method. Having the steady state values, we can apply the steady inputs u_s to the system and based on the steady outputs decide about its condition. Assume that the actual output of the system at steady state is y_s. Then the fault candidate is f_c such that:

$$c = \underset{i\in\{0,\ldots,n\}}{\operatorname{argmin}} |y_s - y_{s_i}| \tag{7.18}$$

Note that in this method there is no need to estimate the states of the system and diagnosis can be done just by measuring outputs. It is possible to maximize the difference d which is used for distinguishability by adding the term $-\alpha \cdot d$ to the cost function in (7.16):

$$\begin{aligned} \min_{x_{s_i},u_s,\delta_{s_i},z_{s_i},d} \quad & \Sigma_{i=0}^{n} \|Q_{1_i}(y_{s_i} - y_r)\|_p + \|Q_{2_i}(x_{s_i} - x_r)\|_p \\ & \|Q_{3_i}(u_s - u_r)\|_p + \|Q_{4_i}(\delta_{s_i} - \delta_r)\|_p + \\ & \|Q_{5_i}(z_{s_i} - z_r)\|_p - \alpha \cdot d \end{aligned} \tag{7.19}$$

, where α is a weighting parameter.

The proposed method could be used in the design phase to decide about sensor locations to guarantee diagnosability in the steady states. Different output candidates can be considered. Then the optimization problem is solved. Feasibility of the optimization problem with the output candidate means diagnosability of the system with this method.

4 Example

In this section, the proposed method is tested on the two tank system. The two tank system is shown in Fig. 7.3. The system consists of two cylindrical tanks with cross sectional area A which are connected by two pipes at the bottom and at level h_v. The flows through the pipes, denoted by $Q_{12}V_{12}$ and $Q_{12}V_1$, are controlled using two on/off valves V_{12} and V_1. There is a flow Q_1 through a pump to tank 1 which is a continuous input.

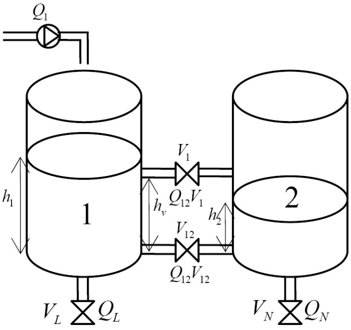

Figure 7.3: Two-tank system

Dynamical equations of the system are as follows.

$$\dot{h}_1 = \frac{1}{A}(Q_1 - Q_{12}V_{12} - Q_{12}V_1 - Q_L), \tag{7.20}$$

$$\dot{h}_2 = \frac{1}{A}(Q_{12}V_{12} + Q_{12}V_1 - Q_N), \tag{7.21}$$

where h_1 and h_2 denote the levels of tanks 1 and 2 respectively. The flow $Q_{12}V_{12}$ is described by:

$$Q_{12}V_{12} = V_{12}k_{12}sign(h_1 - h_2)\sqrt{2g\,|h_1 - h_2|}, \tag{7.22}$$

where g is the gravity constant and k_{12} is a valve specific constant. Similarly $Q_L = V_L k_L \sqrt{2gh_1}$ and $Q_N = V_N k_N \sqrt{2gh_2}$. The flow through valve V_1 is given by:

$$\begin{aligned} Q_{12}V_1 &= V_1 k_1 sign(max\{h_v, h_1\} - max\{h_v, h_2\}) \\ &\quad \sqrt{|2g(max\{h_v, h_1\} - max\{h_v, h_2\})|} \end{aligned} \tag{7.23}$$

The MLD model of the system is derived as follows (For details see [21].). The nonlinear relation $\sqrt{x}$ is approximated by a straight line x, thus (7.22) becomes:

$$Q_{12}V_{12} = V_{12}k_{12}(h_1 - h_2) \tag{7.24}$$

The auxiliary continuous variable $z_{12} = V_{12}(h_1 - h_2)$ is introduced to transform the above nonlinear equation to the linear equation $Q_{12}V_{12} = k_{12}z_{12}$ with a set of mixed integer linear inequalities. For Q_N and Q_L, using the same method, we will have $Q_N = k_N z_N$ and $Q_L = k_N z_L$ where $z_N = V_N h_2$ and $z_L = V_L h_2$.

In order to transform (7.23) to a linear equation in the MLD framework, first we introduce the following binary variables indicating whether the level in each tank has reached h_v:

$$[\delta_{01}(t) = 1] \leftrightarrow [h_1(t) \geq h_v] \tag{7.25}$$

$$[\delta_{02}(t) = 1] \leftrightarrow [h_2(t) \geq h_v] \tag{7.26}$$

and then the term $max\{h_v, h_1\} - max\{h_v, h_2\}$ is transformed into a linear equation as $Q_{12}V_1 = k_1 z_1$, where

$$z_1 = V_1(z_{01} - z_{02}) \tag{7.27}$$

$$z_{01} = \delta_{01}(h_1 - h_v) \tag{7.28}$$

$$z_{02} = \delta_{02}(h_2 - h_v) \tag{7.29}$$

are introduced auxiliary continuous variables.

Finally, differential equations (7.20), (7.21) are discretized in time by Euler approximation $\dot{h}_i(t) \approx \frac{h_i(t+1) - h_i(t)}{Ts}$, where T_s is the sample time. The final MLD model of the system consists of two continuous states: h_1, h_2, 2 binary inputs: V_1, V_{12}, 1 continuous input: Q_1 and two continuous outputs: h_1, h_2, 2 auxialiary binary variables: δ_0, δ_1 and 5 auxiliary continuous variables: $z_{01}, z_{02}, z_1, z_N, z_L$.

5 Simulation Results

The proposed active diagnosis method is used for sanity check of the upper valve V_1. It is assumed that the valve is stuck in the ON position. It is also assumed that at the beginning both tanks are empty *i.e.* $h_1 = h_2 = 0$. The proposed predictive method is applied to check whether the valve is faulty or normal. The variable d is assumed as 0.02 and the sample time is 10 seconds. It is assumed that the valve V_L is always closed and V_N is always open.

To obtain an MLD model of the two tanks system we use HYSDEL (hybrid system description language)[22], which is a modeling language for Discrete Hybrid Automata (DHA). Given a description of the system, HYSDEL translates it into different computational models like MLD or PWA.

We also assume that at the commissioning phase we want to fill the tanks to $y_r = [0.3\ 0.2]'$ and also we want to do the sanity check for V_1. Therefore we look for the closest steady states to y_r such that the outputs are distinguishable. The results of the optimization problem (7.16) are:

$$\begin{aligned} & y_{s_0} = [0.3\ 0.235]', y_{s_1} = [0.2668\ 0.235]' \\ & Q_s = 0.1196, V_1 = 0, V_2 = 1 \end{aligned} \tag{7.30}$$

After finding the steady values a model predictive control is designed such that the normal system output tracks y_{s_0} with the initial states $[0\ 0]$. Figure 7.4 shows the result. As it can be seen by comparing the actual steady outputs with the expected values of the normal system it can be determined that the system is faulty.

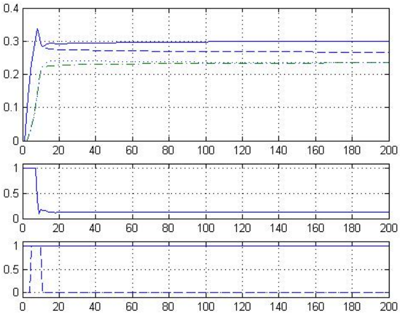

Figure 7.4: Top:Actual versus expected output of the system:h_1'-', h_2'.', $\hat{h}_1$'- -', $\hat{h}_2$'.-', Middle: Continuous input Q_1, Bottom:Binary inputs: V_1 dashes and V_{12} solid

As we said in the introduction, another application of the method is when the faulty system and the normal system have the same behaviors. This situation for the two tank example is demonstrated in Fig. 7.5. In this example a model predictive controller is designed for the two tank system to drive the system from $[0\ 0]$ to the equilibrium point [0.2664 0.2349]. This is a steady state for both the normal system and the faulty system.

Fig. 7.5 shows the simulation of the closed loop system. As one can see, the control variable V_1 is manipulated such that the output of the system in the normal condition and in the faulty one is exactly the same. In this situation if a stuck ON fault happens, no passive diagnoser would be able to diagnose it. In order to detect the fault by our method we look for separating steady state points close to the current steady state. The resulting steady values are:

$$
\begin{aligned}
& y_{s_0} = [0.2667\ 0.2089]', y_{s_1} = [0.2372\ 0.2089]' \\
& Q_s = 0.1063, V_1 = 0, V_2 = 1
\end{aligned}
\tag{7.31}
$$

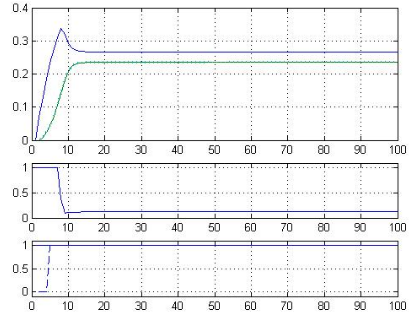

Figure 7.5: Top:Actual versus expected output of the system, Middle:continuous input Q_1, Bottom:discrete inputs: V_1(dashed line), V_{12}(solid line)

The steady inputs are applied to the system and the result is shown in figure 7.6. As it can be seen the condition of the system is detectable by steady values of the output.

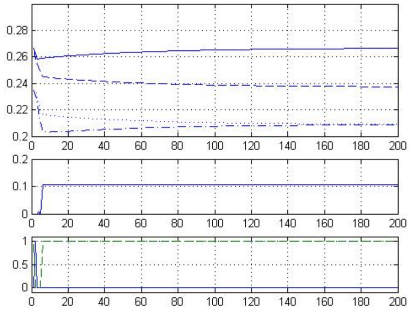

Figure 7.6: Top:Actual versus expected output of the system:h_1'-', h_2'.', $\hat{h}_1$'- -', $\hat{h}_2$'.-', Middle:continuous input Q_1, Bottom:discrete inputs: V_1(dashed line), V_{12}(solid line)

As one can see the steady value for h_2 are always the same for the faulty system and the normal system. Because the system is in the steady state, the input flow is equal to the output flow: $Q_1 = Q_N$. Q_1 is the same in both conditions. Since $Q_N = k_n\sqrt{2gh_2}$, it is obvious that in the steady state $h_{2_0} = h_{2_1}$. Therefore if we do not have measurements from h_1, it is not possible to diagnose the fault by using steady values. This analysis can be used in the design phase of the system to decide where we should put sensors to be able to diagnose the fault using steady values. As it is shown in [13], it is possible to diagnose the fault while it is being perturbed from the steady values, but the problem of that method was that it may lead to instability. This method excludes the possibility of diagnosis using transient but preserves stability. A drawback of the method is that it takes a long time to reach the steady state values and therefore while it does not destabilize the system it needs a long time for diagnosis.

6 Conclusion

In this paper a method for active diagnosis of MLD system based on analysis of steady state values of the system in normal and faulty modes is presented. The excitation obtained by this method does not destabilize the system because it moves the system to a steady state, but it is possible that there are not enough distinguishable steady output values and therefore the fault is not diagnosable using steady state values. However, this analysis method can be used in a design phase to decide about the location of sensors to guarantee diagnosability. While the method guarantee the stability during diagnosis because we should wait till the system reaches steady values the approach need a long period for diagnosis.

References

[1] R. Nikoukhah, S. L. Campbell, K. G. Horton, and F. Delebecque, “Auxiliary signal design for robust multimodel identification,” *IEEE Transactions on Automatic Control*, vol. 47, no. 1, pp. 158–164, 2002.

[2] S. L. Campbell, K. G. Horton, and R. Nikoukhah, “Auxiliary signal design for rapid multi-model identification using optimization,” *Automatica*, vol. 38, no. 8, pp. 1313–1325, 2002.

[3] R. Nikoukhah, S. L. Campbell, A. Savkin, and R. Selmic, “A multi-model approach to failure detection in uncertain sampled-data systems,” *European journal of control*, vol. 11, no. 3, pp. 255–268, 2005.

[4] R. Nikoukhah and S. L. Campbell, “Auxiliary signal design for active failure detection in uncertain linear systems with a priori information,” *Automatica*, vol. 42, no. 2, pp. 219–228, 2006.

[5] D. Choe, S. L. Campbell, and R. Nikoukhah, “Optimal piecewise-constant signal design for active fault detection,” *International Journal of Control*, vol. 82, no. 1, pp. 130–146, 2009.

[6] H. H. Niemann and N. K. Poulsen, “Active fault diagnosis in closed-loop systems,” in *Proceedings of the 16th IFAC World Congress*, 2005.

[7] H. H. Niemann, “A setup for active fault diagnosis,” *IEEE Transactions on Automatic Control*, vol. 51, no. 9, pp. 1572–1578, 2006.

[8] N. K. Poulsen and H. H. Niemann, “Active fault diagnosis based on stochastic tests,” *International Journal of Applied Mathematics and Computer Science*, vol. 18, no. 4, pp. 487–496, 2008.

[9] X. Zhang, *Auxiliary signal design in fault detection and diagnosis*. Springer Verlag, 1989.

[10] S. Campbell and R. Nikoukhah, *Auxiliary Signal Design for Failure Detection*. Princeton University Press, 2004.

[11] S. Tabatabaeipour, A. P. Ravn, R. Izadi-Zamanabadi, and T. Bak, "Active fault diagnosis of linear hybrid systems," in *Safeprocess09*, 2009, pp. 211–216.

[12] S. Tabatabaeipour, R. Izadi-Zamanabadi, T. Bak, and A. P. Ravn, "Automatic sensor assignment of a supermarket refrigeration system," in *IEEE Multi-Conference on Control Applications, (CCA) & Intelligent Control, (ISIC),*, July 2009, pp. 1319–1324.

[13] S. Tabatabaeipour, A. P. Ravn, R. Izadi-Zamanabadi, and T. Bak, "Active fault diagnosis-a model predictive approach," in *IEEE ICCA'09*, 2009.

[14] M. J. Daigle and G. Biswas, "Improving diagnosability of hybrid systems through active diagnosis," in *Safeprocess09*, 2009, pp. 217–222.

[15] Y. M. Zhang and J. Jiang, "Issues on integration of fault diagnosis and reconfigurable control in active fault-tolerant control systems," in *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processed*, 2006, pp. 1513–1524.

[16] H. H. Niemann and J. Stoustrup, "Passive fault tolerant control of a double inverted penduluma case study," *Control engineering practice*, vol. 13, no. 8, pp. 1047–1059, 2005.

[17] J. Stoustrup, "An observer parameterization approach to active fault diagnosis with applications to a drag racing vehicle," in *Safeprocess09*, 2009, pp. 591–596.

[18] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, pp. 407–428, 1999.

[19] W. Heemels, B. D. Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," *Automatica*, vol. 37, no. 7, pp. 1085–1091, 2001.

[20] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*. Springer-Verlag, 2006.

[21] D. Mignone, "Control and estimation of hybrid systems with mathematical optimization," Ph.D. dissertation, Swiss Federal Institute of Technology, 2002.

[22] F. D. Torrisi and A. Bemporad, "HYSDEL-a tool for generating computational hybrid models for analysis and synthesis problems," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 2, pp. 235–249, 2004.

Paper D

Stabilizable Active Diagnosis of Hybrid Systems

Seyedmojtaba Tabatabaeipour, Roozbeh Izadi-Zamanabadi, Anders. P. Ravn,
and Thomas Bak

This paper is submitted to:
International Journal of Control

Abstract

A method for active diagnosis of hybrid systems is proposed. The diagnosis is done by predicting the future output of both normal and fault affected models of the system. Then an optimization problem is solved with the objective of making an observable difference between the predicted normal and faulty outputs. To ensure that the system remains stable with this excitation, a model predictive controller is superimposed on the active diagnoser. Stabilizability of the system to an equilibrium state is guaranteed by imposing constraints on the diagnosis optimization which requires the final state of the system to be in the feasible set of a model predictive controller. If the optimization is feasible, the fault is diagnosable and reconfigurable. Once the fault is isolated, the MPC constraints are updated and the system is reconfigured.

It is demonstrated how the excitation signal generated by the active diagnoser can be used as a test signal in a sanity check in the commissioning of a system for detection of faults hidden by regulatory actions of the controller. The method is demonstrated on the two tank benchmark example.

1 Introduction

In a complex control system there are many components with strong interaction between them. Hence the overall system performance depends on the individual performance of components. A fault in a single component may, therefore, degrade the overall performance of the system and may even lead to unacceptable loss of system functionality. Thus fault diagnosis is of crucial importance in automatic control of complex systems.

Diagnosis methods can be divided into two main categories: active and passive. In passive diagnosis, the diagnoser observes the input and output of the system and based on the observation decides whether a fault has occurred or not. The input is generated by an external input or by the controller. In active fault diagnosis the diagnoser generates an input, which excites the system, to decide whether the output represents a normal or a faulty behaviour and if possible decide which faulty behaviour occurred. The generated input moves the system from the operation point, but at the same time it should not lead the system to instability or to an unacceptable performance area.

Active diagnosis (AD)is useful in the following circumstances: (i) for generation of the test signal in the commissioning phase for sanity check of the system, (ii) for faster detection of faults during normal operation, and (iii) for detection of hidden faults where, because of regulatory actions of the controller, the normal and the faulty system exhibit the same behaviour.

Typical industrial systems include both continuous and discrete components. For a precise modeling of them a hybrid system formulation is useful. Generally speaking, a hybrid system is a dynamical system with both continuous and discrete behaviours and non-trivial interaction between continuous evolutions and discrete transitions. Hybrid systems have been subject of intensive research in recent years, for an overview see [1]. Fault diagnosis of hybrid systems has been investigated recently, for a survey see [2], [3] and [4].

Most of results are in the area of passive diagnosis.

The area of active diagnosis has attracted a considerable attentions in recent years, see papers [5], [6], [7], [8], [9], [10], [11], [12], and books [13], [14]. Most of the available methods are in open-loop configuration and for linear systems. A qualitative event-based approach for active diagnosis of hybrid systems is presented in [15], where diagnosis is improved by executing or blocking controllable events. [10] and [11] present a method for active diagnosis of parametric faults in closed loop systems based on YJKB parameterization.

In previous work [16], we proposed an active fault diagnosis method for linear hybrid systems in discrete time based on reach set computation for faulty and normal systems. The results are extended to automatic sensor assignment in [17]. Because of computational complexity of reach set computation, the problem is reformulated in [18] as a mixed integer optimization problem for active diagnosis of hybrid system using the Mixed Logical Dynamical framework.

Stability is an important issue in fault tolerant control systems. When a fault occurs, it takes time for the fault detection module to detect the fault and even when it is detected it needs some time to isolate and identify the fault. During this period the system is working in a faulty condition. For a closed-loop system, because the controller is designed for the nominal system the performance of the system in this period is mainly dependent on the severity of the fault and the robustness of the nominal controller. It is clear that the controlled system may become unstable in this period, see [19].

For active diagnosis the stability issue is more critical, because we are exciting the system with the aim of detecting the fault. When the AD starts the diagnosis, it is not known whether the system is in the normal or the faulty condition. A stability preserving method for diagnosis of additive, parametric and multiplicative faults for linear systems based on observer parameterization is proposed in [20]. In [21], a method using distinguishable steady states for diagnosis of MLD systems is presented which preserves the stability. But if there are no distinguishable steady states, the method cannot use the information during the transient. In this work the diagnosis is done by perturbing the system from the operating point. It is guaranteed that this perturbation does not destabilize the system. It is guaranteed that the diagnosed system can be stabilized by a model predictive control with constraints updated based on the information provided by diagnosis. This is done by imposing some constraints on the diagnosis optimization problem which requires the final state of the system, in whatever condition the system is, to be in the feasible set of a predictive controller which stabilizes the system to an equilibrium point.

For modeling of hybrid system we use the Mixed Logical Dynamical (MLD) framework of [22], [23] which covers important classes of hybrid system. By using the MLD framework, the optimization problem used for fault diagnosis will be transformed to a mixed integer linear or quadratic problem for which there are many efficient solvers. The method is tested on the two-tank benchmark example.

The structure of the paper is as follows: In Section 1 we introduce mixed logical dynamical systems; then model predictive control in introduced briefly and the problem of active diagnosis is formulated. Section 2 gives the proposed active diagnosis algorithm. Section 4 describes the two tank benchmark example and simulation results are presented in Section 5. The paper concludes in Section 6.

2 Preliminaries and Problem formulation

We first introduce the MLD framework and then the model predictive control of MLD systems is explained and finally the active diagnosis problem is formulated.

2.1 Mixed Logical Dynamical Systems

For modeling of hybrid systems, the mixed logical dynamical (MLD) framework proposed in [22] is used. The equations describing an MLD system are as follows:

$$x(t+1) = Ax(t) + B_1 u(t) + B_2 \delta(t) + B_3 z(t) \tag{8.1}$$

$$y(t) = Cx(t) + D_1 u(t) + D_2 \delta(t) + D_3 z(t) \tag{8.2}$$

$$E_2 \delta(t) + E_3 z(t) \leq E_1 u(t) + E_4 z(t) + E_5 \tag{8.3}$$

where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_l}$ are states, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}$ are the inputs, $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_l}$ are the outputs. $\delta \in \{0,1\}^{r_l}$ and $z \in \mathbb{R}^{r_c}$ are auxiliary binary and continuous variables. A trajectory of MLD system, starting from initial state $x(t_0) = x_0$, when the input sequence $\{u\}_{t_0}^{t-1} = \{u(t_0), u(t_0+1), \cdots, u(t-1))\}$ is applied to the system, is denoted by $x(t, t_0, x_0, \{u\}_{t_0}^{t-1})$.

Definition 8.1 (Equilibrium state:). $x_e \in \mathbb{R}^{n_c} \times \{0,1\}^{n_l}$ is an equilibrium state of the MLD system (8.1)-(8.3) with input $u_e \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}$ if $x(t, t_0, x_e, u_e) = x_e \; \forall t \geq t_0, \forall t_0 \in \mathbb{Z}$. The corresponding output y_e is called the equilibrium output and the pair (x_e, u_e) is called the equilibrium pair.

The MLD framework has the capability of modeling various classes of hybrid systems such as PieceWise Affine (PWA) systems, linear systems with piecewise linear output functions, linear systems with discrete inputs or with qualitative outputs, bilinear systems, and finite state machines in which an LTI system generates the events, see[22].

Equivalence of MLD systems with other classes of hybrid systems such as PWA systems, linear complementary (LC) systems, extended linear complementary (ELC) systems, and max-min-plus-scaling (MMPS) systems under some assumptions is shown in [24].

Using the MLD framework, different problems such as optimal control, state estimation, etc. can be reformulated as mixed-integer programing problems and be solved using mixed integer programming techniques.

2.2 Model Predictive Control

Consider the MLD system (8.1)-(8.3) with constraints on input and states *i.e.* $x(k) \in \mathbb{X} \times \{0,1\}^{n_l}$ and $u(k) \in \mathbb{U} \times \{0,1\}^{m_l}$, where $\mathbb{X} \subseteq \mathbb{R}^{n_c}$ and $\mathbb{U} \subseteq \mathbb{R}^{m_c}$ are compact polyhedral sets that contain the equilibrium pair (x_{c_e}, u_{c_e}) in their interior.

Define $x(k|t) \triangleq x(t+k, t, x(t), \{u\}_t^{k-1})$ and let $\delta(k|t), z(k|t), y(k|t)$ similarly defined . Let $\mathbf{x}_k(x(k), \{u\}_k^{k+T-1}) \triangleq \{x(k+1|k), \cdots, x(k+T|k)\}$ be the sequence generated from the initial state $x(k|k) = x(k)$ by applying the input sequence $\{u\}_k^{k+T-1} \triangleq \{u(k+1|k, \cdots, u(k+T-1|k))\}$. Assuming the equilibrium pair (x_e, u_e) as the desired target point, then

$$\mathcal{U}_T(x(k)) \triangleq$$
$$\{(u) \in \mathbb{U}^T \times \{0,1\}^{Tm_l} | \mathbf{x}_k(x(k), \{u\}_k^{k+T-1}) \in \mathbb{X}^T \times \{0,1\}^{Tn_l}, x(T|k) = x_e\} \tag{8.4}$$

is the class of admissible input sequences with respect to x_e and $x(k)$. The cost function $J(x(k), \mathbf{u}_k)$ is defined as:

$$\begin{aligned} J(x(k), \mathbf{u}_k) \triangleq & \|Q_1(y(k|t) - y_e)\|_p + \|Q_2(x(k|t) - x_e)\|_p + \|Q_3(u(k|t) - u_e)\|_p + \\ & \|Q_4(\delta(k|t) - \delta_e)\|_p + \|Q_5(z(k|t) - z_e)\|_p \end{aligned} \tag{8.5}$$

,where Q_1, Q_2, Q_3, Q_4, Q_5 are symmetric positive definitive matrices.

Given $x(t)$, the optimal MPC minimizes, at each time $t \in \mathbb{Z}$, the objective function J subject to constrainsts:

$$\begin{cases} x(T|t) = x_e \\ x(t|t) = x(t) \\ x(k+1|t) = Ax(k|t) + B_1 u(k) + B_2\delta(k|t) + B_3 z(k|t) \\ y(k|t) = Cx(k|t) + D_1 u(k) + D_2\delta(k|t) + D_3 z(k|t) \\ E_2\delta(k|t) + E_3 z(k|t) \leq E_1 u(t) + E_4 z(k|t) + E_5 \\ x(k) \in \mathbb{X} \times \{0,1\}^{n_l} \\ u(k) \in \mathbb{U} \times \{0,1\}^{m_l} \end{cases} \tag{8.6}$$

We assume that there exists an optimal sequence $\mathbf{u}_k^* \triangleq \{u^*(k|k), \cdots, u^*(k+N-1|k))\}$ for this problem. The MPC control law is defined as the first element of this sequence:

$$u^{MPC} x(k) \triangleq u^*(k|k) \tag{8.7}$$

The input is applied to the system and the whole procedure is repeated at the next time instance.

In (8.6), the constraint $x(T|t) = x_e$ is the stability constraint and $x(k) \in \mathbb{X} \times \{0,1\}^{n_l}, u(k) \in \mathbb{U} \times \{0,1\}^{m_l}$ are state and input constraints.

The set of states for which the constraints (8.6) are feasible is called as feasible set.

Definition 8.2 (Feasible set)**.** The feasible set $\mathbb{X}_F(T)$ is defined as

$$\mathbb{X}_F(T) = \{x \in \mathbb{X} \times \{0,1\}^{n_l} | \mathcal{U}_T(x) \neq \emptyset\} \tag{8.8}$$

The Following theorem shows that in the MPC problem, feasibility is preserved over time and that feasibility implies stability.

Theorem 8.1. *Assume that (x_e, u_e) is an equilibrium pair. Fix $T \in \mathbb{Z}_{\geq 1}$. If the optimization problem (8.5) with constraints (8.6) is feasible for $x(t)$ at time t, then it is feasible at time $t+1$ for state $x(t+1)$ which is evolving based on MLD system equation in (8.1)-(8.3) with input $u^{MPC}x(k)$. Moreover the MPC law (8.7) stabilizes the system.*

Proof. see [22]. □

2.3 Active Diagnosis Problem

A passive model-based diagnoser, as depicted in Fig. 8.1-a, is a system which receives a sequence of input/output measurements and checks the consistency of the measured I/O sequence with a given model of the normal system $\mathcal{B}_0$ and models of the system subject to different faults , namely $\mathcal{B}_1, \ldots, \mathcal{B}_n$. The output of the diagnoser is a candidate index

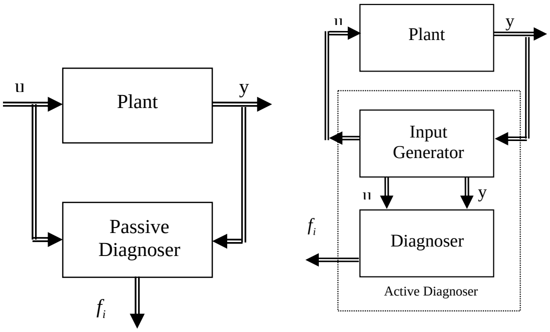

Figure 8.1: Passive versus Active fault diagnoser

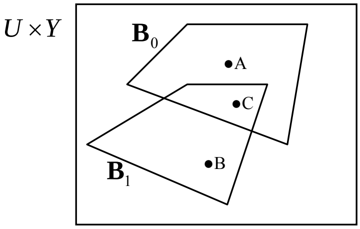

Figure 8.2: System behaviour

$f \in 0, \ldots, n$ such that the observed I/O sequence is consistent with the corresponding behaviour $\mathcal{B}_f$, [25].

The structure of an active diagnoser in depicted in Fig. 8.1-b. It consists of a generator and a diagnoser. The generator generates an input sequence $U = \langle u(0), \ldots, u(m) \rangle$ which is applied to the system and then occurrence of fault f is determined by the diagnoser by observing the applied input sequence and the output sequence $Y = \langle y(0), \ldots, y(m) \rangle$.

The active diagnosis problem can be stated as follows:

Problem 8.1 (Active diagnosis problem:)**.** *Given the set $\mathcal{B} = \{\mathcal{B}_0, \ldots, \mathcal{B}_n\}$ describing behaviors of the system with no fault and subject to faults $\{f_1, \ldots, f_n\}$, find a sequence of inputs U such that (U, Y) belongs only to a unique $\mathcal{B}_i$.*

If such an input sequence exists, *i.e.* if the system is diagnosable then we can look for the optimal solution, where optimality can be interpreted in different senses. In this work we propose an algorithm which looks for a sequence with minimum length.

The main advantage of active diagnosis is when different behaviours of the system overlap, see Fig. 8.2. The faultless behaviour and the behaviour of the system subject to the fault f_1 are displayed by the sets $\mathcal{B}_0$ and $\mathcal{B}_1$ respectively. As long as the observed I/O pair uniquely belongs to the set $\mathcal{B}_0$ or $\mathcal{B}_1$, such as points A or B, it can be decided whether the system is faulty or not. But if the observed pair belongs to the intersection of $\mathcal{B}_0$ and $\mathcal{B}_1$, like C , it is impossible to diagnose the fault. The main idea of the proposed algorithm is to generate an input signal to move the system from C to an area which belongs uniquely either to the set $\mathcal{B}_0$ or $\mathcal{B}_1$.

3 The Proposed Algorithm

It is assumed that the states of the system are available or estimated by means of an observer. The observer could be of the kind proposed in [26] or the MLD estimator proposed in [23]. Using the latter yields a more unified framework. It is supposed that the initial state is in the area where the faulty behaviour and the normal behaviour overlap because otherwise the fault could be diagnosed by means of a passive diagnoser.

We assume that the model of the faulty system and the normal system is given in MLD form as in (8.1)-(8.3) with subscript 0 indicating the normal system and i indicating the system equation for the system subject to fault f_i. The aim of the diagnosis is to find a minimum sequence of inputs such that the outputs based on the different dynamics becomes observably different:

$$|y_i(T_d) - y_j(T_d)| \geq d \quad \forall\, i, j \in \{0, \dots, n\}, i \neq j \tag{8.9}$$

or if a relative separation is used: $|y_i(T) - y_j(T)| \geq d \cdot |y_i(T)|$, where T is the length of the sequence and d is a separation distance that is dependent on the level of noise.

The equation (8.9) aims for achieving isolability for every single fault and are very demanding. Also, one can consider the following scenarios which are less demanding:

- **Fault detection**: In fault detection, we are just interested to detect if the system is working normally or is subject to any fault. In this case (8.9) becomes:

$$|y_0(T_d) - y_i(T_d)| \geq d \;\; \forall\, i \in \{1, \dots, n\}, \tag{8.10}$$

- **Fault isolation for a set of faults**: In such a scenario, we look for a set of faults that have the same impact on the functionality of the system and also require the same fault accommodation or control reconfiguration actions. Therefore we just aim at isolation of the set. Assuming that indices for these faults is given by the set $\mathcal{F}$, then (8.9) becomes:

$$|y_i(T_d) - y_j(T_d)| \geq d \;\; \forall\, i \in \mathcal{F}, j \notin \mathcal{F}, \tag{8.11}$$

In the sequel we address the fault isolation problem. Later it will be explained how it can be reformulated for fault detection or fault isolation for a set of faults.

We are looking for the minimum T such that the condition (8.9) is satisfied. This can be formulated as an optimization problem in the following form:

$$\min_{T, \{u, \delta_i, z_i\}_0^T} \mathbf{1} \tag{8.12}$$

$$s.t. \begin{cases} x_i(t|t) = x_i(t) \\ x_i(k+1|t) = A_i x_i(k|t) + B_{1_i} u(t) + B_{2_i} \delta_{_i}(k|t) + B_{3_i} z_i(k|t) \\ y_i(k|t) = C_i x_i(k|t) + D_{1_i} u(k) + D_{2_i} \delta(k|t) + D_{3_i} z(k|t) \\ E_{2_i} \delta_i(k|t) + E_{3_i} z_i(k|t) \leq E_{1_i} u_i(t) + E_{4_i} z_i(k|t) + E_{5_i} \\ i = 0, \cdots, n \\ |y_i(T_d) - y_j(T_d)| \geq d, \; i, j \in \{0, \dots, n\}, i \neq j \end{cases}$$

Since minimum of a constant is that constant, the above optimization problem is a constraint satisfaction problem. The optimization problem (8.12) can be transformed to a Mixed Integer Linear Programming (MILP) problem by introducing the following auxiliary binary variables.

$$
\begin{gathered}
[s_{ij1}(t) = 1] \leftrightarrow [y_i(t) - y_j(t) \leq d] \\
[s_{ij2}(t) = 1] \leftrightarrow [y_j(t) - y_i(t) \leq d] \\
s_{ij}(t) = s_{ij1}(t) \wedge s_{ij2}(t), i, j \in \{0, \ldots, n\}, i \neq j \\
S(t) = \vee_{i=0}^{n} s_{ij}(t)
\end{gathered}
\tag{8.13}
$$

The introduced variable $S(t)$ is for isolation of every single fault. For other scenarios $S(t)$ should be constructed as follows:

- **Fault detection**:
$$S(t) = \vee_{i=1}^{n} s_{0i}(t) \tag{8.14}$$
- **Fault isolation for a set of faults**:
$$S(t) = \vee s_{ij}(t), \;\; \forall\, i \in \mathcal{F}, j \notin \mathcal{F} \tag{8.15}$$

Using the introduced auxiliary variable, the optimization problem (8.12) can be rewritten as:

$$\min_{T, \{u, \delta_i, z_i\}_0^T} \;\; \mathbf{1} \tag{8.16}$$

$$
s.t. \begin{cases}
x_i(t|t) = x_i(t) \\
x_i(k+1|t) = A_i x_i(k|t) + B_{1_i} u(t) + B_{2_i} \delta_i(k|t) + B_{3_i} z_i(k|t) \\
y_i(k|t) = C_i x_i(k|t) + D_{1_i} u(k) + D_{2_i} \delta(k|t) + D_{3_i} z(k|t) \\
E_{2_i} \delta_i(k|t) + E_{3_i} z_i(k|t) \leq E_{1_i} u_i(t) + E_{4_i} z_i(k|t) + E_{5_i} \\
i = 0, \cdots, n \\
S(T_d) = 1
\end{cases}
$$

where the constraints $|y_i(T_d) - y_j(T_d)| \geq d, \; i, j \in \{0, \ldots, n\}, i \neq j$ are replaced with the corresponding mixed integer linear inequalities obtained from transforming logical propositions in (8.13) to equivalent mixed integer inequalities using the technique introduced in [22].

The optimization problem is similar to a minimum time optimal control problem. Given a normal model and faulty models of the system subject to the faults $\{f_1, \ldots, f_n\}$, an initial state, and a target set, we want to find the minimum T_d and an input sequence $u(t), t = 1, \ldots, T_d$ such that $y(T_d)$ belongs to the target set.

For a fixed T_d, it is actually a Mixed Integer Feasibility Test (MIFT). To solve it, we find a lower bound, T_l, and an upper bound, T_u, for T_d such that it is infeasible for T_l and feasible for T_u. Then we find the minimum feasible T_d namely T_d^* by running a bisection algorithm. When the minimum time for diagnosis is found, the corresponding input sequence is applied to the system. At the end of period, the output $y(T_d)$ is compared with the expected outputs $y_i(T_d)$ and the fault candidate f_c is chosen as the following.

$$f_c = f_i, \; i = \underset{i \in \{0, \ldots, n\}}{\text{argmin}} \, |y(T_d) - y_i(T_d)| \tag{8.17}$$

It is also possible to have a fixed feasible T_d, and then find an input sequence which separates the outputs and tries to keep the distance between them by solving the following optimization problem:

$$\min_{\{u,\delta_i,z_i\}_0^{T_d}} \sum_{k=t}^{t+T_d} S(k) \tag{8.18}$$

$$s.t. \begin{cases} x_i(t|t) = x_i(t) \\ x_i(k+1|t) = A_i x_i(k|t) + B_{1_i} u(t) + B_{2_i} \delta_i(k|t) + B_{3_i} z_i(k|t) \\ y_i(k|t) = C_i x_i(k|t) + D_{1_i} u(k) + D_{2_i} \delta(k|t) + D_{3_i} z(k|t) \\ E_{2_i} \delta_i(k|t) + E_{3_i} z_i(k|t) \leq E_{1_i} u_i(t) + E_{4_i} z_i(k|t) + E_{5_i} \\ S(t) = \vee_{i=1}^{n} s_i(t) i = 0, \cdots, n \end{cases}$$

3.1 Stabilizable Active Diagnosis

The minimum input sequence diagnose the fault; but it may destabilize the system. In order to avoid instability we mount an MPC controller on the top of diagnoser. The aim of the MPC controller is to steer the state of the diagnosed system to an equilibrium point. For the equilibrium point we consider two cases dependent on the aim of diagnosis. First, assume we are in the commissioning phase. We want to steer the system states to an equilibrium state x_r but at the same time we want to perform a sanity check. It might happen that x_r is not an equilibrium state for the system subject to fault f_c. Therefore another equilibrium state close to x_r must be found for the faulty system namely x_{er}. And then the system will be steered to a new steady state x_{er}. A steady state value for an MLD system can be obtained by solving a mixed integer problem of the following form:

$$\begin{aligned} \min_{x_s,u_s,\delta_s,z_s} \; & \|Q_1(y_s - y_r)\|_p + \|Q_2(x_s - x_r)\|_p + \|Q_3(u_s - u_r)\|_p + \\ & \|Q_4(\delta_s - \delta_r)\|_p + \|Q_5(z_s - z_r)\|_p \end{aligned} \tag{8.19}$$

$$s.t. \begin{cases} x_s = A x_s + B_1 u_s + B_2 \delta_s + B_3 z_s \\ y_s = C x_s + D_1 u_s + D_2 \delta_s + D_3 z_s \\ E_2 \delta_s + E_3 z_s \leq E_1 u_s + E_4 x_s + E_5 \end{cases}$$

In the second case, diagnosis is done in the operation phase and the fault is hidden. After revealing the fault, the aim might be fault hiding. In any case we denote the equilibrium goal of the system subject to fault f_i by $x_{i_{er}}$

In order to make sure that the final state of diagnosis $x_i(T_d)$ is steerable to the corresponding equilibrium point, it should be in the feasible set of $\mathbb{X}_F^i(T)$ for the system subject to f_i. In other words, for all $x_i(T_d)$ there must exist a $\{u_i\}_{T_d}^{T_d+T-1} \in \mathbb{U}^T \times \{0,1\}^{Tm_l}$ such that $\mathbf{x}_{i_{T_d}}(x_i(T_d), \{u_i\}_{T_d}^{T_d+T-1}) \in \mathbb{X}^T \times \{0,1\}^{Tn_l}, x(T|T_d) = x_{i_{er}}$. Therefore the

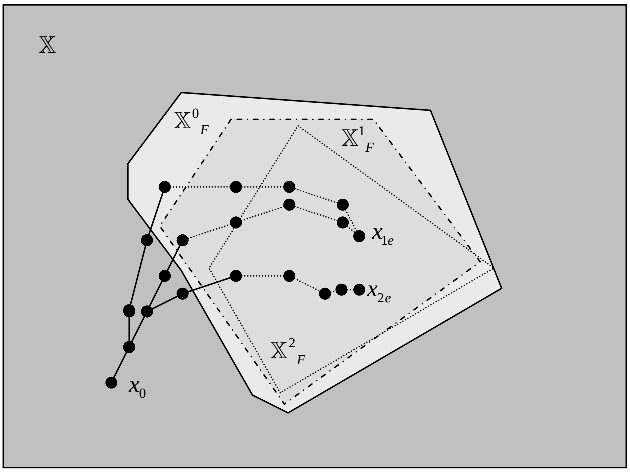

Figure 8.3: Stabilizable diagnosis of a system with 2 faults:the state space $\mathbb{X}$, Distinguishable trajectories(solid), Feasible set of MPC for the normal system $\mathbb{X}^0_F$, fault 1: $\mathbb{X}^1_F$ and fault 2: $\mathbb{X}^2_F$, trajectories to the corresponding equilibriums(dots)

whole optimization problem is a MIFT as follows:

$$\begin{cases} x_i(t|t) = x_i(t) \\ x_i(k+1|t) = A_i x_i(k|t) + B_{1_i} u(t) + B_{2_i} \delta_{_i}(k|t) + B_{3_i} z_i(k|t) \\ y_i(k|t) = C_i x_i(k|t) + D_{1_i} u(k) + D_{2_i} \delta(k|t) + D_{3_i} z(k|t) \\ E_{2_i} \delta_i(k|t) + E_{3_i} z_i(k|t) \leq E_{1_i} u_i(t) + E_{4_i} z_i(k|t) + E_{5_i} \\ i = 0, \cdots, n \\ S(T_d) = 1 \\ x_i(T_d|T_d) = x_i(T_d) \\ x_i(k+1|T_d) = A_i x_i(k|T_d) + B_{1_i} u_i(k) + B_{2_i} \delta_i(k|T_d) + B_{3_i} z_i(k|T_d) \\ y_i(k|T_d) = C_i x_i(k|T_d) + D_{1_i} u_i(k) + D_{2_i} \delta_i(k|T_d) + D_{3_i} z_i(k|T_d) \\ E_{2_i} \delta_i(k|T_d) + E_{3_i} z_i(k|T_d) \leq E_{1_i} u_i(k) + E_{4_i} z_i(k|T_d) + E_{5_i} \\ x_i(T|T_d) = x_{i_{er}} \\ x_i(k) \in \mathbb{X} \times \{0,1\}^{n_l} \\ u_i(k) \in \mathbb{U} \times \{0,1\}^{m_l} \end{cases} \tag{8.20}$$

The output of this programming is a sequence of inputs:$\{u\}_t^{t+T_d-1}$ which diagnose the fault and n sequences $\{u_i\}_{T_d}^{T_d+T-1}$ which guarantee that regardless of the condition of the system there is a sequence of input for each of them, which steers their final state, $x_i(T_d)$, to the equilibrium point. This is shown in Fig. 8.3 for a system with 2 faults.

3.2 Reconfiguration

The sequence $\{u\}_t^{t+Td-1}$ will be applied to the system. At $t = T_d$ the fault is diagnosed and then the model predictive controller is reconfigured for the faulty system simply by changing the constraints to reflect the identified fault. The cost function (8.5) is solved subject to constraints of the faulty system and the corresponding stability and input and state constraints:

$$\begin{aligned}\min J(x_c(k),\{u_c\}_k) \triangleq & \|Q_1(y_c(k|t)-y_{c_{er}})\|_p + \|Q_2(x_c(k|t)-x_{c_{er}})\|_p \\ & +\|Q_3(u_c(k|t)-u_{c_{er}})\|_p + \|Q_4(\delta_c(k|t)-\delta_{c_{er}})\|_p + \\ & \|Q_5(z_c(k|t)-z_{c_{er}})\|_p \end{aligned} \tag{8.21}$$

$$s.t. \begin{cases} x_c(k+1|T_d) = A_c x_c(k|T_d) + B_{1_i} u_c(k) + B_{2_c}\delta_c(k|T_d) + B_{3_c} z_i(k|T_d) \\ y_i(k|T_d) = C_c x_c(k|T_d) + D_{1_c} u_c(k) + D_{2_i}\delta_c(k|T_d) + D_{3_c} z_c(k|T_d) \\ E_{2_c}\delta_i(k|T_d) + E_{3_c} z_i(k|T_d) \leq E_{1_c} u_c(k) + E_{4_c} z_i(k|T_d) + E_{5_c} \\ x_c(T|T_d) = x_{c_{er}} \\ x_c(k) \in \mathbb{X} \times \{0,1\}^{n_l} \\ u_c(k) \in \mathbb{U} \times \{0,1\}^{m_l} \end{cases}$$

This problem has a feasible solution which is the sequence $\{u_c\}_{T_d}^{T_d+T-1}$ found in (8.20), therefore the MPC law stabilizes the system.

4 Example

The proposed method is tested on the two tank system shown in Fig. 8.4. The system consists of two cylindrical tanks with cross sectional area A. These two tanks are connected by two pipes at the bottom and at level h_v. The flows through the pipes, denoted by $Q_{12}V_{12}$ and $Q_{12}V_1$, are controlled using two on/off valves V_{12} and V_1. There is a flow Q_1 through a pump to tank 1 which is a continuous input.

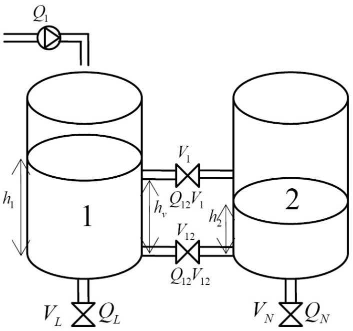

Figure 8.4: Two-tank system

Dynamical equations of the system are as follows:

$$\dot{h}_1 = \frac{1}{A}(Q_1 - Q_{12}V_{12} - Q_{12}V_1 - Q_L), \tag{8.22}$$

$$\dot{h}_2 = \frac{1}{A}(Q_{12}V_{12} + Q_{12}V_1 - Q_N), \tag{8.23}$$

where h_1 and h_2 denote the levels of tanks 1 and 2 respectively. The flow $Q_{12}V_{12}$ is described by:

$$Q_{12}V_{12} = V_{12}k_{12}sign(h_1-h_2)\sqrt{2g\,|h_1-h_2|}, \tag{8.24}$$

where g is the gravity constant and k_{12} is a valve specific constant. Similarly $Q_L = V_L k_L \sqrt{2gh_1}$ and $Q_N = V_N k_N \sqrt{2gh_2}$. The flow through valve V_1 is given by:

$$\begin{aligned} Q_{12}V_1 \quad &= \quad V_1 k_1 sign(max\{h_v, h_1\} - max\{h_v, h_2\}) \\ &\quad \sqrt{|2g(max\{h_v, h_1\} - max\{h_v, h_2\})|} \end{aligned} \tag{8.25}$$

The MLD model of the system is derived as follows (For details see [27].). The nonlinear relation $\sqrt{x}$ is approximated by a straight line x, thus (8.24) becomes:

$$Q_{12}V_{12} = V_{12}k_{12}(h_1 - h_2) \tag{8.26}$$

The auxiliary continuous variable $z_{12} = V_{12}(h_1 - h_2)$ is introduced to transform the above nonlinear equation to the linear equation $Q_{12}V_{12} = k_{12}z_{12}$ with a set of mixed integer linear inequalities. For Q_N and Q_L, using the same method, we will have $Q_N = k_N z_N$ and $Q_L = k_N z_L$ where $z_N = V_N h_2$ and $z_L = V_L h_2$.

In order to transform (8.25) to a linear equation in the MLD framework, first we introduce the following binary variables indicating whether the level in each tank has reached h_v:

$$[\delta_{01}(t) = 1] \leftrightarrow [h_1(t) \geq h_v] \tag{8.27}$$

$$[\delta_{02}(t) = 1] \leftrightarrow [h_2(t) \geq h_v] \tag{8.28}$$

and then the term $max\{h_v, h_1\} - max\{h_v, h_2\}$ is transformed into a linear equation as $Q_{12}V_1 = k_1 z_1$, where

$$z_1 = V_1(z_{01} - z_{02}) \tag{8.29}$$

$$z_{01} = \delta_{01}(h_1 - h_v) \tag{8.30}$$

$$z_{02} = \delta_{02}(h_2 - h_v) \tag{8.31}$$

are introduced auxiliary continuous variables.

Finally, differential equations (8.22), (8.23) are discretized in time by Euler approximation $\dot{h}_i(t) \approx \frac{h_i(t+1) - h_i(t)}{Ts}$, where T_s is the sample time. The final MLD model of the system consists of two continuous states: h_1, h_2, 2 binary inputs: V_1, V_{12}, 1 continuous input: Q_1 and two continuous outputs: h_1, h_2, 2 auxiliary binary variables: δ_0, δ_1 and 5 auxiliary continuous variables: $z_{01}, z_{02}, z_1, z_N, z_L$.

5 Simulation Results

In this section, the proposed active diagnosis method is used for sanity check of the upper valve V_1. We assume that the upper valve is stuck in the ON position. It is assumed that the valve V_L is always closed and V_N is always open and a sample time of 10 seconds is considered. To obtain an MLD model of the two tanks system we use HYSDEL (hybrid system description language), [28], which is a modeling language for Discrete Hybrid Automata (DHA). Given a description of the system, HYSDEL translates it into different computational models like MLD or PWA.

First we consider the case of commissioning. We assumed that at the beginning both tanks are empty *i.e.* $h_1 = h_2 = 0$. The proposed method is applied to check whether the valve is faulty or normal. We first test the minimum time active diagnosis method. The minimum length of diagnosis depends on d, which is the separation distance required in the output for successful diagnosis. The variable d is assumed as 0.01 and we require h_2 and h_{2f} to be separated. The minimum length for diagnosis is 5 sampling time. The result is depicted in Fig. 8.5. As it was expected the diagnosis strategy is to close both valves V_1, V_{12} and open Q_1 to the maximum, such that level in tank 1 will reach h_v as soon as possible. Then for the faulty system, V_1 is always open. Therefore there would be a flow to tank 2 and h_2 will increase. But if the system is normal this flow is zero and h_2 will be zero.

As d grows the minimum length and the size of optimization problem and therefore the computational time will increase. For $d = 0.04$, the minimum lengths is 7. The result is shown in Fig. 8.6. We assume that the control aim is to keep $h_2 = 0.2$. When the fault is diagnosed, the system must be reconfigured and controlled. To find the corresponding steady states for $h_2 = 0.2$, we solve (8.19) and (8.20) for the normal and faulty system with $h_{2r} = 0.2$. The corresponding values are $h_1 = 0.255$ for the normal system and $h_{1f} = 0.227$ for the faulty system. The condition of the system is determined at $t = 8$. Then the MPC constraints are updated based on the condition of the system and the system is controlled toward the equilibrium point by the MPC controller. The result for both conditions is depicted in Fig. 8.7. As one can see, the system states are steered to the equilibrium states.

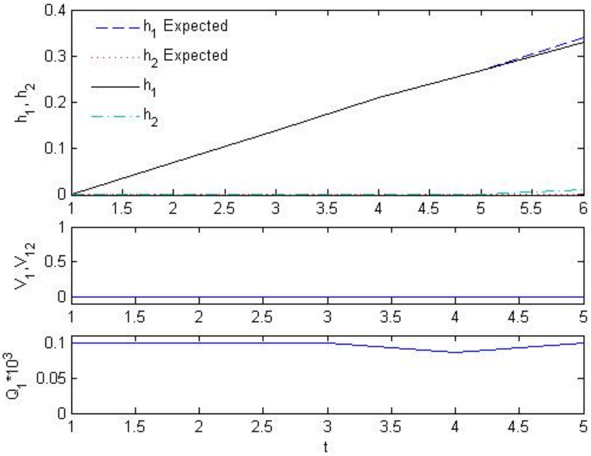

Figure 8.5: Actual versus expected output of the system, Middle: Binary inputs: V_1 dashes and V_{12} solid, Bottom: Continuous input Q_1

As mentioned in the introduction, another application of the method is when the faulty system and the normal system have the same behaviours. Assume that the system is controlled with the reference $h_2 = 0.2$ as in Fig. 8.8. As one can see, because the equilibrium value for V_1 is 1, the output of the system in the normal and the faulty condition is exactly the same. In this situation if a stuck ON fault happens, no passive diagnoser would be able to diagnose it, while the active diagnoser proposed here is capable of detecting this

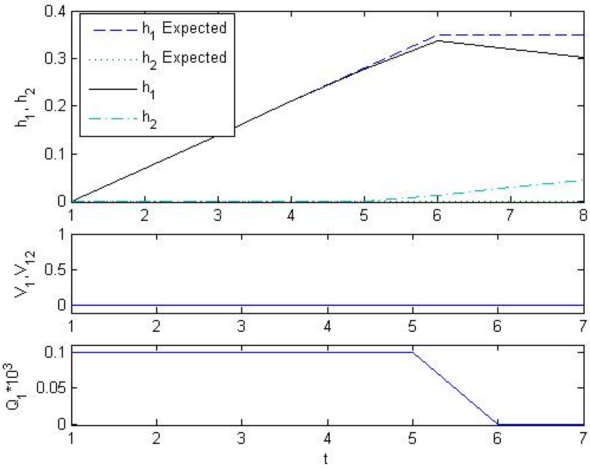

Figure 8.6: Actual versus expected output of the system, Middle: Binary inputs: V_1 dashes and V_{12} solid, Bottom: Continuous input Q_1

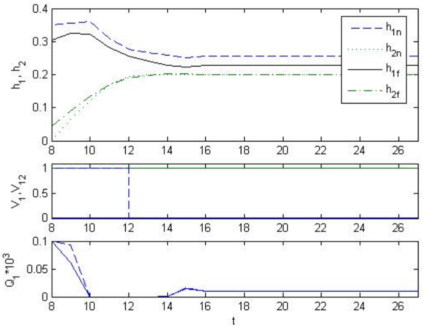

Figure 8.7: Reconfiguration of the system, Middle: Binary inputs, Bottom: Continuous input Q_1

fault by exciting the system. In this case the method can be used periodically to detect the fault.

We apply the method $t = 200$ sec. d is considered as 0.04 and the system is diagnosed in 4 sample times. Then it is reconfigured in the case of fault or controlled if it is normal with the MPC. The result for both normal and faulty condition is shown in Fig. 8.9.

6 Conclusion

In this paper a new method for active diagnosis of hybrid systems is presented. The active diagnosis problem is reformulated as a mixed integer optimization problem using the MLD framework. The method guarantees that the excitation signal does not destabilize

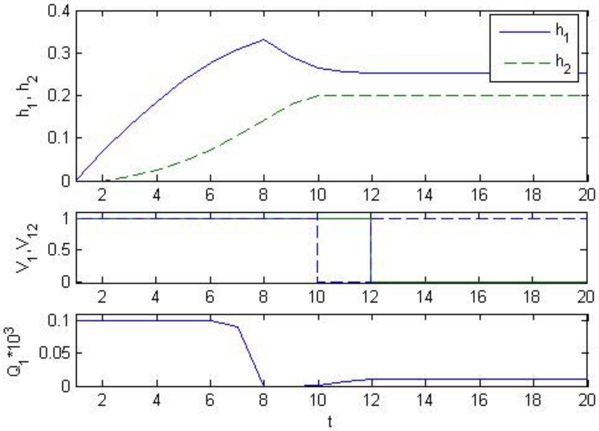

Figure 8.8: Top:h_1 and h_2, Middle:discrete inputs: V_1(dashed), V_{12}(solid)Bottom:continuous input Q_1

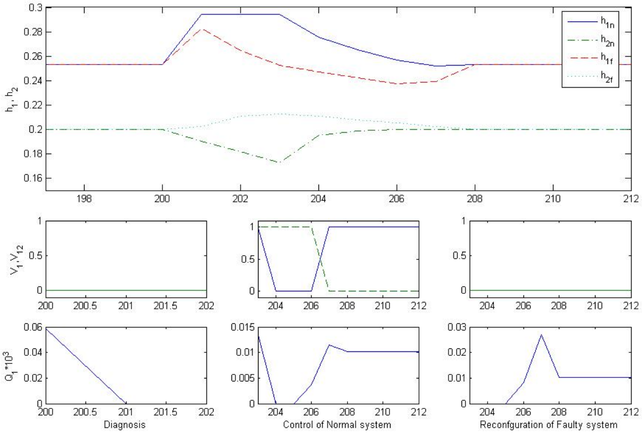

Figure 8.9: Active fault diagnosis and reconfiguration of the system. Top: output of the system: normal and faulty, Bottom: (left) Diagnosis inputs, (middle) Control input for the normal system, (right): reconfiguration input for the faulty system

the system. It guarantees that the system is reconfigurable, with the aim of fault hiding or set point redesign, if it is faulty. When the fault is diagnosed, it is reconfigured by updating the MPC constraints.

References

[1] P. Antsaklis and X. Koutsoukos, “Hybrid systems: Review and recent progress,” in *Software-Enabled Control*, T. Samad and G. Balas, Eds. IEEE Press, 2003, pp. 271–298.

[2] H. Yang, B. Jiang, and V. Cocquempot, “Fault tolerant control in hybrid systems: A brief survey,” in *Safeprocess09*, 2009.

[3] S. Narasimhan and G. Biswas, “Model-based diagnosis of hybrid systems,” *IEEE transactions on man and cybernetics*, vol. 37, no. 3, pp. 347–361, 2007.

[4] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, and P. Cheung, “Monitoring and fault diagnosis of hybrid systems,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, pp. 1225–1240, 2005.

[5] R. Nikoukhah, S. L. Campbell, K. G. Horton, and F. Delebecque, “Auxiliary signal design for robust multimodel identification,” *IEEE Transactions on Automatic Control*, vol. 47, no. 1, pp. 158–164, 2002.

[6] S. L. Campbell, K. G. Horton, and R. Nikoukhah, “Auxiliary signal design for rapid multi-model identification using optimization,” *Automatica*, vol. 38, no. 8, pp. 1313–1325, 2002.

[7] R. Nikoukhah, S. L. Campbell, A. Savkin, and R. Selmic, “A multi-model approach to failure detection in uncertain sampled-data systems,” *European journal of control*, vol. 11, no. 3, pp. 255–268, 2005.

[8] R. Nikoukhah and S. L. Campbell, “Auxiliary signal design for active failure detection in uncertain linear systems with a priori information,” *Automatica*, vol. 42, no. 2, pp. 219–228, 2006.

[9] D. Choe, S. L. Campbell, and R. Nikoukhah, “Optimal piecewise-constant signal design for active fault detection,” *International Journal of Control*, vol. 82, no. 1, pp. 130–146, 2009.

[10] H. H. Niemann and N. K. Poulsen, “Active fault diagnosis in closed-loop systems,” in *Proceedings of the 16th IFAC World Congress*, 2005.

[11] H. H. Niemann, “A setup for active fault diagnosis,” *IEEE Transactions on Automatic Control*, vol. 51, no. 9, pp. 1572–1578, 2006.

[12] N. K. Poulsen and H. H. Niemann, “Active fault diagnosis based on stochastic tests,” *International Journal of Applied Mathematics and Computer Science*, vol. 18, no. 4, pp. 487–496, 2008.

[13] X. Zhang, *Auxiliary signal design in fault detection and diagnosis*. Springer Verlag, 1989.

[14] S. Campbell and R. Nikoukhah, *Auxiliary Signal Design for Failure Detection*. Princeton University Press, 2004.

[15] M. J. Daigle and G. Biswas, “Improving diagnosability of hybrid systems through active diagnosis,” in *Safeprocess09*, 2009, pp. 217–222.

[16] S. Tabatabaeipour, A. P. Ravn, R. Izadi-Zamanabadi, and T. Bak, “Active fault diagnosis of linear hybrid systems,” in *Safeprocess09*, 2009, pp. 211–216.

[17] S. Tabatabaeipour, R. Izadi-Zamanabadi, T. Bak, and A. P. Ravn, “Automatic sensor assignment of a supermarket refrigeration system,” in *IEEE Multi-Conference on Control Applications, (CCA) & Intelligent Control, (ISIC),*, July 2009, pp. 1319–1324.

[18] S. Tabatabaeipour, A. P. Ravn, R. Izadi-Zamanabadi, and T. Bak, “Active fault diagnosis-a model predictive approach,” in *IEEE ICCA’09*, 2009.

[19] Y. M. Zhang and J. Jiang, “Issues on integration of fault diagnosis and reconfigurable control in active fault-tolerant control systems,” in *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processed*, 2006, pp. 1513–1524.

[20] J. Stoustrup, “An observer parameterization approach to active fault diagnosis with applications to a drag racing vehicle,” in *Safeprocess09*, 2009, pp. 591–596.

[21] S. Tabatabaeipour, A. P. Ravn, R. Izadi-Zamnabadi, and T. Bak, “Active diagnosis of MLD systems using distinguishable steady outputs,” in *IEEE International Symposium on Industrial Electronics*, 2010.

[22] A. Bemporad and M. Morari, “Control of systems integrating logic, dynamics, and constraints,” *Automatica*, vol. 35, pp. 407–428, 1999.

[23] A. Bemporad, D. Mignone, and M. Morari, “Moving horizon estimation for hybrid systems and fault detection,” in *American Control Conference*, San Diego, June 1999, pp. 2471–2475.

[24] W. Heemels, B. D. Schutter, and A. Bemporad, “Equivalence of hybrid dynamical models,” *Automatica*, vol. 37, no. 7, pp. 1085–1091, 2001.

[25] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control.* Springer-Verlag, 2006.

[26] A. Balluchi, L. Benvenuti, M. D. D. Benedetto, and A. Sangiovanni-Vincentelli, “Design of observers for hybrid systems,” in *5th International Workshop on Hybrid Systems: Computation and Control.* London, UK: Springer-Verlag, 2002, pp. 76–89.

[27] D. Mignone, “Control and estimation of hybrid systems with mathematical optimization,” Ph.D. dissertation, Swiss Federal Institute of Technology, 2002.

[28] F. D. Torrisi and A. Bemporad, “HYSDEL-a tool for generating computational hybrid models for analysis and synthesis problems,” *IEEE Transactions on Control Systems Technology*, vol. 12, no. 2, pp. 235–249, 2004.

Paper E

Passive Fault-tolerant Control of Piecewise Linear Systems against Actuator Faults

Seyedmojtaba Tabatabaeipour, Roozbeh Izadi-Zamanabadi, Thomas Bak, and Anders P. Ravn

This paper is submitted to:
International Journal of Systems Science

Abstract

In this paper, we propose a new method for passive fault-tolerant control of discrete time piecewise linear systems. Actuator faults are considered. A reliable piecewise linear quadratic regulator (LQR) state feedback is designed such that it can tolerate actuator faults. A sufficient condition for the existence of a passive fault-tolerant controller is derived and formulated as the feasibility of a set of linear matrix inequalities (LMIs). The upper bound on the performance cost can be minimized using a convex optimization problem with LMI constraints which can be solved efficiently. Our result can also be used for passive fault tolerant control of discrete time switched linear systems with arbitrary switching. The approach is illustrated on a numerical example.

1 Introduction

The complexity of modern control systems is increasing. In such systems, there are many components. The functionality of the overall system depends crucially on the performance of each component. A fault in an actuator, a sensor or other components may degrade the overall performance of the system and may even lead to unacceptable loss of the system functionality. Due to increasing demand for safety and reliability, it is desirable to design control systems that can tolerate potential faults; control systems that can preserve the stability of the overall system and ensure a tolerable performance degradation in the faulty process (graceful degradation). A control system with these properties is called a Fault Tolerant Control (FTC) system. The area of fault tolerant control has attracted a lot of attentions is the past 15 years, see review papers [1], [2], [3], [4], [5] and books [6] and [7].

FTC systems are either passive (PFTC) or active (AFTC). In AFTC systems, a fault is detected and diagnosed by a fault detection and diagnosis (FDD) scheme. Then the controller is redesigned or reconfigured in the case of severe faults. Control reconfiguration considers the problem of changing the control law or the controller structure by selecting a new set of inputs and outputs. After choosing the new configuration, new control parameters should be found such that the new controller can achieve the original system performance, if it is possible, or at least ensure a tolerable performance degradation in the faulty process, see [7].

In a PFTC system, the controller does not react to the occurrence of a fault. The structure and the parameter of the controller are fixed and designed such that the system can tolerate a set of faults without any change. The advantage of PFTC scheme can be explained as follows. When a fault occurs, it takes some time for the FDD module to detect the fault and to isolate and identify the fault. During this period, the system is working with the controller that is designed for the normal system. The performance of the system in this period is mainly dependent on the severity of the fault and the robustness of the nominal controller. It is clear that the controlled system may become unstable in this period, see [8]. For safety-critical systems, e.g. aircraft flight control or nuclear power plants, when a fault occurs, the time window in which the system remains stabilizable is too small to perform an accurate fault isolation and estimation. In these systems a PFTC system or a reliable control is useful because it does not need a FDD scheme.

The area of PFTC or reliable control systems has attracted considerable attention in recent years. [9] presents a method for the design of a reliable linear quadratic state feedback control such that it can tolerate actuator outages. The method also provides a guaranteed upper bound on the performance index despite actuator outages. Reliable control using redundant controllers is addressed in [10]. In [11], sensor and actuator faults are modeled by a scaling factors and a disturbance. The proposed reliable method provides a guaranteed H_∞ performance.

Reliable H_∞ control for nonlinear systems is presented in [12] where only actuator outage are considered. The authors in [13], also consider the partial degradation of actuators.

In recent years there has been a growing interest in hybrid systems. Generally speaking, a hybrid system is a dynamical system with both continuous and discrete behaviors and non-trivial interaction between continuous evolutions and discrete transitions. Among different classes of hybrid systems, PFTC is mainly studied for switched systems and piecewise linear systems. In [14], a sufficient condition for a class of switched nonlinear systems is derived such that the controlled system with actuator failures is stable with a H_∞ norm bound. For uncertain nonlinear switched systems with delay, a reliable L_∞ method is proposed in [15].

PieceWise Linear (PWL) systems, are a class of hybrid system which can approximate nonlinear system efficiently. They also arise in any practical system that contains PWL components such as dead-zones, saturation, hysteresis, etc. PFTC for PWL continuous time systems using state feedback is presented in [16]. The approach uses common Lyapunov functions. A common Lyapunov function may not always exist. In [17], a guaranteed cost control for uncertain PWL continuous time systems using output feedback is proposed. The problem is reformulated as the feasibility of a set of Bilinear Matrix Inequalities (BMIs), which are NP-hard and computationally expensive to solve globally. The non-convex optimization problem is solved using a method that combines genetic algorithm and semi definite programming. The paper, assumes that the plant and the controller always switch from the same region to the same region at the same time. In other words, the controller does not switch based on the estimated states but based on the real state of the system which are not available. This is not a realistic assumption.

Recent control systems are mainly implemented through computers. To implement a continuous time controller in a computer, one needs to emulate the designed continuous time controller as a discrete time controller. This is not a trivial step and is a subject of research. Moreover, stability analysis and control synthesis of discrete time systems have two major differences with that of continuous time systems, see [18]. Firstly, in the continuous time, only continuous Lyapunov function are allowed, while in the discrete time they can be discontinuous. Secondly, in the discrete time, a transition between non-adjacent regions may occur. [19], studies the problem of robust stability of autonomous discrete time piecewise affine systems, but the case of controller design is not addressed. In this paper, we consider the problem of PFTC for PWL discrete time systems. We use piecewise quadratic Lyapunov functions to derive a sufficient condition for the existence of a piecewise linear state feedback controller that stabilizes the system asymptotically and can tolerate loss of efficiency in actuators. A Quadratic cost function is considered as a performance index for the closed loop system. The approach provides an upper bound on a given performance index. This is cast as the feasibility of a set of LMIs which can be solved numerically using available software like YALMIP, see [20]. The optimal upper bound can be obtained by solving a convex optimization problem with LMI constraints if the initial condition is given or if it is considered as a random variable distributed in a bounded region.

2 Piecewise linear systems and actuator fault models

2.1 Piecewise Linear Systems

We consider a piecewise linear discrete time system of the following form:

$$x(k+1) = A_i x(k) + B_i u(k) \quad for\; x \in \mathcal{X}_i, \tag{9.1}$$

where $x(k) \in \mathbb{R}^n$ is the state and $u(k) \in \mathbb{R}^m$ is the control input. $\{\mathcal{X}_i\}_{i=1}^{s} \subseteq \mathbb{R}^n$ denotes a partition of the state into a number of polyhedral regions $\mathcal{X}_i, i \in \mathcal{I} = \{1, \cdots, s\}$. Each polyhedral region is represented by:

$$\mathcal{X}_i = \{x | H_i x \leq h_i\} \tag{9.2}$$

All possible switchings from region $\mathcal{X}_i$ to $\mathcal{X}_j$ are represented by the set $\mathcal{S}$:

$$\mathcal{S} := \{(i,j) | x(k) \in \mathcal{X}_i, x(k+1) \in \mathcal{X}_j\} \tag{9.3}$$

The set $\mathcal{S}$ can be computed using reachability analysis for MLD systems, see [21].

2.2 Fault Model

In this work, we consider actuator faults. Let u_j denote the $j'th$ actuator and u_j^F the failed $j'th$ actuator. We model a loss of gain in an actuator as:

$$u_j^F = (1-\alpha_j)u_j, \;\; 0 \leq \alpha_j \leq \alpha_{M_j}, \tag{9.4}$$

where α_j is the percentage of failure in the $j'th$ actuator, α_{M_j} is the maximum loss in the $j'th$ actuator. $\alpha_j = 0$ presents the case of no fault in the jth actuator, $0 < \alpha_j < 1$ corresponds to the partial loss of it, and $\alpha_j = 1$ corresponds to complete loss of it.

We define α as

$$\alpha = diag\{\alpha_1, \alpha_2, \ldots, \alpha_m\}. \tag{9.5}$$

Then

$$\mathbf{u}^F = \Gamma \mathbf{u}, \tag{9.6}$$

where $\Gamma = (I - \alpha)$. The PWL model of the system with the loss of gain in actuators can be describes by:

$$x(k+1) = A_i x(k) + B_i \Gamma_i u(k) \quad for\; x \in \mathcal{X}_i, \tag{9.7}$$

3 State Feedback Design for PWL systems

3.1 Piecewise Quadratic Stability

The problem of piecewise linear state feedback design is to design a state feedback of the form:

$$u(k) = K_i x(k) \;\; for \; x(k) \in \mathcal{X}_i \tag{9.8}$$

such that the closed loop piecewise linear system

$$x(k+1) = \mathcal{A}_i x(k), \tag{9.9}$$

where $\mathcal{A}_i = A_i + B_i K_i$, is exponentially stable.

Theorem 9.1. *([21]) The system in (9.9) is exponentially stable if there exist matrices $P_i = P_i^T > 0$, $\forall i \in \mathcal{I}$, such that the positive definite function $V(x(k)) = x^T(k)P_i x(k)$, $\forall x \in \mathcal{X}_i$ satisfies $V(x(k+1)) - V(x(k)) < 0$.*

The piecewise quadratic Lyapunov function in Theorem 9.1 can be computed by solving the following LMIs:

$$\mathcal{A}_i P_j \mathcal{A}_i - P_i < 0, \;\; \forall (i,j) \in \mathcal{S} \tag{9.10}$$

$$P_i = P_i^T > 0, \;\; \forall i \in \mathcal{I} \tag{9.11}$$

3.2 PWL Quadratic Regulator (PWLQR)

The quadratic cost function associated with the system is:

$$J = \sum_{k=0}^{\infty} x^T(k)Qx(k) + u^T(k)Ru(k), \tag{9.12}$$

where $Q \geq 0$ and $R \geq 0$ are given weighting matrices of appropriate dimensions.

Lemma 9.1. ***Upper bound on the performance cost:*** *The system in (9.1) with the controller in (9.8) satisfies the following upper bound on the performance cost*

$$J \leq x(0)^T P_{i_0} x(0) \tag{9.13}$$

with $x(0) \in \mathcal{X}_{i_0}$, i.e. i_0 is the index of the initial region, if there exist matrices $P_i = P_i^T > 0$, $\forall i \in \mathcal{I}$ such that

$$(A_j + B_j K_j)^T P_i (A_j + B_j K_j) - P_j + Q + K_j^T R K_j < 0, \;\; \forall (i,j) \in \mathcal{S} \tag{9.14}$$

Proof. Pre and post-multiplying (9.14) by $x^T(k)$ and $x(k)$ we have:

$$\begin{gathered} x^T(k)(A_j + B_j K_j)^T P_i (A_j + B_j K_j) x(k) - \\ x^T(k) P_j x(k) + x^T(k) Q x(k) + x^T(k) K_j^T R K_j x(k) < 0 \end{gathered} \tag{9.15}$$

as $x(k+1) = (A_j + B_j K_j)x(k)$, it implies:

$$V(x(k+1)) - V(x(k)) + x^T(k)Qx(k) + u^T(k)Ru(k) < 0 \tag{9.16}$$

Summing up the above equation from $k = 0$ to $k = \infty$ we have:

$$V(x(\infty)) - V(x(0)) + \Sigma_0^{\infty}(x^T(k)Qx(k) + u^T(k)Ru(k)) < 0 \tag{9.17}$$

As $V(x(\infty)) = 0$ and $V(x(0)) = x(0)^T P_{i_0} x(0)$ therefore we have:

$$\sum_{k=0}^{\infty}(x^T(k)Qx(k) + u^T(k)Ru(k)) < x^T(0)P_{i_0}x(0).$$

□

The inequality (9.14) is a nonlinear matrix inequality and difficult to solve. In the following, an LMI equivalent of it is presented.

Lemma 9.2. *([22]) Let $Z_j = Z_j^T > 0$ and G_j be invertible for all $j \in \mathcal{I}$. Then (9.14) is equivalent to the following LMI:*

$$\begin{bmatrix} G_j + G_j^T - Z_j & G_j^T & Y_j^T & (A_jG_j + B_jY_j)^T \\ G_j & Q^{-1} & 0 & 0 \\ Y_j & 0 & R^{-1} & 0 \\ (A_jG_j + B_jY_j) & 0 & 0 & Z_i \end{bmatrix} > 0 \tag{9.18}$$

The feedback gains are given by $K_j = Y_j G_j^{-1}$.

Before presenting the main result of this paper in the next section, the following lemma is introduced.

Lemma 9.3. *([23]) Let M, N, H be real matrices. If $H^T H \leq I$, then for every scalar $\epsilon > 0$ the following inequality holds:*

$$MHN + N^T H^T M^T \leq \epsilon M M^T + \epsilon^{-1} N^T N. \tag{9.19}$$

3.3 Passive fault-tolerant Control

Definition 9.1. A piecewise linear control law of the form (9.8) is a passive fault-tolerant guaranteed cost control for the system (9.1) and the performance function (9.12) if the following matrix inequality is satisfied:

$$(A_j+B_j\Gamma_jK_j)^T P_i(A_j+B_j\Gamma_jK_j)-P_j+Q+K_j^T\Gamma_j^T R\Gamma_jK_j < 0, \ \ \forall(i,j) \in \mathcal{S}. \tag{9.20}$$

The PWL system with the law obtained from solving (9.20) is quadratically stable and for every admissible α, the performance function satisfies:

$$J \leq x^T(0)P_{i_0}x(0).$$

Note that the inequality (9.20) is not linear in terms of variables P_i and K_i. In the following theorem, two equivalent LMI formulations are provided as sufficient conditions for (9.20).

Theorem 9.2. *There exist a passive-fault tolerant control law for the PWL system (9.1) with the performance function (9.12) if there exist symmetric matrices $Z_j = Z_j^T > 0$, and invertible matrices G_j, and matrices Y_j and positive scalars $\epsilon_j > 0, j \in \mathcal{I}$ such that the* *following LMI is satisfied:*

$$\begin{bmatrix} -\epsilon_j I & \alpha_j Y_j & 0 & 0 & 0 \\ Y_j^T \alpha_j & Z_j - G_j - G_j^T & G_j^T & Y_j^T & (A_j G_j + B_j Y_j)^T \\ 0 & G_j & Q^{-1} & 0 & 0 \\ 0 & Y_j & 0 & R^{-1} + \epsilon_j I & \epsilon_j B_j^T \\ 0 & (A_j G_j + B_j Y_j) & 0 & \epsilon_j B_j & Z_i + \epsilon_j B_j^T B_j \end{bmatrix} < 0, \ \ \forall (i,j) \in \mathcal{S} \tag{9.21}$$

or if there exist symmetric matrices $X_j = X_j^T > 0$ *and matrices* Y_j *and positive scalars* $\epsilon_j > 0, \ j \in \mathcal{I}$ *such that*

$$\begin{bmatrix} -X_i + \epsilon_j B_j^T B_j & \epsilon_j B_j & (A_j X_j + B_j Y_j) & 0 & 0 \\ \epsilon_j B_j^T & -R^{-1} + \epsilon_j I & Y_j & 0 & 0 \\ (A_j X_j + B_j Y_j)^T & Y_j^T & -X_j & X_j & \epsilon_j Y_j^T \\ 0 & 0 & X_j & -Q^{-1} & 0 \\ 0 & 0 & \epsilon_j Y_j & 0 & -\epsilon_j I \end{bmatrix} < 0, \ \ \forall (i,j) \in \mathcal{S} \tag{9.22}$$

Then the piecewise linear feedback gains are given by:

$$K_i = Y_i G_i^{-1} \tag{9.23}$$

with Y_i *and* G_i *from solving (9.21) or*

$$K_i = Y_i X_i^{-1} \tag{9.24}$$

with Y_i *and* X_i *from solving (9.22) and the performance function of the closed loop system (9.9) satisfies:*

$$J \leq x^T(0) Z_{i_0}^{-1} x(0) \tag{9.25}$$

or

$$J \leq x^T(0) X_{i_0}^{-1} x(0). \tag{9.26}$$

Proof. Using Schur complement, inequality (9.20) is equivalent to:

$$\begin{bmatrix} -P_j & I & K_j^T \Gamma_j^T & (A_j + B_j \Gamma_j K_j)^T \\ I & Q^{-1} & 0 & 0 \\ \Gamma_j K_j & 0 & R^{-1} & 0 \\ (A_j + B_j \Gamma_j K_j) & 0 & 0 & P_i^{-1} \end{bmatrix} < 0 \tag{9.27}$$

Substituting $\Gamma_j = I - \alpha_j$, implies that the left side of (9.27) is equal to:

$$\begin{bmatrix} -P_j & I & K_j^T & (A_j + B_j K_j)^T \\ I & Q^{-1} & 0 & 0 \\ K_j & 0 & R^{-1} & 0 \\ (A_j + B_j K_j) & 0 & 0 & P_i^{-1} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ I \\ B_j \end{bmatrix} \begin{bmatrix} \alpha_j K_j & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} K_j^T \alpha_j \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & I & B_j^T \end{bmatrix} \tag{9.28}$$

Using Lemma 9.3 with $H = -I$, it follows that:

$$(9.28) \leq (*) + \epsilon_j \begin{bmatrix} 0 \\ 0 \\ I \\ B_j \end{bmatrix} \begin{bmatrix} 0 & 0 & I & B_j^T \end{bmatrix} + \epsilon_j^{-1} \begin{bmatrix} K_j^T \alpha_j \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \alpha_j K_j & 0 & 0 & 0 \end{bmatrix}, \tag{9.29}$$

where $(*)$ is the first matrix in (9.28). We have $\alpha_j \leq \alpha_{M_j}$, therefore it holds that:

$$(9.29) \leq \begin{bmatrix} -P_j + \epsilon_j^{-1} K_j^T \alpha_{M_j} \alpha_{M_j} K_j & I & K_j^T & (A_j + B_j K_j)^T \\ I & Q^{-1} & 0 & 0 \\ K_j & 0 & R^{-1} + \epsilon_j I & \epsilon_j B_j^T \\ (A_j + B_j K_j) & 0 & \epsilon_j B_j & P_i^{-1} + \epsilon_j B_j^T B_j \end{bmatrix} \tag{9.30}$$

Pre- and post multiplying the right side of (9.30) by $diag\{G_j^T, I, I, I\}$ and $diag\{G_j, I, I, I\}$ and using Schur complement we get:

$$\begin{bmatrix} -\epsilon_j I & \alpha_j K_j G_j & 0 & 0 & 0 \\ G_j^T K_j^T \alpha_j & -G_j^T P_j G_j & G_j^T & G_j^T K_j^T & (A_j G_j + B_j K_j G_j)^T \\ 0 & G_j & Q^{-1} & 0 & 0 \\ 0 & K_j G_j & 0 & R^{-1} + \epsilon_j I & \epsilon_j B_j^T \\ 0 & (A_j G_j + B_j K_j G_j) & 0 & \epsilon_j B_j & P_i^{-1} + \epsilon_j B_j^T B_j \end{bmatrix} \tag{9.31}$$

Using the fact that $G_j^T P_j G_j \geq G_j + G_j^T - P_j^{-1}$, and replacing $Y_j = K_j G_j$ we derive the LMI in (9.21) as a sufficient condition for (9.20).

To show (9.22), a very similar procedure is used. Using Schur complement, inequality (9.20) is equivalent to:

$$\begin{bmatrix} -P_i^{-1} & 0 & (A_j + B_j \Gamma_j K_j) \\ 0 & -R^{-1} & \Gamma_j K_j \\ (A_j + B_j \Gamma_j K_j)^T & K_j^T \Gamma_j & Q_j - P_j \end{bmatrix} < 0 \tag{9.32}$$

$\Gamma_j = I - \alpha_j$, therefore the right hand of the above inequality is equal to:

$$\begin{bmatrix} -P_i^{-1} & 0 & (A_j + B_j K_j) \\ 0 & -R^{-1} & K_j \\ (A_j + B_j K_j)^T & K_j^T & Q_j - P_j \end{bmatrix} - \begin{bmatrix} B_j \\ I \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & \alpha_j K_j \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ K_j^T \alpha_j \end{bmatrix} \begin{bmatrix} B_j^T & I & 0 \end{bmatrix} \tag{9.33}$$

Using Lemma 9.3, implies:

$$(9.33) \leq \begin{bmatrix} -P_i^{-1} & 0 & (A_j + B_j K_j) \\ 0 & -R^{-1} & K_j \\ (A_j + B_j K_j)^T & K_j^T & Q_j - P_j \end{bmatrix} + \epsilon_j \begin{bmatrix} B_j \\ I \\ 0 \end{bmatrix} \begin{bmatrix} B_j^T & I & 0 \end{bmatrix} + \epsilon_j^{-1} \begin{bmatrix} 0 \\ 0 \\ K_j^T \alpha_j \end{bmatrix} \begin{bmatrix} 0 & 0 & \alpha_j K_j \end{bmatrix} \tag{9.34}$$

We have $\alpha_j \leq \alpha_{M_j}$, therefore it holds that:

$$(9.34) \leq \begin{bmatrix} -P_i^{-1} + \epsilon_j B_j B_j^T & \epsilon_j B_j & (A_j + B_j K_j) \\ \epsilon_j B_j^T & -R^{-1} + \epsilon_j I & K_j \\ (A_j + B_j K_j)^T & K_j^T & Q - P_j + \epsilon_j^{-1} K_j^T \alpha_{M_j} \alpha_{M_j} K_j \end{bmatrix} \tag{9.35}$$

We pre- and post-multiply the right hand of the above inequality by $diag\{I, I, P_j^{-1}\}$ and its transpose. Then, we get:

$$\begin{bmatrix} -P_i^{-1} + \epsilon_j B_j B_j^T & \epsilon_j B_j & (A_j P_j^{-1} + B_j K_j P_j^{-1}) \\ \epsilon_j B_j^T & -R^{-1} + \epsilon_j I & K_j P_j^{-1} \\ (A_j P_j^{-1} + B_j K_j P_j^{-1})^T & P_j^{-1} K_j^T & P_j^{-1} Q P_j^{-1} - P_j^{-1} + \epsilon_j^{-1} P_j^{-1} K_j^T \alpha_{M_j} \alpha_{M_j} K_j P_j^{-1} \end{bmatrix} \tag{9.36}$$

as a sufficient condition for (9.20). Define $X_j = P_j^{-1}, Y_j = K_j P_j^{-1}$. By applying the Schur complement to (9.36), we conclude the LMI (9.22). □

Remark 1*: By considering the set of all possible switching as $\mathcal{S} = \mathcal{I} \times \mathcal{I}$, the result can be used to design passive fault tolerant controllers for discrete time switched linear systems with arbitrary switching.*

The upper bound of (9.12), could be minimized by the following constrained optimization problem:

$$\min_{Z_i, Y_i, G_i, \epsilon_i, t} \quad t \tag{9.37}$$

$$s.t. \begin{cases} (21) \\ \begin{bmatrix} -t & x(0)^T \\ x(0) & -Z_{i_0} \end{bmatrix} < 0 \end{cases}$$

Using Schur complement, the new LMI constraint is equivalent to $-t + x^T(0) Z_{i_0}^{-1} x(0) < 0$. Therefore, (9.37) is equal to $min\ x^T(0) P_{i_0} x(0)$ with the remaining constraints. The problem with the above formulation is that the upper bound is dependent on the initial state $x(0)$. To remove the dependency on the initial state, using a similar procedure to that of [17], the initial condition is considered as a random variable with uniform distribution in a bounded region $\overline{\mathcal{X}}$. Then, it is tried to minimize the expected value of the cost function. We have:

$$E(J) \leq E(tr(P_{i_0} x(0) x^T(0))) \leq \sum_{i \in \mathcal{I}} \sigma_i tr(P_i L_i), \tag{9.38}$$

where $L_i = E(x(0) x^T(0))$ is the expectation of $x(0) x^T(0)$ corresponding to $x(0) \in \mathcal{X}_i, i \in \mathcal{I}$, $tr(\cdot)$ is the trace operator and σ_i is the probability of $x(0) \in \mathcal{X}_i$. Then, the optimization problem is:

$$\min_{Z_i, Y_i, G_i} \sum_{i \in \mathcal{I}} \sigma_i tr(Z_i^{-1} L_i) \tag{9.39}$$

$$s.t. \begin{cases} (21) \\ Z_i = Z_i^T > 0 \end{cases}$$

or

$$\min_{X_i,Y_i} \sum_{i\in\mathcal{I}} \sigma_i tr(X_i^{-1}L_i) \tag{9.40}$$
$$s.t. \begin{cases} (22) \\ X_i = X_i^T > 0 \end{cases}$$

Both optimization problems in (9.39) and (9.40) are non-convex because (9.39) includes Z_i and its inverse and similarly (9.40) includes X_i and its inverse. To convert them to a convex optimization problem , we introduce new variables $V_i, i \in \mathcal{I}$, which satisfies:

$$\begin{bmatrix} V_i & I \\ I & Z_i \end{bmatrix} \geq 0. \tag{9.41}$$

Using Schur complement, the above constraint is equivalent to $Z_i^{-1} \leq V_i$. Therefore, the objective function in (9.39), which is nonlinear in term of Z_i, can be converted to $\sum_{i\in\mathcal{I}} \sigma_i tr(V_i L_i)$. Consequently, the optimization problem (9.39) can be transformed to the following optimization problem in terms of variables $Z_i, Y_i, G_i, \epsilon_i$ and the introduced variables V_i:

$$\min_{Z_i,Y_i,G_i,V_i,\epsilon_i} \sum_{i\in\mathcal{I}} \sigma_i tr(V_i L_i) \tag{9.42}$$
$$s.t. \begin{cases} (21) \\ \begin{bmatrix} V_i & I \\ I & Z_i \end{bmatrix} \geq 0, \ i \in \mathcal{I} \\ Z_i = Z_i^T > 0, \ i \in \mathcal{I} \\ V_i = V_i^T > 0, \ i \in \mathcal{I} \end{cases}$$

Similarly, (9.40) can be transformed to the following convex optimization problem:

$$\min_{X_i,Y_i,V_i,\epsilon_i} \sum_{i\in\mathcal{I}} \sigma_i tr(V_i L_i) \tag{9.43}$$
$$s.t. \begin{cases} (22) \\ \begin{bmatrix} V_i & I \\ I & X_i \end{bmatrix} \geq 0, i \in \mathcal{I} \\ X_i = X_i^T > 0, \ i \in \mathcal{I} \\ V_i = V_i^T > 0, \ i \in \mathcal{I} \end{cases}$$

Both of the above optimization problems are convex optimization problems with LMI constraints and can be solved efficiently using available softwares like YALMIP/SeDuMi or YALMIP/LMILAB, see [20].

3.4 Example

To illustrate our approach we consider the following open-loop unstable PWL system from [24]:

$$x(k+1) = \begin{bmatrix} -0.2523 & 0.4856 & 0.6467 \\ 0.5290 & -0.2616 & 0.3128 \\ -0.4415 & -0.2713 & -0.6967 \end{bmatrix} x(k) + \begin{bmatrix} 0.5656 \\ 0.5460 \\ 0.9389 \end{bmatrix} u(k) \texttt{ for } x \in \mathcal{X}_1$$

$$x(k+1) = \begin{bmatrix} 0.0647 & 0.1729 & -0.6542 \\ -0.3131 & -0.6691 & -0.6516 \\ -0.3085 & 0.0613 & 0.0099 \end{bmatrix} x(k) + \begin{bmatrix} 0.6543 \\ 0.5266 \\ -0.0558 \end{bmatrix} u(k) \texttt{ for } x \in \mathcal{X}_2$$

$$x(k+1) = \begin{bmatrix} 0.6402 & -0.5409 & -0.5629 \\ -0.6693 & -0.6874 & 0.1748 \\ -0.2812 & 0.4898 & -0.3526 \end{bmatrix} x(k) + \begin{bmatrix} 0.7580 \\ -0.8050 \\ -0.4059 \end{bmatrix} u(k) \texttt{ for } x \in \mathcal{X}_3$$

$$x(k+1) = \begin{bmatrix} -0.3501 & 0.2590 & 0.6695 \\ -0.4808 & 0.1905 & 0.3865 \\ -0.1217 & 0.2631 & -0.0013 \end{bmatrix} x(k) + \begin{bmatrix} 0.6961 \\ -0.7619 \\ 0.2590 \end{bmatrix} u(k) \texttt{ for } x \in \mathcal{X}_4$$

and the state space partition, $\{\mathcal{X}_i\}_{i=1}^{5}$, is given by $\mathcal{X}_i = \{x|H_i x > 0\}$: for $i = 1, 3$, and $\mathcal{X}_i = \{x|H_i x \geq 0\}$ for $2, 4$, where the H_i matrices are:

$$H_1 = -H_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad H_2 = -H_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{9.44}$$

The weighting matrices of the performance index (9.12) are $Q = 0.02I$ and $R = 0.01$. We use the optimization problem formulation in (9.43) to design a passive fault-tolerant controller. The system can tolerate maximum of 90% loss of the actuator ($\alpha_M = 0.9$). For losses greater than 90%, the optimization problem becomes infeasible. We assume the initial state to be a random variable which is uniformly distributed on $\bar{\mathcal{X}} = [-5, 5]^3$. The upper bound on the expected value of the cost fucntion is 4.47. The resulting PWL controller is:

$$\begin{aligned} K_1 &= \begin{bmatrix} 0.0635 & 0.0979 & 0.0549 \end{bmatrix} \\ K_2 &= \begin{bmatrix} 0.0559 & 0.0939 & 0.7037 \end{bmatrix} \\ K_3 &= \begin{bmatrix} -0.7791 & 0.1537 & 0.2302 \end{bmatrix} \\ K_4 &= \begin{bmatrix} -0.0741 & 0.0587 & -0.1366 \end{bmatrix} \end{aligned}$$

and the piecewise Lyapunov function with matrices:

$$P_1 = \begin{bmatrix} 0.1156 & -0.0505 & 0.0024 \\ -0.0505 & 0.1661 & 0.1511 \\ 0.0024 & 0.1511 & 0.2667 \end{bmatrix} \quad P_2 = \begin{bmatrix} 0.1070 & 0.0679 & 0.0313 \\ 0.0679 & 0.1332 & 0.0468 \\ 0.0313 & 0.0468 & 0.1920 \end{bmatrix}$$

$$P_3 = \begin{bmatrix} 0.2913 & -0.0466 & -0.0755 \\ -0.0466 & 0.2060 & 0.0670 \\ -0.0755 & 0.0670 & 0.1550 \end{bmatrix} \quad P_4 = \begin{bmatrix} 0.1162 & -0.0603 & -0.0928 \\ -0.0603 & 0.0781 & 0.0601 \\ -0.0928 & 0.0601 & 0.1722 \end{bmatrix}.$$

It is worthwhile to point out that the feasibility of the optimization problem depends on the choice of weighting matrices. For example, for $Q = 0.1I, R = 0.1$, the optimization problem becomes infeasible for $\alpha > 0.5$.

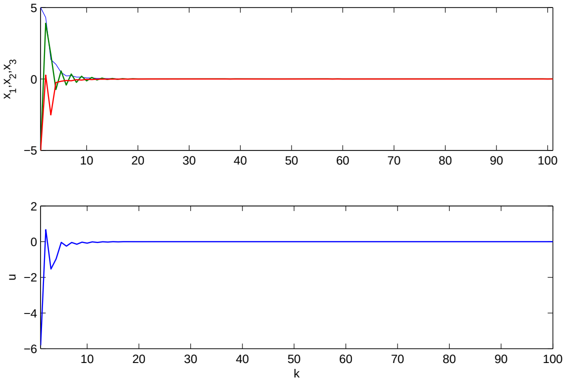

Figure 9.1: Simulation results with a controller designed to tolerate 90% loss of the actuation with the real system with $\alpha = 0$, i.e. actuator operates normally.

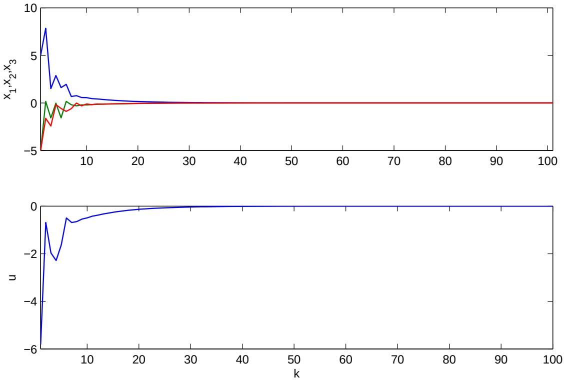

Figure 9.2: Simulation results with a controller designed to tolerate 90% loss of the actuation with the real system with $\alpha = 0.8$

Figure 9.1, 9.2, 9.3 show respectively the simulation result for the controlled system with 0%, 80% and 90% loss of actuator gains with a controller designed to tolerate $\alpha_M = 0.9$, with the initial condition $x_0 = [5\ -5\ -5]^T$. As it can be seen, the performance of the system is similar for $0 \leq \alpha \leq 0.8$, but it decreases considerably for $\alpha = 0.9$. Figure 9.4, shows how the optimal upper bound on the expected value of the cost function, $E^*(J)$, grows by increasing the maximum partial loss of the actuator, namely α_M, which is to be tolerated. As one can see, however it is possible to design a controller which tolerates up to 90% partial loss of the actuators, but the cost increases sharply for $\alpha_M >= 0.8$. Therefore, $\alpha_M = 0.8$ could be considered as a trade off between the cost to be paid and the degree of the tolerance to faults.

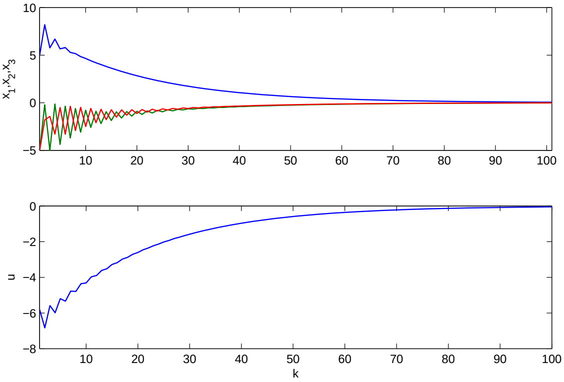

Figure 9.3: Simulation results with a controller designed to tolerate 90% loss of the actuation with the real system with $\alpha = 0.90$

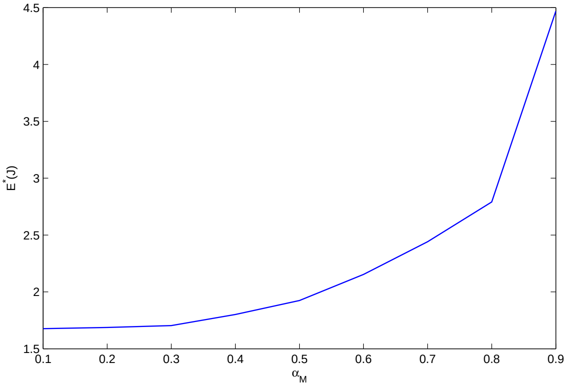

Figure 9.4: optimal expected value of cost function vs. maximum partial loss of the actuators.

4 Conclusion

We proposed an approach to passive fault-tolerant control of PWL discrete time systems. Using piecewise quadratic Lyapunov functions, a piecewise linear state feedback is designed for the closed loop system such that it can tolerate partial loss of actuator gains. The existence of the controller is reformulated as the feasibility of a set of LMIs. Two equivalent forms of the LMI condition are derived. The approach provides an upper bound on the performance cost which can be minimized using a convex minimization problem with LMI constraints. The result is illustrated on a numerical example and it is explained how one can choose a trade of between the performance of the system and the degree of the system tolerance to the partial loss of actuator gains.

References

[1] M. Blanke, R. Izadi-Zamanabadi, S. A. Bogh, and Z. P. Lunau, “Fault-tolerant control systems-a holistic view,” *Control Engineering Practice*, vol. 5, no. 5, pp. 693–702, 1997.

[2] R. J. Patton, “Fault-tolerant control systems: The 1997 situation,” in *3rd IFAC symposium on fault detection supervision and safety for technical processes*, vol. 3, 1997, pp. 1033–1054.

[3] M. Blanke, C. Frei, F. Kraus, R. J. Patton, and M. Staroswiecki, “What is fault tolerant control?” in *4th IFAC symposium on fault detection, supervision and safety for technical processes*, 2000, pp. 40–51.

[4] M. Blanke, M. Staroswiecki, and N. E. Wu, “Concepts and methods in fault-tolerant control,” in *American Control Conference*, vol. 4, 2001, pp. 2606–2620.

[5] J. Jiang, “Fault-tolerant control systems-an introductory overview,” *Acta Automatica Sinica*, vol. 31, no. 1, pp. 161–174, 2005.

[6] R. Isermann, *Fault-diagnosis systems*. Springer Verlag, 2006.

[7] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*. Springer-Verlag, 2006.

[8] Y. M. Zhang and J. Jiang, “Issues on integration of fault diagnosis and reconfigurable control in active fault-tolerant control systems,” in *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processed*, 2006, pp. 1513–1524.

[9] R. Veillette, “Reliable linear-quadratic state-feedback control,” *Automatica*, vol. 31, no. 1, pp. 137–144, 1995.

[10] G. H. Yang, Z. S. Y., J. Lam, and J. Wang, “Reliable control using redundant controllers,” *IEEE Transactions on Automatic Control*, vol. 43, no. 11, pp. 1588–1593, Nov 1998.

[11] G. Yang, J. Wang, and Y. Soh, “Reliable H_∞ controller design for linear systems,” *Automatica*, vol. 37, no. 5, pp. 717–725, 2001.

[12] G. H. Yang, J. Lam, and J. Wang, “Reliable H_∞ control for affine nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 43, no. 8, pp. 1112–1117, Aug 1998.

[13] G. Yang, J. Wang, and Y. Soh, “Reliable guaranteed cost control for uncertain nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 45, no. 11, pp. 2188–2192, 2000.

[14] R. Wang, M. Liu, and J. Zhao, “Reliable H_∞ control for a class of switched nonlinear systems with actuator failures,” *Nonlinear Analysis: Hybrid Systems*, vol. 1, no. 3, pp. 317–325, 2007.

[15] Z. Xiang and R. Wang, “Robust L_∞ reliable control for uncertain nonlinear switched systems with time delay,” *Applied Mathematics and Computation*, vol. 210, no. 1, pp. 202–210, 2009.

[16] N. Nayebpanah, L. Rodrigues, and Y. Zhang, “Fault-tolerant controller synthesis for piecewise-affine systems,” in *IEEE American Control Conference*, 2009, pp. 222–226.

[17] J. Zhang and W. Tang, “Output feedback optimal guaranteed cost control of uncertain piecewise linear systems,” *International Journal of Robust and Nonlinear Control*, vol. 19, pp. 596–590, 2009.

[18] D. Mignone, G. Ferrari-Trecate, and M. Morari, “Stability and stabilization of piecewise affine and hybrid systems: an LMI approach,” in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 1, 2000, pp. 504 –509 vol.1.

[19] M. Kantner, “Robust stability of piecewise linear discrete time systems,” in *Proceedings of American Control Conference*, vol. 2, jun 1997, pp. 1241 –1245 vol.2.

[20] J. Löfberg, “YALMIP : A toolbox for modeling and optimization in MATLAB,” in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[21] F. Cuzzola and M. Morari, “A generalized approach for analysis and control of discrete-time piecewise affine and hybrid systems,” *Hybrid Systems: Computation and Control*, vol. 2034, pp. 189–203, 2001.

[22] M. Lazar, W. Heemels, S. Weiland, and A. Bemporad, “Stabilizing model predictive control of hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 51, no. 11, pp. 1813–1818, Nov. 2006.

[23] I. Petersen, “A stabilization algorithm for a class of uncertain linear systems.” *System and Control Letters*, vol. 8, no. 4, pp. 351–357, 1987.

[24] M. Lazar, W. Heemels, S. Weiland, and A. Bemporad, “Stabilizing model predictive control of hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 51, no. 11, pp. 1813 –1818, Nov. 2006.

Paper F

Output Feedback Guaranteed Cost Control of Uncertain Discrete-time Piecewise Linear Systems

Seyedmojtaba Tabatabaeipour, Thomas Bak, and Roozbeh Izadi-Zamanabadi

This paper is submitted to:
IET Control Theory and Applications

The layout has been revised

Abstract

This paper proposes a new approach for output feedback control of uncertain discrete time piecewise linear systems. The output feedback controller is designed using piecewise quadratic Lyapunov functions. We do not make the common restrictive assumption that the controller switches to one region based on the state of the system and therefore both the controller and the system are always in the same region, but it is assumed that the controller switching is based on the estimated state. A sufficient condition for the existence of an output feedback controller is derived and formulated as the feasibility of a set of bilinear matrix inequalities (BMIs). The upper bound on the optimal cost of the controller is minimized solving an optimization problem with BMI constraints. The optimization problem is solved using the V-K iteration algorithm. The approach is illustrated on a numerical example.

1 Introduction

In recent years there has been a growing interest in hybrid systems. In general, a hybrid system is a dynamical system with both continuous and discrete behaviors and non-trivial interactions between continuous evolutions and discrete transitions. PieceWise Linear (PWL) systems, are an attractive class of hybrid systems as they can approximate nonlinear systems efficiently. They also arise in any practical system that contains PWL components such as dead-zones, saturation, hysteresis, etc.

The problem of controller design and stability analysis for PWL systems has attracted a lot of attention in recent years, e.g see [1], [2], [3], [4]. These approaches use state feedback for controller design of PWL systems. But, the states of a system are not usually available. Therefore, it is important to have an output feedback controller. The output feedback could be static or dynamic. [5], [6] investigate the problem of dynamic output feedback control for continuous time PWL systems. It is shown that the problem can be formulated as a Bilinear Matrix Inequality (BMI) problem which is solved using iterative algorithms. In their work, the observer switches based on the estimated states and not based on the measured output.

Due to modeling error or external disturbance or faults, piecewise linear systems are subject to uncertainties. [7] studies robust control of uncertain PWL systems using state feedback and continuous piecewise Lyapunov functions. In [8], a method for robust H_∞ output feedback control design for uncertain piecewise affine systems is proposed by solving a set of Bilinear Matrix Inequalities. In [9], a guaranteed cost control method using output feedback is proposed. The problem is reformulated as the feasibility of a set of BMIs. The non-convex optimization problem is solved using a method that combines genetic algorithms and semi definite programming. Both works, assume that switching of the controller is based on the real state of the system and not based on the estimated state of the system. In other words, the plant and the controller are always in the same region. This is not a realistic assumption. All of the aforementioned works are in the continuous time domain.

Recent control systems are mainly implemented through computers. To implement a continuous time controller in a computer, one needs to emulate the designed continuous time controller as a discrete time controller. This is not a trivial step and is a subject of research. Moreover, stability analysis and control synthesis of discrete time systems have two major differences with that of continuous time systems, see [10]. Firstly, in the continuous time, only continuous Lyapunov functions are allowed, while in the discrete time they can be discontinuous. Secondly, in the discrete time, a transition between non-adjacent regions may occur. In the discrete time domain, [11] proposes a dynamic output feedback method for PWL systems. This work assumes that the partitioning is on the output space and not on the state space. Therefore, the switching is based on the measured output and the controller and the system are always in the same region. The problem of static output feedback control for switched systems is addressed in [12] and for PWL systems is addressed in [13]. They formulated existence of a stabilizing static output feedback controller as the feasibility of a set of Linear matrix inequalities (LMIs). An extension of the method to incorporate H_∞ performance is also given.

The problem of robust stability of autonomous discrete time piecewise affine systems is studies in [14], but the case of controller design is not addressed. [15] proposes a robust H_∞ control approach for uncertain discrete time piecewise affine systems. They consider time varying parameter uncertainties. The approach uses state feedback and formulates the problem as linear matrix inequalities.

In this paper, we consider the problem of dynamic output feedback control for PWL discrete time systems. We consider norm bounded uncertainties. In our approach, we do not make the assumption that the switching of the controller is based on the system's state but it is assumed to be based on the estimated state, i.e. the general case that the controller might be at one region while the system is at another region is considered. We use piecewise quadratic Lyapunov functions to derive a sufficient condition for the existence of a dynamic output feedback controller. A Quadratic cost function is considered as a performance index for the closed loop system. The approach provides an upper bound on the performance cost. This is cast as the feasibility of a set of BMIs. The optimal upper bound can be obtained by solving an optimization problem with BMI constraints. To solve the optimization problem we use the V-K iteration algorithm provided by [16].

The paper is organized as follows. In section 2, the uncertain PWL model with bounded uncertainties is presented. Section 3, explains the design of a guaranteed cost piecewise LQR state feedback control for an uncertain PWL system . In section 4, the output feedback controller design is investigated and it is explained how the upper bound on the cost function can be minimized. The method is tested on a numerical example in Section 5. Conclusions are given in Section 6.

2 Uncertain Piecewise linear systems

We consider an uncertain piecewise linear discrete time system of the following form:

$$\begin{aligned} x(t+1) &= (A_i + \Delta A_i)x(t) + (B_i + \Delta B_i)u(t) \quad for\ x \in \mathcal{X}_i \\ y(t) &= (C_i + \Delta C_i)x(t), \end{aligned} \tag{10.1}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control input, and $y(t) \in \mathbb{R}^p$ is the output. $\{\mathcal{X}_i\}_{i=1}^{s} \subseteq \mathbb{R}^n$ denotes a partition of the state into a number of polyhedral regions $\mathcal{X}_i, i \in \mathcal{I} = \{1, \cdots, s\}$. Each polyhedral region is represented by:

$$\mathcal{X}_i = \{x|\mathcal{H}_i x \leq h_i\}, \tag{10.2}$$

and $\Delta A_i, \Delta B_i, \Delta C_i$ are parameter uncertainties in the parameters of the subsystem i of the following form:

$$\begin{bmatrix}\Delta A_i, & \Delta B_i\end{bmatrix} = M_{1_i} H \begin{bmatrix}N_{A_i}, & N_{B_i}\end{bmatrix}, \tag{10.3}$$

$$\Delta C_i = M_{2_i} H N_{C_i}, \tag{10.4}$$

where H is an uncertain matrix bounded by:

$$HH^T \leq I. \tag{10.5}$$

and $M_{1_i}, M_{2_i}, N_{A_i}, N_{B_i}, N_{C_i}$ are known constant matrices of appropriate dimensions. All possible switchings from region $\mathcal{X}_i$ to $\mathcal{X}_j$ are represented by the set $\mathcal{S}$:

$$\mathcal{S} := \{(i, j)|x(t) \in \mathcal{X}_i, x(t+1) \in \mathcal{X}_j\} \tag{10.6}$$

3 State Feedback Design for uncertain PWL systems

3.1 Piecewise Quadratic Stability

The problem of piecewise linear state feedback design is to design a state feedback of the form:

$$u(t) = K_i x(t) \;\; for \; x(t) \in \mathcal{X}_i \tag{10.7}$$

such that the closed loop piecewise linear system

$$x(t+1) = \mathcal{A}_i x(t), \tag{10.8}$$

where $\mathcal{A}_i = A_i + \Delta A_i + (B_i + \Delta B_i)K_i$, is exponentially stable.

Theorem 10.1. *([4]) The system in (10.8) is exponentially stable if there exist matrices $P_i = P_i^T > 0$, $\forall i \in \mathcal{I}$, such that the positive definite function $V(x(t)) = x^T(t)P_i x(t)$, $\forall x \in \mathcal{X}_i$ satisfies $V(x(t+1)) - V(x(t)) < 0$.*

The piecewise quadratic Lyapunov function in Theorem 10.1 can be computed by solving the following matrix inequalities:

$$\mathcal{A}_i P_j \mathcal{A}_i - P_i < 0 \;\; \forall (i, j) \in S \tag{10.9}$$

$$P_i = P_i^T > 0 \;\; \forall i \in \mathcal{I} \tag{10.10}$$

3.2 PWL Quadratic Regulator (PWLQR)

The quadratic cost function associated with the system is:

$$J = \sum_{k=0}^{\infty} x^T(t) Q_i x(t) + u^T(t) R_i u(t), \tag{10.11}$$

where $Q_i \geq 0$ and $R_i \geq 0$ are given weighting matrices of appropriate dimensions.

Lemma 10.1. ***Upper bound on the performance cost:*** *The system in (10.1) with the controller in (10.7) satisfies the following upper bound on the performance cost*

$$J \leq x(0)^T P_{i_0} x(0) \tag{10.12}$$

with $x(0) \in \mathcal{X}_{i_0}$ and P_{i_0}, if there exist matrices $P_i = P_i^T > 0$, $\forall i \in \mathcal{I}$ such that

$$\begin{aligned}(A_i + \Delta A_i + (B_i + \Delta B_i)K_i)^T P_j (A_i + \Delta A_i + (B_i + \Delta B_i)K_i)\\ -P_i + Q_i + K_i^T R K_i < 0 \, \forall (i,j) \in S\end{aligned} \tag{10.13}$$

Proof. Pre and post-multiplying (10.13) by $x^T(t)$ and $x(t)$ we have:

$$\begin{aligned}x^T(t)(A_i + \Delta A_i + (B_i + \Delta B_i)K_i)^T P_j (A_i + \Delta A_i + (B_i + \Delta B_i)K_i)x(t) - \\ x^T(t)P_i x(t) + x^T(t)Q_i x(t) + x^T(t)K_i^T R_i K_i x(t) < 0\end{aligned} \tag{10.14}$$

which implies:

$$V(x(t+1)) - V(x(t)) + x^T(t)Q_i x(t) + u^T(t)R_i u(t) < 0 \tag{10.15}$$

Summing up the above equation from $k = 0$ to $k = \infty$ we have:

$$V(x(\infty)) - V(x(0)) + \Sigma_0^\infty (x^T(t)Q_i x(t) + u^T(t)R_i u(t)) < 0 \tag{10.16}$$

As $V(x(\infty)) = 0$ and $V(x(0)) = x(0)^T P_{i_0} x(0)$ therefore we have:

$$\sum_{k=0}^{\infty} (x^T(t)Q_i x(t) + u^T(t)R_i u(t)) < x^T(0)P_{i_0} x(0)$$

□

The inequality (10.13) is a nonlinear matrix inequality and difficult to solve. In the following, an LMI equivalent of it is presented.

Before presenting the main result of this paper in the next section, the following lemma is introduced.

Lemma 10.2. *([17]) Let M, N, H be real matrices. If $H^T H \leq I$, then for every scalar $\epsilon > 0$ the following inequality holds:*

$$MHN + N^T H^T M^T \leq \epsilon M M^T + \epsilon^{-1} N^T N \tag{10.17}$$

Definition 10.1. A piecewise linear control law of the form (10.7) is a guaranteed cost control for the uncertain system (10.1) and the performance function (10.11) if the following matrix inequality is satisfied:

$$\begin{aligned}(A_i + \Delta A_i + (B_j + \Delta B_i)K_j)^T P_j (A_i + \Delta A_i + (B_i + \Delta B_i)K_i)\\ -P_i + Q_i + K_i^T R_i K_i < 0, \quad \forall (i,j) \in \mathcal{S}.\end{aligned} \tag{10.18}$$

Then, the PWL system with the law obtained from solving (10.18) is quadratically stable and for the performance function satisfies:

$$J \leq x^T(0)P_{i_0} x(0).$$

Theorem 10.2. *There exists a state feedback control law for the uncertain PWL system (10.1) with the performance function (10.11) if there exist symmetric matrices $X_i = X_i^T > 0$ and matrices Y_i and positive scalars $\epsilon_i > 0, i \in \mathcal{I}$ such that the following LMI is satisfied:*

$$\begin{bmatrix} X_j + 2\epsilon_i M_{1_i} M_{1_i}^T & (A_i X_i + B_i Y_i) & 0 & 0 & 0 & 0 \\ (A_i X_i + B_i Y_i)^T & -X_i & Y_i^T & X_i & N_{A_i}^T & Y_i^T N_{B_i}^T \\ 0 & Y_i & -R^{-1} & 0 & 0 & 0 \\ 0 & X_i & 0 & -Q_i^{-1} & 0 & 0 \\ 0 & N_{A_i} & 0 & 0 & -\epsilon_{1_i} & 0 \\ 0 & N_{B_i} Y_i & 0 & 0 & 0 & -\epsilon_{1_i} \end{bmatrix} < 0$$

$$\forall (i,j) \in \mathcal{S}. \tag{10.19}$$

Then the piecewise linear feedback gains are given by:

$$K_i = Y_i X_i^{-1}, \tag{10.20}$$

with Y_i and X_i from solving (10.19) and the performance function of the closed loop system satisfies:

$$J \leq x^T(0) X_{i_0}^{-1} x(0). \tag{10.21}$$

Proof. Using Schur complement (10.18) is equivalent to:

$$\begin{bmatrix} -P_j^{-1} & (A_i + \Delta A_i + (B_i + \Delta B_i)K_i) & 0 \\ (A_i + \Delta A_i + (B_i + \Delta B_i)K_i)^T & -P_i + Q_i & K_i^T \\ 0 & K_i & -R_i^{-1} \end{bmatrix} < 0, \tag{10.22}$$

which implies:

$$\phi + \psi < 0, \tag{10.23}$$

where

$$\phi = \begin{bmatrix} -P_j^{-1} & (A_i + B_i K_i) & 0 \\ (A_i + B_i K_i)^T & -P_i + Q_i & K_i^T \\ 0 & K_i & -R^{-1} \end{bmatrix}, \tag{10.24}$$

$$\psi = \begin{bmatrix} 0 & (\Delta A_i + \Delta B_i K_i) & 0 \\ (\Delta A_i + \Delta B_i K_i)^T & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{10.25}$$

From (10.3), (10.4), we have:

$$\psi = R_1 + R_2, \tag{10.26}$$

with

$$R_1 = \begin{bmatrix} M_{1_i}^T & 0 & 0 \end{bmatrix}^T H \begin{bmatrix} 0 & N_{A_i} & 0 \end{bmatrix} + \begin{bmatrix} 0 & N_{A_i} & 0 \end{bmatrix}^T H^T \begin{bmatrix} M_{1_i}^T & 0 & 0 \end{bmatrix}, \tag{10.27}$$

and

$$R_2 = \begin{bmatrix} M_{1_i}^T & 0 & 0 \end{bmatrix}^T H \begin{bmatrix} 0 & N_{B_i} K_i & 0 \end{bmatrix} + \begin{bmatrix} 0 & N_{B_i} K_i & 0 \end{bmatrix}^T H^T \begin{bmatrix} M_{1_i}^T & 0 & 0 \end{bmatrix}. \tag{10.28}$$

Using Lemma 10.2, for any ϵ_i we have:

$$R_1 \leq \epsilon_i \begin{bmatrix} M_{1_i} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} M_{1_i}^T & 0 & 0 \end{bmatrix} + \epsilon_i^{-1} \begin{bmatrix} 0 \\ N_{A_i}^T \\ 0 \end{bmatrix} \begin{bmatrix} 0 & N_{A_i} & 0 \end{bmatrix} \tag{10.29}$$

and

$$R_2 \leq \epsilon_i \begin{bmatrix} M_{1_i} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} M_{1_i}^T & 0 & 0 \end{bmatrix} + \epsilon_{1_i}^{-1} \begin{bmatrix} 0 \\ K_i^T N_{B_i}^T \\ 0 \end{bmatrix} \begin{bmatrix} 0 & N_{B_i} K_i & 0 \end{bmatrix}. \tag{10.30}$$

Therefore, we have:

$$\phi + \psi \leq \begin{bmatrix} -P_j^{-1} + 2\epsilon_i M_{1_i} M_{1_i}^T & (A_i + B_i K_i) & 0 \\ (A_i + B_i K_i)^T & -P_i + Q_i + \epsilon_i^{-1} N_{A_i}^T N_{A_i} + \epsilon_i^{-1} K_i^T N_{B_i}^T N_{B_i} & K_i^T \\ 0 & K_i & R_i^{-1} \end{bmatrix} \tag{10.31}$$

We pre- and post-multiply the right hand side of the above inequality by diag$\{I, P_i^{-1}, I\}$ and its transpose. Then we get:

$$\begin{bmatrix} -P_j^{-1} + 2\epsilon_j M_i M_{1_i}^T & (A_i + B_i K_i) P_i^{-1} & 0 \\ P_i^{-1} (A_i + B_i K_i)^T & X_{22} & P_i^{-1} K_i^T \\ 0 & K_i P_i^{-1} & R_i^{-1} \end{bmatrix}, \tag{10.32}$$

where

$$X_{22} = -P_i^{-1} + P_i^{-1} Q_i P_i^{-1} + \epsilon_i^{-1} P_i^{-1} N_{A_i}^T N_{A_i} P_i^{-1} + \epsilon_i^{-1} P_i^{-1} K_i^T N_{B_i}^T N_{B_i} K_i P_i^{-1} \tag{10.33}$$

Define $X_i = P_i^{-1}$ and $Y_i = K_i P_i^{-1}$. Applying Schur complement to the above equation we get the LMI condition (10.19) as a sufficient condition for (10.18). □

The upper bound in (10.11), could be minimized by the following constrained optimization problem:

$$\min_{X_i, Y_i, \epsilon_i, \rho} \quad \rho \tag{10.34}$$

$$s.t. \begin{cases} (19) \\ \begin{bmatrix} -\rho & x(0)^T \\ x(0) & -X_{i_0} \end{bmatrix} < 0 \end{cases}$$

Using Schur complement, the new LMI constraint is equivalent to $-\rho + x^T(0) X_{i_0}^{-1} x(0) < 0$. Therefore, (10.34) is equal to $\min \; x^T(0) P_{i_0} x(0)$ with the remaining constraints. The problem with the above formulation is that the upper bound is dependent on the initial state $x(0)$. To remove this dependency, the initial condition is considered as a random variable with uniform distribution in a bounded region $\overline{\mathcal{X}}$. Then, it is tried to minimize the expected value of the cost function. We have:

$$E(J) \leq E(tr(P_{i_0} x(0) x^T(0))) \leq \sum_{i \in \mathcal{I}} \sigma_i tr(P_i L_i), \tag{10.35}$$

where $L_i = E(x(0)x^T(0))$ is the expectation of $x(0)x^T(0)$ corresponding to $x(0) \in \mathcal{X}_i, i \in \mathcal{I}$, $tr(\cdot)$ is the trace operator and σ_i is the probability of $x(0) \in \mathcal{X}_i$. Then, the optimization problem is:

$$\min_{X_i,Y_i} \sum_{i\in\mathcal{I}} \sigma_i tr(X_i^{-1}L_i) \tag{10.36}$$
$$s.t. \begin{cases} (19) \\ X_i = X_i^T > 0 \end{cases}$$

The optimization problems in (10.36) is non-convex because it includes X_i and its inverse. To convert it to a convex optimization problem , we introduce new variables $V_i, i \in \mathcal{I}$, which satisfies:

$$\begin{bmatrix} V_i & I \\ I & X_i \end{bmatrix} \geq 0. \tag{10.37}$$

Using Schur complement, the above constraint is equivalent to $X_i^{-1} \leq V_i$. Therefore, the objective function in (10.36), which is nonlinear in term of X_i, can be converted to $\sum_{i\in\mathcal{I}} \sigma_i tr(V_i L_i)$. Consequently, the optimization problem (10.36) can be transformed to the following optimization problem in terms of variables X_i, Y_i, ϵ_i and the introduced variables V_i:

$$\min_{X_i,Y_i,V_i,\epsilon_i} \sum_{i\in\mathcal{I}} \sigma_i tr(V_i L_i) \tag{10.38}$$
$$s.t. \begin{cases} (22) \\ \begin{bmatrix} V_i & I \\ I & X_i \end{bmatrix} \geq 0, i \in \mathcal{I} \\ X_i = X_i^T > 0,\ i \in \mathcal{I} \\ V_i = V_i^T > 0,\ i \in \mathcal{I} \end{cases}$$

The above optimization problems is a convex optimization problems with LMI constraints and can be solved efficiently using available softwares like YALMIP/SeDuMi or YALMIP/LMILAB, see [18].

4 Output Feedback Control

In this section, we consider dynamic output feedback control for uncertain piecewise affine systems of the following form:

$$\begin{aligned} x_c(t+1) &= A_{c_i}x_c(t) + B_{c_i}y(t) \quad for\ x_c \in \mathcal{X}_i \\ u(t) &= C_{c_i}x_c(t) + D_{c_i}y(t) \end{aligned} \tag{10.39}$$

We define the set of all possible transitions for the controller state from region $\mathcal{X}_k$ to $\mathcal{X}_l$ as:

$$\hat{\mathcal{S}} := \{(t,l)|x_c(t) \in \mathcal{X}_k, x_c(t+1) \in \mathcal{X}_l\} \tag{10.40}$$

Considering the general case that the system is in mode i, $(x(t) \in \mathcal{X}_i)$, and the controller is in the mode j, $(x_c(t) \in \mathcal{X}_j)$, the dynamic of the closed loop system is:

$$x(t+1) = (A_i + \Delta A_i)x(t) + (B_i + \Delta B_i)\left[C_{c_j}x_c(t) + D_{c_j}(C_i + \Delta C_i)x(t)\right] \tag{10.41}$$

and the dynamic of the augmented system is:

$$\begin{bmatrix} x(t+1) \\ x_c(t+1) \end{bmatrix} = \begin{bmatrix} A_i + \Delta A_i + (B_i + \Delta B_i) D_{c_j} (C_i + \Delta C_i) & (B_i + \Delta B_i) C_{c_j} \\ B_{c_j}(C_i + \Delta C_i) & A_{c_j} \end{bmatrix} \begin{bmatrix} x(t) \\ x_c(t) \end{bmatrix} \tag{10.42}$$

The quadratic cost function associated with the system is:

$$J = \sum_{t=0}^{\infty} x^T(t) Q_i x(t) + u^T(t) R_i u(t), \tag{10.43}$$

where $Q_i \geq 0$ and $R_i \geq 0$ are given weighting matrices of appropriate dimensions. We define the augmented state as $\tilde{x} = \begin{bmatrix} x^T(t) & x_c^T(t) \end{bmatrix}^T$ and the following notations for convenience:

$$\tilde{A}_i = \begin{bmatrix} A_i & 0 \\ 0 & 0 \end{bmatrix}, \; \tilde{B}_i = \begin{bmatrix} B_i & 0 \\ 0 & I \end{bmatrix}, \; \tilde{C}_i = \begin{bmatrix} C_i & 0 \\ 0 & I \end{bmatrix} \tag{10.44}$$

$$\tilde{M}_{1_i} = \begin{bmatrix} M_{1_i} \\ 0 \end{bmatrix}, \; \tilde{M}_{2_i} = \begin{bmatrix} M_{2_i} \\ 0 \end{bmatrix}, \tag{10.45}$$

$$\tilde{N}_{A_i} = \begin{bmatrix} N_{A_i} & 0 \end{bmatrix}, \; \tilde{N}_{B_i} = \begin{bmatrix} N_{B_i} & 0 \end{bmatrix}, \; \tilde{N}_{C_i} = \begin{bmatrix} N_{C_i} & 0 \end{bmatrix}, \tag{10.46}$$

$$K_i = \begin{bmatrix} D_{c_i} & C_{c_i} \\ B_{c_i} & A_{c_i} \end{bmatrix}, \; \tilde{Q}_i = \begin{bmatrix} Q_i & 0 \\ 0 & 0 \end{bmatrix}, \; \tilde{R}_i = \begin{bmatrix} R_i & 0 \\ 0 & 0 \end{bmatrix}. \tag{10.47}$$

Using the above notations, the uncertainty matrices are rewritten as:

$$\begin{bmatrix} \Delta\tilde{A}_i & \Delta\tilde{B}_i \end{bmatrix} = \tilde{M}_{1_i} H \begin{bmatrix} \tilde{N}_{A_i} & \tilde{N}_{B_i} \end{bmatrix}, \; \Delta\tilde{C}_i = \tilde{M}_{2_i} H \tilde{N}_{C_i} \tag{10.48}$$

Then, the dynamic of the augmented system in terms of new variables is:

$$\tilde{x}(t+1) = \left[\tilde{A}_i + \Delta\tilde{A}_i + (\tilde{B}_i + \Delta\tilde{B}_i)(K_j)(\tilde{C}_i + \Delta\tilde{C}_i) \right] \tilde{x}(t) \tag{10.49}$$

and the associated cost function can be rewritten as:

$$J = \sum_{t=0}^{\infty} \tilde{x}^T(t) \left[\tilde{Q}_i + (\tilde{C}_i + \Delta\tilde{C}_i)^T K_j \tilde{R}_i K_j (\tilde{C}_i + \Delta\tilde{C}_i) \right] \tilde{x}(t) \tag{10.50}$$

The following theorem gives us a sufficient condition in terms of BMIs to ensure closed loop stability of the uncertain PWL system and provides us with an upper bound on the performance cost.

Theorem 10.3. *Consider the uncertain piecewise linear system in (10.1) with the dynamic output feedback controller in (10.43), if for given $\epsilon_{3_i}, \epsilon_{4_i} > 0$ there exist positive constants $\epsilon_{1_i}, \epsilon_{2_i}, \epsilon_{5_i}$ and symmetric matrices $X_{ik} = X_{ik}^T > 0$, such that the following BMIs are satisfied:*

$$
\begin{bmatrix}
\Xi & \tilde{A}_{cl_{ik}}X_{ik} & \tilde{B}_iK_k\tilde{M}_{2_i} & 0 & 0 & 0 & 0 & 0 & 0 \\
* & -X_{ik} & 0 & X_{ik}(\tilde{N}_{B_i}K_k\tilde{C}_i)^T & X_{ik}(\tilde{R}_i^{1/2}K_k\tilde{C}_i)^T & X_{ik}\tilde{N}_{A_i} & X_{ik}\tilde{N}_{C_i} & 0 & X_{ik}\tilde{Q}_i^{1/2} \\
* & 0 & -\epsilon_{3_i}I & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & * & 0 & -\epsilon_{2_i}I & 0 & 0 & 0 & 0 & 0 \\
0 & * & 0 & 0 & -I & 0 & 0 & \tilde{R}_i^{1/2}\tilde{M}_{2_i} & 0 \\
0 & * & 0 & 0 & 0 & -\epsilon_{1_i}I & 0 & 0 & 0 \\
0 & * & 0 & 0 & 0 & 0 & -1/3\epsilon_{3_i}^{-1}I & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\epsilon_{3_i}I & 0 \\
0 & * & 0 & 0 & 0 & 0 & 0 & 0 & -I
\end{bmatrix} < 0, \tag{10.51}
$$

$$
\begin{bmatrix}
-\epsilon_{4_i}^{-1}I & 0 & 0 & \tilde{N}_{B_i}^T \\
0 & -\epsilon_{4_i}I & K_k\tilde{M}_{2_i} & 0 \\
0 & * & -\epsilon_{3_i} & 0 \\
* & 0 & 0 & -\epsilon_{1_i}I
\end{bmatrix} < 0, \tag{10.52}
$$

for all $(i,k) \in \mathcal{S}$ *and* $(j,l) \in \hat{\mathcal{S}}$*, where:*

$$\Xi = -X_{jl} + 2\epsilon_{1_i}\tilde{M}_{1_i}\tilde{M}_{1_i}^T + \epsilon_{2_i}\tilde{M}_{2_i}\tilde{M}_{2_i}^T, \tag{10.53}$$

$$\tilde{A}_{cl_{ik}} = \tilde{A}_i + \tilde{B}_iK_k\tilde{C}_i, \tag{10.54}$$

then the closed loop system is globally exponentially stable, and the upper bound on the performance function satisfies:

$$E(J) \leq \sum_{i,k\in\mathcal{I}\times\mathcal{I}} \sigma_{ik} tr(X_{ik}^{-1}L_{ik}). \tag{10.55}$$

Proof. We consider a piecewise quadratic Lyapunov function candidate as:

$$V(\tilde{x}(t)) = \tilde{x}^T(t)P_{ij}\tilde{x}(t). \tag{10.56}$$

with $P_{ij} = P_{ij}^T > 0$.

For the closed loop augmented system to be stable and the cost function (10.43) to be guaranteed, the following inequality must hold:

$$V(\tilde{x}(t+1)) - V(\tilde{x}(t)) + \tilde{x}^T(t)[\tilde{Q}_i + (\tilde{C}_i + \Delta\tilde{C}_i)^T K_k^T \tilde{R}_i K_k (\tilde{C}_i + \Delta\tilde{C}_i)]\tilde{x}(t) < 0 \tag{10.57}$$

Therefore for all for all $(i,j) \in \mathcal{S}$ and $(k,l) \in \hat{\mathcal{S}}$ we must have:

$$\tilde{x}^T(t+1)P_{jl}\tilde{x}(t+1) - \tilde{x}^T(t)P_{ik}\tilde{x}(t)$$
$$+\tilde{x}^T(t)[\tilde{Q}_i + (\tilde{C}_i + \Delta\tilde{C}_i)^T K_k^T R_i K_k (\tilde{C}_i + \Delta\tilde{C}_i)]\tilde{x}(t) < 0. \tag{10.58}$$

In the above equation, it is assumed that at time t the system is in the mode i and the controller is in the mode k and at time $t+1$, the system switches to the mode j and the controller to the mode l. Then, it implies:

$$\begin{aligned}&[\tilde{\mathcal{A}}_{cl_{ik}}]^T P_{jl} [\tilde{\mathcal{A}}_{cl_{ik}}] - P_{ik} + \\ &[\tilde{Q}_i + (\tilde{C}_i + \Delta\tilde{C}_i)^T K_k^T R_i K_k (\tilde{C}_i + \Delta\tilde{C}_i)] < 0,\end{aligned} \tag{10.59}$$

where

$$\tilde{\mathcal{A}}_{cl_{ik}} = \tilde{A}_i + \Delta\tilde{A}_i + (\tilde{B}_i + \Delta\tilde{B}_i)(K_k)(\tilde{C}_i + \Delta\tilde{C}_i) \tag{10.60}$$

Equation (10.59)which can be rewritten as:

$$\begin{bmatrix} -P_{jl}^{-1} & \tilde{\mathcal{A}}_{cl_{ik}} \\ \tilde{\mathcal{A}}_{cl_{ik}}^T & -P_{ik} + \tilde{Q}_i + (\tilde{C}_i + \Delta\tilde{C}_i)^T K_k \tilde{R}_i K_k (\tilde{C}_i + \Delta\tilde{C}_i) \end{bmatrix} < 0, \tag{10.61}$$

The above equation can be written as:

$$\Phi_0 + \Phi_1 \leq 0, \tag{10.62}$$

where

$$\Phi_0 = \begin{bmatrix} -P_{jl}^{-1} & (\tilde{A}_i + \tilde{B}_i K_k \tilde{C}_i) \\ (\tilde{A}_i + \tilde{B}_i K_k \tilde{C}_i)^T & -P_{ik} + \tilde{Q}_i + (\tilde{C}_i + \Delta\tilde{C}_i)^T K_k^T R_i K_k (\tilde{C}_i + \Delta\tilde{C}_i) \end{bmatrix} \tag{10.63}$$

and

$$\Phi_1 = \begin{bmatrix} 0 & \Delta\tilde{A}_i + \tilde{B}_i K_k \Delta\tilde{C}_i + \Delta\tilde{B}_i K_k \tilde{C}_i + \Delta\tilde{B}_i K_k \Delta\tilde{C}_i \\ * & 0 \end{bmatrix}. \tag{10.64}$$

Φ_1 can be rewritten as:

$$\Phi_1 = \psi_1 + \psi_2 + \psi_3 + \psi_4, \tag{10.65}$$

with

$$\psi_1 = \begin{bmatrix} \tilde{M}_{1_i} \\ 0 \end{bmatrix} H \begin{bmatrix} 0 & \tilde{N}_{A_i} \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{N}_{A_i}^T \end{bmatrix} H \begin{bmatrix} \tilde{M}_{1_i}^T & 0 \end{bmatrix}, \tag{10.66}$$

$$\psi_2 = \begin{bmatrix} \tilde{M}_{1_i} \\ 0 \end{bmatrix} H \begin{bmatrix} 0 & \tilde{N}_{B_i} K_k \tilde{C}_i \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{C}_i^T K_k^T \tilde{N}_{B_i}^T \end{bmatrix} H \begin{bmatrix} \tilde{M}_{1_i}^T & 0 \end{bmatrix}, \tag{10.67}$$

$$\psi_3 = \begin{bmatrix} \tilde{B}_i K_k \tilde{M}_{2_i} \\ 0 \end{bmatrix} H \begin{bmatrix} 0 & \tilde{N}_{C_i} \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{N}_{C_i}^T \end{bmatrix} H \begin{bmatrix} \tilde{M}_{2_i}^T K_k^T \tilde{B}_i^T \end{bmatrix}, \tag{10.68}$$

$$\psi_4 = \begin{bmatrix} \Delta\tilde{B}_i \\ 0 \end{bmatrix} \begin{bmatrix} 0 & K_k \Delta\tilde{C}_k \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta\tilde{C}_i^T K_k^T \end{bmatrix} \begin{bmatrix} \Delta\tilde{B}_i^T & 0 \end{bmatrix}. \tag{10.69}$$

Using Lemma 10.2 we have:

$$\psi_1 \leq \epsilon_{1_i} \begin{bmatrix} \tilde{M}_{1_i} \\ 0 \end{bmatrix} \begin{bmatrix} \tilde{M}_{1_i}^T & 0 \end{bmatrix} + \epsilon_{1_i}^{-1} \begin{bmatrix} 0 \\ \tilde{N}_{A_i}^T \end{bmatrix} \begin{bmatrix} 0 & \tilde{N}_{A_i} \end{bmatrix}, \tag{10.70}$$

$$\psi_2 \leq \epsilon_{2_i} \begin{bmatrix} \tilde{M}_{2_i} \\ 0 \end{bmatrix} \begin{bmatrix} \tilde{M}_{2_i}^T & 0 \end{bmatrix} + \epsilon_{2_i}^{-1} \begin{bmatrix} 0 \\ \tilde{C}_i^T K_k^T \tilde{N}_{B_i}^T \end{bmatrix} \begin{bmatrix} 0 & \tilde{N}_{B_i} K_k \tilde{C}_i \end{bmatrix}, \tag{10.71}$$

$$\psi_3 \leq \epsilon_{3_i}^{-1} \begin{bmatrix} \tilde{B}_i K_k \tilde{M}_{2_i} \\ 0 \end{bmatrix} \begin{bmatrix} \tilde{M}_{2_i}^T K_k^T \tilde{B}_i^T & 0 \end{bmatrix} + \epsilon_{3_i} \begin{bmatrix} 0 \\ \tilde{N}_{C_i}^T \end{bmatrix} \begin{bmatrix} \tilde{N}_{C_i} 0 \end{bmatrix}, \tag{10.72}$$

$$\psi_4 \leq \epsilon_{4_i} \begin{bmatrix} \Delta\tilde{B}_i \\ 0 \end{bmatrix} \begin{bmatrix} \Delta\tilde{B}_i^T \end{bmatrix} + \epsilon_{4_i}^{-1} \begin{bmatrix} 0 \\ \Delta\tilde{C}_j^T K_k^T \end{bmatrix} \begin{bmatrix} K_k \Delta\tilde{C}_i \end{bmatrix}. \tag{10.73}$$

Therefore:

$$\Phi_0 + \Phi_1 \leq \begin{bmatrix} -P_{jl}^{-1} + \xi_1 & (\tilde{A}_i + \tilde{B}_i K_k \tilde{C}_i) \\ * & -P_{ik} + \tilde{Q}_i + (\tilde{C}_i + \Delta\tilde{C}_i)^T K_k^T \tilde{R}_i K_k (\tilde{C}_i + \Delta\tilde{C}_i) + \xi_2 \end{bmatrix}, \tag{10.74}$$

where

$$\xi_1 = \epsilon_{1_i} \tilde{M}_{1_i} \tilde{M}_{1_i}^T + \epsilon_{2_i} \tilde{M}_{2_i} \tilde{M}_{2_i}^T + \epsilon_{3_i}^{-1} \tilde{B}_i K_k \tilde{M}_{2_i} \tilde{M}_{2_i}^T \tilde{B}_i^T K_k^T + \epsilon_{4_i} \Delta\tilde{B}_i \Delta\tilde{B}_i^T, \tag{10.75}$$

$$\xi_2 = \epsilon_{1_i}^{-1} \tilde{N}_{A_i}^T \tilde{N}_{A_i} + \epsilon_{2_i}^{-1} \tilde{C}_i^T K_k^T \tilde{N}_{B_i}^T \tilde{N}_{B_i} K_k \tilde{C}_i + \epsilon_{3_i} \tilde{N}_{C_i}^T \tilde{N}_{C_i} + \epsilon_{4_i}^{-1} \Delta\tilde{C}_i^T K_k^T K_k \Delta\tilde{C}_i. \tag{10.76}$$

Applying Schur complement, see [19], we have:

$$\Phi_0 + \Phi_1 \leq \Omega =$$
$$\begin{bmatrix} \Omega_{11} & \tilde{A}_{cl_{ik}} & \tilde{B}_i K_k \tilde{M}_{2_i} & 0 & \Delta\tilde{B}_i & 0 & 0 & 0 & 0 \\ * & -P_{ik} + \tilde{Q}_i & 0 & (\tilde{N}_{B_i} K_k \tilde{C}_i)^T & 0 & (K_i \Delta\tilde{C}_i)^T & \Omega_{26} & \tilde{N}_{A_i} & \tilde{N}_{C_i} \\ * & 0 & -\epsilon_{3_i} I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\epsilon_{2_i} I & 0 & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & -\epsilon_{4_i}^{-1} I & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & -\epsilon_{4_i} I & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & 0 & -I & 0 & 0 \\ 0 & * & 0 & 0 & 0 & 0 & 0 & -\epsilon_{1_i} I & 0 \\ 0 & * & 0 & 0 & 0 & 0 & 0 & 0 & -\epsilon_{3_i}^{-1} I \end{bmatrix} \tag{10.77}$$

with

$$\Omega_{11} = -P_{jl}^{-1} + \epsilon_{1_i} \tilde{M}_{1_i} \tilde{M}_{1_i}^T + \epsilon_{2_i} \tilde{M}_{2_i} \tilde{M}_{2_i}^T, \tag{10.78}$$

$$\Omega_{26} = (\tilde{R}_i^{1/2} K_k (\tilde{C}_i + \Delta\tilde{C}_i))^T. \tag{10.79}$$

This can be written as:

$$\Omega = \Omega_0 + L_1 + L_2 + L_3, \tag{10.80}$$

where

$$L_1 = \begin{bmatrix} \tilde{M}_{1_i} \\ 0_{8\times1} \end{bmatrix} H \begin{bmatrix} 0_{1\times6} & \tilde{N}_{B_i} & 0_{1\times2} \end{bmatrix} + \begin{bmatrix} 0_{6\times1} \\ \tilde{N}_{B_i}^T \\ 0_{2\times1} \end{bmatrix} H \begin{bmatrix} \tilde{M}_{1_i}^T & 0_{1\times8} \end{bmatrix}, \tag{10.81}$$

$$L_2 = \begin{bmatrix} 0 \\ \tilde{N}_{C_i}^T \\ 0_{7\times1} \end{bmatrix} H \begin{bmatrix} 0_{1\times7} & \tilde{M}_{2_i}^T K_k^T & 0 \end{bmatrix} + \begin{bmatrix} 0_{7\times1} \\ K_k \tilde{M}_{2_i} \\ 0 \end{bmatrix} \begin{bmatrix} 0 & \tilde{N}_{C_i} & 0_{7\times1} \end{bmatrix}, \tag{10.82}$$

$$L_3 = \begin{bmatrix} 0 \\ \tilde{N}_{C_i}^T \\ 0_{7\times1} \end{bmatrix} H \begin{bmatrix} 0_{1\times8} & \tilde{M}_{2_i}^T K_k^T \tilde{R}_i^{1/2} \end{bmatrix} + \begin{bmatrix} 0_{8\times1} \\ \tilde{R}_i^{1/2} K_k \tilde{M}_{2_i} \\ 0 \end{bmatrix} \begin{bmatrix} 0 & \tilde{N}_{C_i} & 0_{7\times1} \end{bmatrix}. \tag{10.83}$$

Applying Lemma 10.2, we have:

$$L_1 \leq \epsilon_{1_i} \begin{bmatrix} \tilde{M}_{1_i} \\ 0_{8\times1} \end{bmatrix} \begin{bmatrix} \tilde{M}_{1_i}^T & 0_{1\times8} \end{bmatrix} + \epsilon_{1_i}^{-1} \begin{bmatrix} 0_{6\times1} \\ \tilde{N}_{B_i}^T \\ 0_{2\times1} \end{bmatrix} \begin{bmatrix} 0_{1\times6} & \tilde{N}_{B_i} & 0_{1\times2} \end{bmatrix}, \tag{10.84}$$

$$L_2 \leq \epsilon_{3_i} \begin{bmatrix} 0 \\ \tilde{N}_{C_i}^T \\ 0_{7\times1} \end{bmatrix} \begin{bmatrix} 0 & \tilde{N}_{C_i} & 0_{1\times7} \end{bmatrix} + \epsilon_{3_i}^{-1} \begin{bmatrix} 0_{7\times1} \\ K_k \tilde{M}_{2_i} \\ 0 \end{bmatrix} \begin{bmatrix} 0_{1\times7} & \tilde{M}_{2_i}^T K_k^T & 0 \end{bmatrix}, \tag{10.85}$$

$$L_3 \leq \epsilon_{3_i} \begin{bmatrix} 0 \\ \tilde{N}_{C_i}^T \\ 0_{7\times1} \end{bmatrix} \begin{bmatrix} 0 & \tilde{N}_{C_i} & 0_{7\times1} \end{bmatrix} + \epsilon_{3_i}^{-1} \begin{bmatrix} 0_{8\times1} \\ \tilde{R}_i^{1/2} K_k \tilde{M}_{2_i} \end{bmatrix} \begin{bmatrix} 0_{1\times8} & \tilde{M}_{2_i}^T K_k^T \tilde{R}_i^{1/2} \end{bmatrix}, \tag{10.86}$$

which implies:

$$\Omega \leq$$
$$\begin{bmatrix} \Xi & \tilde{A}_{cl_{ik}} & \tilde{B}_i K_k \tilde{M}_{2_i} & 0 & 0 & 0 & 0 & 0 & 0 \\ * & -P_{ik} + \tilde{Q}_i & 0 & (\tilde{N}_{B_i} K_k \tilde{C}_i)^T & 0 & 0 & (\tilde{R}_i^{\frac{1}{2}} K_k \tilde{C}_i)^T & \tilde{N}_{A_i} & \tilde{N}_{C_i} \\ * & 0 & -\epsilon_{3_i} I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & -\epsilon_{2_i} I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma_2 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & 0 & \gamma_3 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & 0 & 0 & -\epsilon_{1_i} I & 0 \\ 0 & * & 0 & 0 & 0 & 0 & 0 & 0 & -1/3\epsilon_{3_i}^{-1} I \end{bmatrix}, \tag{10.87}$$

with

$$\gamma_1 = -\epsilon_{4_i}^{-1} I + \epsilon_{1_i}^{-1} \tilde{N}_{B_i} \tilde{N}_{B_i}^T, \tag{10.88}$$

$$\gamma_2 = -\epsilon_{4_i} I + \epsilon_{3_i}^{-1} K_k \tilde{M}_{2_i} \tilde{M}_{2_i}^T K_k^T, \tag{10.89}$$

$$\gamma_3 = -I + \epsilon_{3_i}^{-1} \tilde{R}_i^{\frac{1}{2}} K_k \tilde{M}_{2_i} \tilde{M}_{2_i}^T K_k^T \tilde{R}_i^{1/2}. \tag{10.90}$$

Using Schur complement we can see the following inequalities are sufficient conditions for the above inequality:

$$\begin{bmatrix} \Xi & \tilde{A}_{cl_{ik}} & \tilde{B}_i K_k \tilde{M}_{2_i} & 0 & 0 & 0 & 0 & 0 \\ * & -P_{ik} + \tilde{Q}_i & 0 & (\tilde{N}_{B_i} K_k \tilde{C}_i)^T & (\tilde{R}_i^{\frac{1}{2}} K_k \tilde{C}_i)^T & \tilde{N}_{A_i} & \tilde{N}_{C_i} & 0 \\ * & 0 & -\epsilon_{3_i} I & 0 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & -\epsilon_{2_i} I & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & -I & 0 & 0 & \tilde{R}_i^{\frac{1}{2}} \tilde{M}_{2_i} \\ 0 & * & 0 & 0 & 0 & -\epsilon_{1_i} I & 0 & 0 \\ 0 & * & 0 & 0 & 0 & 0 & -\frac{1}{3}\epsilon_{3_i}^{-1} I & 0 \\ 0 & 0 & 0 & 0 & * & 0 & 0 & -\epsilon_{3_i} I \end{bmatrix} < 0 \tag{10.91}$$

$$\begin{bmatrix} -\epsilon_{4_i}^{-1} I & 0 & 0 & \tilde{N}_{B_i}^T \\ 0 & -\epsilon_{4_i} I & K_k \tilde{M}_{2_i} & 0 \\ 0 & * & -\epsilon_{3_i} & 0 \\ * & 0 & 0 & -\epsilon_{1_i} I \end{bmatrix} < 0. \tag{10.92}$$

We pre- and post-multiply the right hand side of (10.91) by diag$\{I, P_{ik}^{-1}, I, I, I, I, I, I, I\}$ and its transpose. Using Schur complement again and defining $X_{ik} = P_{ik}^{-1}$, we conclude BMIs (10.51) and (10.52) as sufficient conditions for (10.57). □

To optimize the upper bound on the performance function we use a similar approach as before. The upper bound can be minimized using the following minimization problem with BMI constraints:

$$\min_{X_{ik}, V_{ik}, K_k, \epsilon_{1_i}, \epsilon_{2_i}} \sum_{i,k \in \mathcal{I} \times \mathcal{I}} \sigma_{ik} tr(V_{ik} L_{ik}) \tag{10.93}$$

$$s.t. \begin{cases} (48) \\ (49) \\ \begin{bmatrix} V_{ik} & I \\ I & X_{ik} \end{bmatrix} \geq 0 \\ X_{ik} = X_{ik}^T > 0 \\ V_{ik} = V_{ik}^T > 0 \end{cases}$$

This is a non-convex optimization problem with BMI constraints. To solve that we use the V-K iteration approach by [16] which provides us with a suboptimal solution. The V-K iteration approach consists of iterations over two steps: The V and the K step. The parameters of the BMI are divided into two sets, namely $\mathcal{V}$ and $\mathcal{K}$, such that assuming each set to be constants, the BMI is converted to an LMI. In the V step, the parameters in $\mathcal{V}$ are fixed and an optimization problem with LMI constraints is solved. The optimization results are the parameters in $\mathcal{K}$. In the K step, the parameters in $\mathcal{K}$ are fixed and unknown parameters in $\mathcal{V}$ are found. The algorithm is iterated until a stopping criterion, e.g no improvement in the cost or no change in variables, is met.

The V-K iteration to solve the problem (10.93) can now be summarized as :

- V-Step: Fix K_k, solve:

$$\min_{X_{ik},V_{ik},\epsilon_{1_i},\epsilon_{2_i}} \sum_{i,k\in\mathcal{I}\times\mathcal{I}} \sigma_{ik} tr(V_{ik}L_{ik}) \tag{10.94}$$
$$s.t. \begin{cases} (48) \\ (49) \\ \begin{bmatrix} V_{ik} & I \\ I & X_{ik} \end{bmatrix} \geq 0 \\ X_{ik} = X_{ik}^T > 0 \\ V_{ik} = V_{ik}^T > 0 \end{cases}$$

- K-step: Fix X_{ik}. Solve:

$$\min_{V_{ik},K_i,\epsilon_{1_i},\epsilon_{2_i}} \sum_{i,k\in\mathcal{I}\times\mathcal{I}} \sigma_{ik} tr(V_{ik}L_{ik}) \tag{10.95}$$
$$s.t. \begin{cases} (48) \\ (49) \\ \begin{bmatrix} V_{ik} & I \\ I & X_{ik} \end{bmatrix} \geq 0 \\ V_{ik} = V_{ik}^T > 0 \end{cases}$$

The V-K algorithm provides us with a suboptimal solution because it searches only in restricted directions.

5 Example

We consider the following unstable uncertain PWL system:

$$A_1 = \begin{bmatrix} 0.2834 & 0.7041 & 0.1071 \\ 0.1201 & 0.6523 & 0.6348 \\ 0.5288 & 0.9332 & 0.0955 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0.4376 & 0.9832 & 0.6901 \\ 0.1151 & 0.7553 & 0.3026 \\ 0.7585 & 0.9580 & 0.4180 \end{bmatrix}$$
$$A_3 = \begin{bmatrix} 0.8174 & 0.4479 & 0.5661 \\ 0.5195 & 0.4814 & 0.2914 \\ 0.2078 & 0.1988 & 0.6211 \end{bmatrix}$$
$$B_1 = \begin{bmatrix} 0.7258 \\ 0.1044 \\ 0.4182 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0.3151 \\ 0.7683 \\ 0.9588 \end{bmatrix} \quad B_3 = \begin{bmatrix} 0.3093 \\ 0.0365 \\ 0.8125 \end{bmatrix}$$
$$C_1 = C_2 = C_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

and parameter uncertainties of:

$$M_{1_i} = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}, \; M_{2_i} = \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 \end{bmatrix}$$

$$N_{A_i} = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 0.4 \end{bmatrix}, \; N_{B_i} = \begin{bmatrix} 0.8 \\ 0.8 \\ 0.8 \end{bmatrix}, \; N_{C_1} = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 0.4 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The state partition of the system is given by $\mathcal{X}_1 = \{x|x_1 < 0\}, \mathcal{X}_2 = \{x|x_1 \geq 0, x_2 \geq 0\}$, and $\mathcal{X}_2 = \{x_1 \geq 0, x_2 < 0\}$. The weighting matrices are $Q_i = 0.01I, R_i = 0.01$. Then, the controller gains are given by:

$$K_1 = \begin{bmatrix} -0.4715 & -0.4715 & 0.0741 & -0.3386 & -0.1112 \\ 0.1016 & 0.1016 & 0.1195 & 0.1734 & -0.0914 \\ 0.1323 & 0.1323 & 0.1376 & -0.1379 & -0.1465 \\ -0.0644 & -0.0644 & -0.0767 & -0.2272 & 0.0445 \end{bmatrix} \tag{10.96}$$

$$K_2 = \begin{bmatrix} -0.4717 & -0.4717 & 0.0740 & -0.3389 & -0.1111 \\ 0.0997 & 0.0997 & 0.1134 & 0.1525 & -0.0882 \\ 0.1316 & 0.1316 & 0.1350 & -0.1497 & -0.1456 \\ -0.0629 & -0.0629 & -0.0715 & -0.2097 & 0.0418 \end{bmatrix} \tag{10.97}$$

$$K_3 = \begin{bmatrix} -0.4708 & -0.4708 & 0.0737 & -0.3393 & -0.1109 \\ 0.1076 & 0.1076 & 0.1036 & 0.1816 & -0.0755 \\ 0.1341 & 0.1341 & 0.1296 & -0.1303 & -0.1380 \\ -0.0692 & -0.0692 & -0.0633 & -0.2350 & 0.0309 \end{bmatrix} \tag{10.98}$$

and

$$\begin{aligned} \epsilon_{1_1} &= 3.4415, \epsilon_{1_2} = 4.1738, \epsilon_{1_3} = 3.2964 \\ \epsilon_{2_1} &= 3.0604, \epsilon_{2_2} = 3.3477, \epsilon_{2_3} = 8.5395. \end{aligned} \tag{10.99}$$

We assume the initial state to be a random variable which is uniformly distributed on $\bar{\mathcal{X}} = [-5, 5]^3$ and the initial state of the controller is fixed to $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. The optimal upper bound obtained is 6.13. Figure 10.1 shows a simulation of the nominal system with the controller and Figure 10.2 shows a simulation of the system with the controller with maximum uncertainty. As it is expected the system is stabilized in both cases but the performance of the system is better in the nominal case.

6 Conclusion

We proposed a new approach for output feedback control of uncertain discrete time piecewise linear systems. The controller design is based on piecewise quadratic Lyapunov function. It is assumed that the controller switching is based on the estimated state and

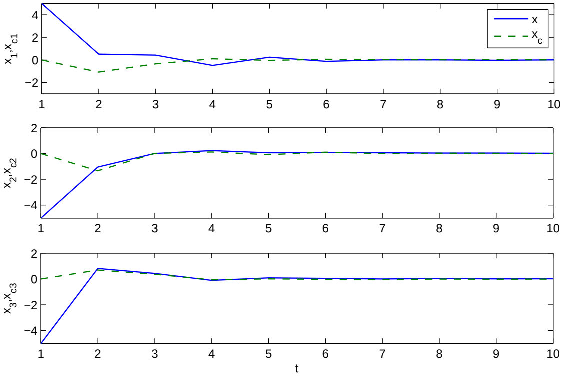

Figure 10.1: Simulation results of the nominal system

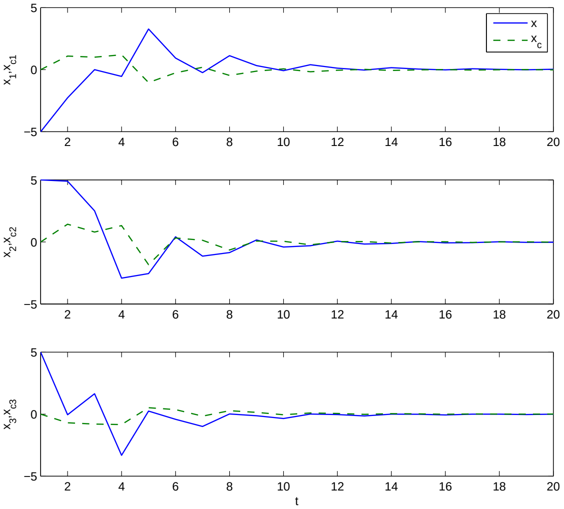

Figure 10.2: Simulation results of system with maximum uncertainty

therefore the transition of the controller and the system are not synchronized. Existence of the controller is casted as the feasibility of a set of BMIs. The controller guarantees an upper bound on the performance cost which can be minimized solving an optimization problem with BMI constraints. The optimization problem is solved using the V-K iteration algorithm.

References

[1] A. Rantzer and M. Johansson, “Piecewise linear quadratic optimal control,” *IEEE Transactions on Automatic Control*, vol. 45, no. 4, pp. 629–637, 2000.

[2] A. Hassibi and S. Boyd, “Quadratic stabilization and control of piecewise-linear systems,” in *Proceedings of American Control Conferenc*, vol. 6, 1998, pp. 3659–3664.

[3] M. Johansson, *Piecewise linear control systems.* Springer-Verlag, 2003.

[4] F. Cuzzola and M. Morari, “A generalized approach for analysis and control of discrete-time piecewise affine and hybrid systems,” *Hybrid Systems: Computation and Control*, vol. 2034, pp. 189–203, 2001.

[5] L. Rodrigues and J. P. How, “Observer-based control of piecewise-affine systems,” in *Proceedings of the 40th IEEE Conference on Decision and Control*, vol. 2, 2001, pp. 1366 –1371.

[6] L. Rodrigue and J. P. How, “Observer-based control of piecewise-affine systems,” *International Journal of Control*, vol. 76, pp. 459–477, 2003.

[7] G. Feng, “Controller design and analysis of uncertain piecewise-linear systems,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 2, pp. 224 –232, 2002.

[8] J. Zhang and W. Tang, “Output feedback H_∞ control for uncertain piecewise linear systems,” *Journal of Dynamical and Control Systems*, vol. 14, no. 1, pp. 121–144, 2008.

[9] ——, “Output feedback optimal guaranteed cost control of uncertain piecewise linear systems,” *International Journal of Robust and Nonlinear Control*, vol. 19, pp. 596–590, 2009.

[10] D. Mignone, G. Ferrari-Trecate, and M. Morari, “Stability and stabilization of piecewise affine and hybrid systems: an LMI approach,” in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 1, 2000, pp. 504 –509 vol.1.

[11] G. Feng, “Observer-based output feedback controller design of piecewise discrete-time linear systems,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 50, no. 3, pp. 448 – 451, mar 2003.

[12] G. M. Bara and M. Boutayeb, “Switched output feedback stabilization of discrete-time switched systems,” in *45th IEEE Conference on Decision and Control*, 2006, pp. 2667 –2672.

[13] D. W. Ding and G. H. Yang, “Static output feedback control for discrete-time switched linear systems under arbitrary switching,” in *American Control Conference*, 2009, pp. 2385 –2390.

[14] M. Kantner, “Robust stability of piecewise linear discrete time systems,” in *Proceedings of American Control Conference*, vol. 2, jun 1997, pp. 1241 –1245 vol.2.

[15] Y. Goa, Z. Liu, and H. Chen, “Robust H_∞ control for constrained discrete-time piecewise affine systems with time-varying parameter uncertainties,” *IET Control Theory and Application*, vol. 3, pp. 1132–1144, 2008.

[16] K. C. Goh, L. Turan, M. G. Safonov, G. P. Papavassilopoulosand, and J. H. Ly, “Biaffine matrix inequality properties and computational methods,” in *American Control Conference*, 1994, pp. 850 – 855.

[17] I. Petersen, “A stabilization algorithm for a class of uncertain linear systems.” *System and Control Letters*, vol. 8, no. 4, pp. 351–357, 1987.

[18] J. Löfberg, “YALMIP : A toolbox for modeling and optimization in MATLAB,” in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[19] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial Mathematics, 1994.