

1-1-2017

On Detectability Of Networked Discrete Event Systems

Yazeed Sasi
Wayne State University,

Follow this and additional works at: http://digitalcommons.wayne.edu/oa_dissertations



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Sasi, Yazeed, "On Detectability Of Networked Discrete Event Systems" (2017). *Wayne State University Dissertations*. 1870.
http://digitalcommons.wayne.edu/oa_dissertations/1870

This Open Access Dissertation is brought to you for free and open access by DigitalCommons@WayneState. It has been accepted for inclusion in Wayne State University Dissertations by an authorized administrator of DigitalCommons@WayneState.

ON DETECTABILITY OF NETWORKED DISCRETE EVENT SYSTEMS

by

YAZEED SASI

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2017

MAJOR: ELECTRICAL ENGINEERING

Approved By:

Advisor

Date

© COPYRIGHT BY

YAZEED SASI

2017

All Rights Reserved

DEDICATION

I dedicate this dissertation to my parents who have worked hard to support me. Also, to all my siblings and friends.

ACKNOWLEDGEMENTS

I am very fortunate to work under the supervision of Professor Feng Lin, one of the greatest names in the area of discrete event systems. Professor Lin's guidance was the candle that led my way through the Ph.D. journey. Therefore, I am very grateful and thankful for his ultimate support. Also, I would like to thank my committee members, Associate Professor Caisheng Wang, Professor Hao Ying, and Professor Nabil Chalhoub, for their valuable comments and suggestions. I would like also to thank my friend Processor Abdurrahman Hakemi for his support. Also, I want to thank my friend Saleh Emparik for his support and cheering; you will be always living in my heart. Finally, special thanks go to my Father, Sasi Omran, the man who is behind me all the time.

TABLE OF CONTENTS

DEDICATION.....	ii
ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	vii
LIST OF TABLES	ix
CHAPTER 1 INTRODUCTION.....	1
1.1 Overview of Discrete Event Systems.....	2
1.1.1 Modeling of Discrete Event System.....	2
1.1.2 Detectability of Discrete Event Systems	4
1.2 Problems and Motivation	6
1.3 Dissertation Organization.....	7
CHAPTER 2 RELATED WORK.....	9
CHAPTER 3 NETWORKED DISCRETE EVENT SYSTEMS AND OBSERVATION	17
3.1 Mathematical Background	17
3.2 Network Detectability of Discrete Event Systems	26
3.2.1 Definitions of Network Detectabilities.....	28
3.2.2 Checking Network Detectabilities.....	29
3.2.3 An Algorithm to Check Network Detectabilities of Discrete Event Systems	34
3.3 An Illustrative Example	36
CHAPTER 4 D-DETECTABILITY OF NETWORKED DISCRETE EVENT SYSTEMS	40

4.1 Definitions of Networked D-detectabilities.	40
4.2 Checking Network D-detectabilities	42
4.3 An Algorithm to Check Network D-detectabilities of Discrete Event Systems	47
4.4 Illustrative Examples.....	49
CHAPTER 5 PROPERTIES OF NETWORK DETECTABILITY	55
5.1 Properties.....	55
5.2 Illustrative Examples.....	58
CHAPTER 6 I-DETECTABILITY OF NETWORKED DISCRETE EVENT SYSTEMS	61
6.1 Mathematical Background and Network I-observer	62
6.2 Definitions of Network I-Detectabilities.....	65
6.3 Checking Network I-detectabilities.....	65
6.3.1 Checking Network I-detectabilities with Observation Losses	66
6.3.2 Checking Network I-detectabilities without Observation Losses.	68
6.4 An Algorithm to Check Network I-detectabilities of Discrete Event Systems with Observation Losses.	72
CHAPTER 7 NETWORK CO-DETECTABILITY	74
7.1 Mathematical Background	75
7.2 Definitions of Network Co-Detectabilities	77
7.3 Checking Network Co-detectabilities	78
7.4 An Algorithm to Check Network Co-detectabilities of Discrete Event Systems	80

CHAPTER 8 CONCLUSION.....	82
REFERENCES.....	83
ABSTRACT.....	95
AUTOBIOGRAPHICAL STATEMENT.....	97

LIST OF FIGURES

Figure 1. System classification.	3
Figure 2. Printer as a discrete event system.	4
Figure 3. A networked discrete event system.	6
Figure 4. Discrete event system G of Example 2.....	24
Figure 5. $GL=LOSS(G)$ of Example 2	25
Figure 6. Observer $GL, obs = OBS(GL)$ of Example 2.....	25
Figure 7. Networked observer $GDL, obs=DL(OBS(GL))$ of Example 2	26
Figure 8. The discrete event system G of Example 3	36
Figure 9. $GL = LOSS(G)$ of Example 3.	37
Figure 10. Observer GL, obs of the system in Example 3	37
Figure 11. Networked observer GDL, obs of the system in Example 3.....	38
Figure 12. The modified discrete event system G' of Example 3	38
Figure 13. Observer $G'L, obs$ of the modified system in Example 3.....	39
Figure 14. Networked observer $G'DL, obs$ of the modified system in Example 3	39
Figure 15. A discrete event system G representing a nuclear reactor.....	49
Figure 16. Automaton GL of the system in Example 4.....	49
Figure 17. Observer GL, obs of the system in Example 4	50
Figure 18. Networked observer GDL, obs of the system in Example 4.....	50
Figure 19. Discrete event system G of the factory.....	51
Figure 20. Networked observer GDL, obs , for the first few states, with $N = 1$	52
Figure 21. Networked observer GDL, obs , for the first few states, with $N = 2$	54
Figure 22. Networked observer $GDL, obsN$ for $N \geq 2$ of Example 6.....	59

Figure 23. Networked discrete event system G of Example 6.....	60
Figure 24. Networked observers $GDL, obsN$ for $N=1,2$ of Example 6	60
Figure 25. Example 8.....	67
Figure 26. Network I-Observer for Example 8.....	68
Figure 27. Network I-Observer for Example 9 $N=1$	71
Figure 28. Network I-Observer for Example 9 $N=0$	72

LIST OF TABLES

Table 1. Some operations on languages.....	18
Table 2. Logic connectives.	18

CHAPTER 1 INTRODUCTION

Discrete event systems have been studied for more than three decades. During this time, the theory of discrete event systems has developed in many aspects. Supervisory control is the main control theory developed for discrete event systems [1-6]. The supervisory control is based on the concept of controllability and observability. The controllability requires that all events to be disabled must be controllable [3]. The observability requires that whenever there are two strings that look the same to the supervisor, the control action after them must be consistent [7]. Controllability and observability characterize the existence condition for a supervisor. Several extensions to basic supervisory control theory have been investigated [8]. Examples include decentralized control [2, 9], on-line control [10], limited or variable lookahead control [11], and robust and adaptive control [12]. In order to control or supervise a discrete event system, we need to estimate the current state of the system. State estimate problems of discrete event systems are first investigated by Wonham [13], Ramadge [14], and Ozveren and Willsky [15]. Since then it has become one of the important problems in discrete event systems. If we cannot determine which state the system is in, then we want to know the set of all possible state that the system may be in. We call this set “state estimate”. There are many examples to show the importance of state estimation. For example, the state estimate of a train is important, if at some point, two trains have to use the same railroad. We need to make sure that we can accurately estimate the state of each train (train’s location) in order to avoid collision. Another important application for state estimation is in medicine, where estimating the disease stage is very important. Therefore, the first question to ask is: Can we determine which state the system is currently in? If the answer is “no”, then the second question is: Can we distinguish safe states from unsafe states? Detectability theory attempts to answer these and other questions. In this dissertation, we investigate detectability of

networked discrete event systems, which ensures that the states of a discrete event system can always be detected when delays and losses are introduced into the system.

1.1 Overview of Discrete Event Systems

1.1.1 Modeling of Discrete Event System

The system can be defined as, according to IEEE Standard Dictionary of Electrical and Electronic Terms, a set of components act together to perform a specific function not possible with individual parts [1]. The system in general can be classified into *dynamic* and *static* systems. In dynamic systems, the output of the system depends on the past values of input. In contrast, the output of the system is independent of the previous values of the input in case of static systems. Dynamic systems can be either time-varying or time-invariant systems. In time-invariant dynamic systems the output of the systems does not depend on time explicitly. Most of the systems we deal with in system analysis are classified as time-invariant dynamic systems. Depending on the nature of the system, time-invariant dynamic systems can be further classified to linear or nonlinear systems. Furthermore, nonlinear time invariant dynamic systems can be classified to continuous-state and discrete-state. Figure 1 illustrates system classification.

Discrete event system can be classified as a nonlinear time-invariant dynamic system. Discrete event system is a system that moves from one state to another when an event occurs. Discrete event systems are called event driven systems because the system stays in one of its states until the occurrence of the next event. One of the methods used to describe a discrete event system is the automaton.

Many systems can be modeled as discrete event systems that consist of discrete states and discrete events [16-18]. For example, the printer can be in three possible states: working, idle, and broken-down state. Transition from working state to idle state can happen when the printer finishes the current task, so we call that the event that causes the printer to change its state from working

state to idle state. The transitions between states are called events of the discrete event system. The system generates infinite sequences of events known as strings. A set of strings is defined as language. The set of all strings started from the initial state is defined as prefix-closed language. Likewise, if the language ended in a marked state, we call it marked language.

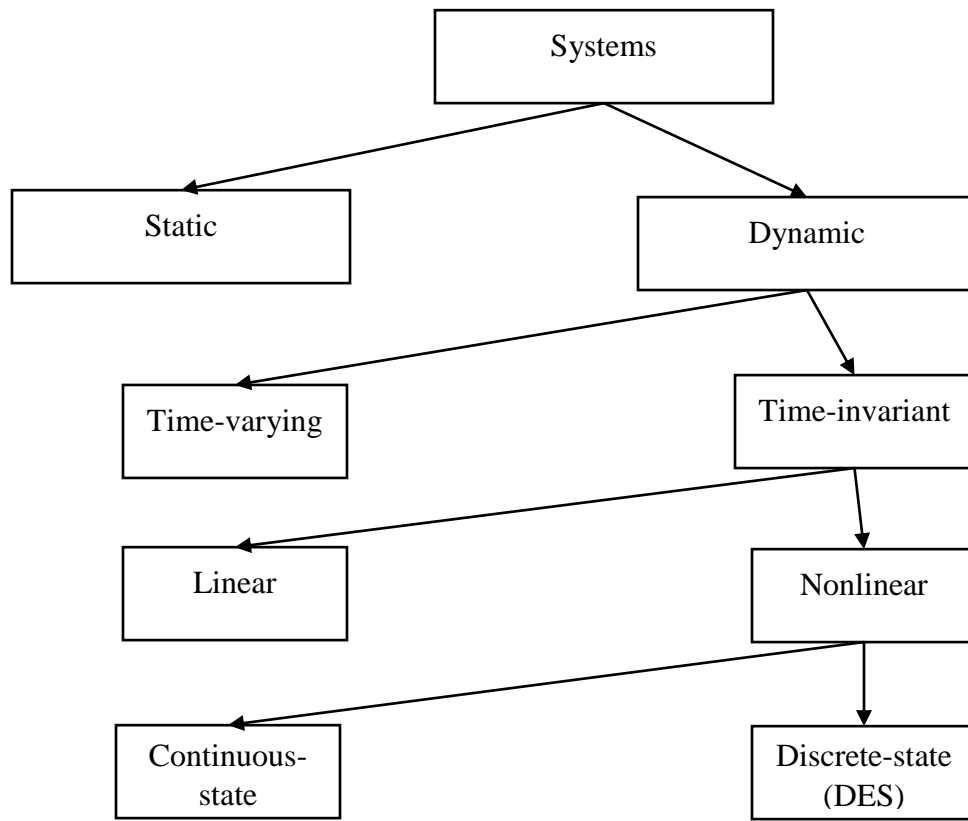


Figure 1. System classification.

Figure 2 shows how printer can be viewed as a discrete event system. In this discrete event system, the states are: working state, idle state, and broken state. The events are: α , β , λ , and τ .

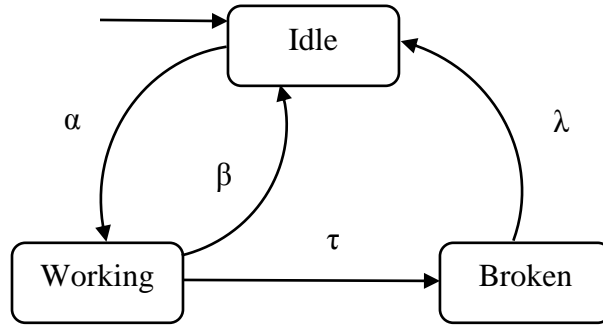


Figure 2. Printer as a discrete event system.

1.1.2 Detectability of Discrete Event Systems

Shu and Lin [19] and Shu et al. [20] investigate state estimates and detectability systematically. Detectability is defined as the ability to determine the current state and subsequent states of a system based on observation. In [19], four types of detectabilities are defined with different uses in different applications. Intuitively, they can be explained as follows. (1) Strong Detectability is the ability to determine the current state and subsequent states of the system for *all* trajectories of the system after finite number of observation. (2) Detectability is the ability to determine the current state and subsequent states of the system for *some* trajectories of the system after finite number of observation. (3) Strong Periodic Detectability is the ability to periodically determine the current state of the system for *all* trajectories of the system. (4) Periodic Detectability is the ability to periodically determine the current state of the system for *some* trajectories of the system. Depending on whether the requirement is strong or weak in an application, different detectabilities can be used. Strong detectability is the strongest among the four, while periodic detectability is the weakest.

A problem related to detectability is diagnosability, which is investigated extensively in discrete event systems. The earliest works on diagnosability appear in [21-64] and many subsequent works have been done. In diagnosability study, a discrete event system may fail. The failures are described by events, which are unobservable. A discrete event system is diagnosable if the failure

events can be detected after some finite observations of events. The difference between detectability and diagnosability is that while detectability tries to detect the state of a system, diagnosability tries to detect failure event in a system.

Another property related to detectability is opacity [65-71]. While detectability describes the ability to determine the states of a discrete event system, opacity describes the ability to hide the states or other information about a system. Obviously, detectability and opacity are used for different applications.

In some applications, determining current state and subsequent states of discrete event system may be too restrictive. Instead, we may need to check whether the system stays in subset of states after finite numbers of observations. Therefore, D-detectability was introduced and investigated by Shu and Lin [19]. D-detectability reduces the need of checking the current and subsequent states to just distinguish certain pairs. D-detectability is defined as the ability to distinguish certain state pairs instead of the current and subsequent states of the system. There are some applications that just require the d-detectability; for instance, checking observability of a language, checking diagnosability of a language, checking feasibility of communication system, and checking detectability of a system [72].

Detectability of discrete event systems has been extended to other classes of detectabilities. For example, I-detectability, Delayed detectability and Co-detectability. I-detectability is defined as the ability to determine the initial state of the system [73]. I-detectability is usually significant in the problems that requires knowing the initial state of the system. Another type of detectability is the delayed detectability. Delayed detectability checks the state of the discrete event systems after observing $k_1 + k_2$ observable events [74]. In Co-detectability, the objective is to define the current state of the system when we have a set of local agents [75]. The discrete even system is

called co-detectable if at least one local agent can determine the current state of the system after finite number of observations. We need Co-detectability when we have distributed systems.

1.2 Problems and Motivation

All detectabilities investigated so far assume that communications between the agent/supervisor and the plant are reliable and instantaneous. In other words, there is no delay and/or loss in communication. This assumption may be true for non-networked discrete event systems but is not true for networked discrete event systems. In a networked discrete event system (Figure 3), where communication between the agent/supervisor and the plant are carried out over a shared communication network, communication delays and losses are unavoidable [76, 77]. How to handle communication delays and losses is an important problem in networked systems, including networked discrete event systems. Control of networked discrete event systems is investigated in [78-86]. Since intermittent sensor failures are equivalent to losses in communication, a new language operation that allowed address communication losses (but not delays) in diagnosis of networked discrete event systems has been introduced in [87].

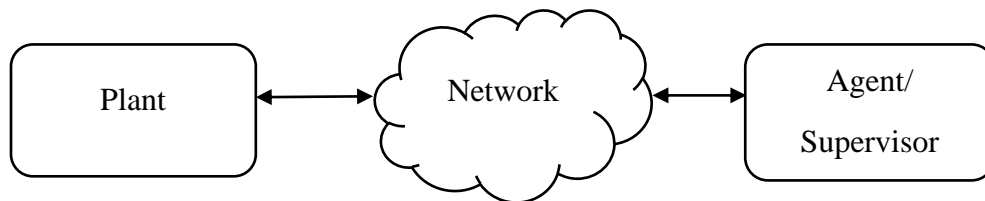


Figure 3. A networked discrete event system.

Networked systems are now widely used in everyday life, because it is modular, flexible, scalable, easy to update, diagnose, and maintain. Because more and more systems are networked systems, it is important to investigate detectability of networked discrete event systems. Introducing the delay and/or loss means that all types of the detectability need to be modified or changed to be applied for the networked discrete event systems. This problem is very serious when

the supervisor misses to detect an event that may take the whole system to a prohibited state and cause the system to stop or crash. To prevent unpleasant consequences, we must be able to detect the discrete event system even under the case of delays and losses.

We assume that the communication channel satisfies FIFO (first in first out) property. In other words, messages may be delayed, but the order in which they will be received is same as the order they are sent. This assumption is made in all works in networked discrete event systems. It is a reasonable assumption if messages are sent using a single channel. On the other hand, if this assumption is violated, then it will be very difficult, if not impossible, to estimate the state of the system from the sequence of events observed, because order is most essential in event sequences.

1.3 Dissertation Organization

The remaining dissertation is organized into three chapters and can be summarized as follows.

In chapter 2, we conduct literature review about what have been done in the area of detectability of discrete event system to give an idea about the subject.

In chapter 3, we introduced different notations used to formalize networked discrete event systems. We assume that the systems can be nondeterministic. We also consider both communication delays and losses. We review how to estimate states under communication delays and losses. Moreover, we define network detectability and strong network detectability. We derive necessary and sufficient conditions for network detectability and strong network detectability. We develop algorithms to check whether a discrete event system is network detectable and/or strongly network detectable.

In chapter 4, we investigate D-detectability of networked discrete event systems. Four types of networked D-detectabilities are defined along with the algorithm to check the different

types of networked D-detectabilities. We give an example of power distribution system as networked discrete event system.

In chapter 5, we discuss various properties of networked discrete event systems. We also give some examples to illustrate these properties. Most of the properties are valid for both networked detectability and D-detectability.

In chapter 6, we investigate I-detectability of networked discrete event systems. This chapter consists of four sections. First, mathematical background required for investigating network I-detectability. Second, definitions of I-detectabilities of networked discrete event systems. Third, checking I-detectabilities of networked discrete event systems. Last, an algorithm to check I-detectabilities of networked discrete event systems.

In chapter 7, we study co-detectability of networked discrete event systems. Like chapter 6, this chapter consists of mathematical background required for investigating network co-detectability, definitions of co-detectabilities of networked discrete event systems, checking I-detectabilities of networked discrete event systems, and an algorithm to check I-detectabilities of networked discrete event systems.

In chapter 8, we conclude and summarize our work and point out the main contribution of our dissertation.

CHAPTER 2 RELATED WORK

State estimation problems of discrete event systems was first investigated in 1986 by Ramadge [14], and since then it becomes one of important problems. Ramadge used a nondeterministic automaton model for a discrete event system to determine the current state of the system from a sequence of past events. The scheme has demonstrated valuable in the theoretic examination of number of fundamental supervisory control issues [88-90]. The motivation to study state estimation problem for the author came from the importance of state estimate in supervisory control. In [14], weak observability, strong observability, and coobservability are investigated. In weak observability, there is no two different states in G that have the same sets of event and output trajectories. Strong observability, on the other hand, is defined as there is no different states in G that have common event sequence that can generate a common output sequence [14]. Ramadge started with a nondeterministic automaton $G = (Q, \Sigma, \delta)$, which means that the initial state is unknown. He concluded that pair (G, h) is trackable if for each pair $(\sigma, q) \in \Sigma \times Q$ if $q_1, q_2 \in \delta(\sigma, q)$ with $q_1 \neq q_2$, then $h(q_1) \neq h(q_2)$. Therefore, the next state can be uniquely determined when we know the current state, the next event and the next output. Also, the author introduced the observation algebra; a subset A of Q is said to be an observation algebra for G if for each $\sigma \in \Sigma$

$$S \in A \text{ implies } \delta(\sigma, S) \in A$$

The conception of observation algebra can be used to solve some tracking problems with minimum of oracle consultations.

In 1988, Caines et al. [91] presented a dynamical logic observer to estimate the current state of input-state-output automaton. the main reason of the paper is to show that Artificial Intelligence and Systems and Control Theory are related [92, 93]. The author used simple

dynamical systems represented by partially observed automata to explore the state estimation problems. The state estimations have been constructed from automata using two forms, construction of classical dynamical system and the construction of dynamical logic system. In classical dynamical system, the system creates a sequence of state estimates. Dynamical logic system, on the other hand, creates sequences of propositions that properly describe the properties of state of the automaton. The paper is basically divided into four main parts. The first part discusses the dynamical observer problem for finite automata. In this section, a deterministic state output finite automaton has been used to model the dynamical observers. Moreover, the section suggests several definitions related to the dynamical observers. The second part of the paper presents the dynamical logic systems. Also, this section introduces some definitions to define some properties of the dynamical logic observers. Third section presents the main theorem that links the observability of input-state-output automaton with the existence of a convergent classical dynamical observer and the existence of a convergent dynamical logic observer [94]. Furthermore, this section shows the general design procedure of the classical dynamical observer for the system output automaton using the notation of DAG observer tree. In the fourth section of the paper, the authors give an example to explain the state estimation problem using the concepts presented in their paper. In conclusion, the paper presented a new conception of a dynamic logic systems or DLS. The paper presented new type of observers, dynamical logic observers. The dynamic logic observer is a DLS designed to yield a state estimates for dynamic system whose dynamics can be specified in the dynamical axioms of DLS [91].

In 1990, Özveren and Willsky [15] introduced concepts of observability and resiliency for discrete event systems [16, 95, 96]. The paper consists of three main sections, and we will try to briefly summarize these sections. In the first section, the authors presented the mathematical

background needed to pursue further in the paper. The authors characterized the notions of state observability, persistent states and always-observability, indistinguishability, and observability with delay. Also, the section suggests algorithms to construct suitable observers. The second section, however, discusses the observer implementation and complexity. The main objective of this section is to argue the complexity of the constructed observer. The computation complexity of the observer discussed can be executed polynomial time, but the cardinality of the state space could be exponential in some cases. The third section of the paper talks about the resiliency of the observers constructed in section one. For example, the authors wanted to know how resilient the observer is in case there is an error in the output string we observe. The authors showed that if the system is observable, then the error propagation will never occur; this means the observer is always resilient. In summary, the authors had developed polynomial algorithms to check the observability and build resilient observers; the observer O_R is always resilient as long as the system is observable. However, the cardinality of the observer's state space can be exponential.

In [20] Shu, Lin, and Ying defined the detectability in discrete event systems by the observation of some event observation and/or some state observation. They assume that the state of the system is not known in the beginning. Detectability of discrete event systems is very important especially when it comes to medical application [97, 98]. The paper is divided into two sections. The first section was to define the basics of the discrete event systems, and how many system can be modeled by the discrete event systems. The discrete event system is modeled using automaton of the generator.

$$G = (Q, \Sigma, \delta)$$

where Q is the set of discrete states, Σ the set of events, and $\delta: Q \times \Sigma \rightarrow Q$ the transition function. As it has been mentioned before, the state estimation used is based on observation of some events

and/or some states. The event observation is described projection $P: \Sigma^* \rightarrow \Sigma_o^*$. The output observation, on the other hand, is described by output map $h: Q \rightarrow Y$. To simplify things, the authors assume that the automaton G is deadlock free so that at least one event is defined for the system at any time. The second section of the paper was designated for state estimation and detectabilities. In this section, the authors defined four properties of the detectabilities. Strong detectability, weak detectability, strong periodic detectability, and weak periodic detectability were defined. Moreover, the authors constructed an observer to check the four types of detectabilities. The observer is used to check detectability through four criterions. The necessary and sufficient conditions for the four types of detectability were driven and tested by constructing an observer. The observer constructed has an exponential complexity, so more time is required to check the detectability of the system.

In [19], Shu and Lin modified the work proposed in [20], and they used a nondeterministic automaton instead of deterministic automaton. However, [19] presented some extra work; for example, the authors devolved a technique, which called detector, to check strong detectability and periodic strong detectability with the polynomial complexity instead of exponential complexity. Another contribution of [19] was introducing D-detectability, which relaxes the requirement of estimating the current of the system to just distinguishing certain pairs of state. this type of detectability is useful in the case where determining the current state and subsequent states is too restrictive. The paper has three main sections. In the first section, the authors started with nondeterministic automaton, and they redefined detectability, detectability, strong periodic detectability, and periodic detectability. The problem of checking these four types of detectability has been solved by constructing a G_{obs} observer. Constructing G_{obs} can be done by changing all the unobservable events in the automaton to empty string and converting the nondeterministic

automaton to a deterministic automaton. By construction the observer, Shu and Lin checked the four types of detectabilities. Polynomial algorithms were the main topic for the second section of the paper. In this section, a detector, G_{det} , for checking detectability was proposed because the computational complexity of the constructing the observer is exponential. The detector reduces the computational complexity from exponential to polynomial, and that was a great contribution in this paper. The last section for this paper was about d-detectability. D-detectability is an extension of the detectability. In D-detectability, the requirement for determining the current state of the discrete event system is relaxed to just distinguishing certain pairs of states of the system. The authors also defined four types of D-detectability (strong D-detectability, D-detectability, strong periodic D-detectability, and periodic D-detectability). Briefly, the work provided in [19] has added some extra effort to [20]. For example, a nondeterministic discrete event system automaton has been used instead of deterministic discrete event system automaton. Also, the computation complexity of checking strong detectability and strong periodic detectability is reduced to polynomial by constructing a detector. Another contribution of this paper is introducing D-detectability, an extended form of detectability.

Shu and Lin also published an IEEE paper [73] in 2013; the paper investigated the initial state estimation or I-detectability of the discrete event systems. I-detectability defined as the ability of estimating the initial state of the system. I-detectability is important in some applications such as offline fault diagnosis [99, 100]. The importance of initial state detectability comes from the fact that sometimes we may need to determine the state of the system after the occurrence of a failure, so it would be easy to fix the system. In [73] two types of I-detectabilities are defined: weak I-detectability and strong I detectability. Besides, I-observer was constructed to check strong I-detectability and weak I-detectability. Authors also constructed the I-detector to check I-

detectability in polynomial complexity. In the first section of the paper, an introduction to the modeling of the discrete event system has been provided, and definitions for I-detectability are given. The second section in this paper was about I-observer and I-detector. I-observer has been constructed to check both types of initial state detectabilities. Because the computational complexity for I-observer was exponential, I-detector has been also constructed to reduce complexity to polynomial. However, Constructing I-observer and I-detector are more complex than constructing G_{obs} and G_{det} because more modifications are required to be applied to the discrete event systems. In the last section, the authors introduced closed-loop I-detectability. When we have weakly I-detectable system, the system is called closed-loop strongly detectable if we can come up with appropriate controller to attain that. The authors developed an algorithm to check if the system is closed-loop detectable or not.

In 2013, delayed detectability of discrete event systems was proposed by Shu and Lin in [74]. The authors extended the detectability problem to delayed detectability. The delayed detectability investigates system state at event k_1 th after observing $k_1 + k_2$ observable events. The paper is divided into four parts. In part one, the authors introduced the discrete event system, and they used nondeterministic automaton to model the discrete event system. However, there were two assumptions used to describe the automaton: First, the automaton is deadlock free. Second, no loops in the automaton contain only unobservable events. Moreover, the authors defined the delayed detectability, or (k_1, k_2) -detectability, as “A discrete event system G is (k_1, k_2) -detectable if after k_1 event observations, we can determine the state of the system after k_2 steps of delays for all trajectories” [9]. In the second part of the paper, various properties of delayed detectability were investigated and proved. Also, in order to check whether the system is delayed detectable or not, an observer has been constructed to check the delayed detectability.

Because the computation complexity of the observer is exponential (bounded by $2^{|Q|}$), a detector G_{det} has been constructed. The cardinality of state space of the detector is bounded by $|Q|^2 + 1$. In the third part, however, the authors suggested four algorithms to check whether a system is (k_1, k_2) -detectable or not for a given k_1 and k_2 . In the fourth part of the paper, the relation between delayed detectability with observability, diagnosability, and detectability are discussed. In summary, the paper investigated the delayed detectability. Also, the authors provided the proofs for various properties of delayed detectability. Another important contribution for this paper is that it provided efficient polynomial algorithm to check delayed detectability.

In [75], Shu and Lin investigated co-detectability or decentralized detectability. Co-detectability is an extension for the detectability. Co-detectability investigates the detectability of the discrete event systems when we have a set of local agents, and each agent has limited observations. Co-detectability can be defined as the ability to determine the current state of the system after limited number of observations using at least one local agent. It is very important to point out that the agents do not communicate among themselves. There are four types of co-detectabilities, and they are co-detectability, strong co-detectability, strong periodic co-detectability, and periodic co-detectability. The paper consists of two parts. The first part was to introduce and defined each type of co-detectability. For example, co-detectability was defined as “the discrete event system is called co-detectable if the current state and subsequent states of the system is known to at least one agent for some trajectories of the systems after finite number of observations”. In the second part of the paper, a co-observer was introduced to check co-detectability of the discrete event systems. Based on the co-observer, theorems to check all types of co-detectabilities of the discrete event systems were introduced. Moreover, the authors constructed a co-detector to check the strong versions of co-detectabilities in polynomial

complexity because the cardinality of the state space of the co-observer is exponential. Overall, the paper extended the detectability of centralized discrete event systems to decentralized discrete event systems. Various types of co-detectabilities were defined and checked using co-observer. Also, to reduce computation complexity, a co-detector was constructed to check strong co-detectability and strong periodic co-detectability in the polynomial complexity. The author suggested to investigate the case when there is some communication between the agents as a future work.

CHAPTER 3 NETWORKED DISCRETE EVENT SYSTEMS AND OBSERVATION

3.1 Mathematical Background

In a networked discrete event system, the agent/supervisor and the plant are connected via a communication network. We assume that the networked discrete event system is modeled by a nondeterministic automaton [1, 7, 101]:

$$G = (Q, \Sigma, \delta, Q_0)$$

where Q is the finite state set, Σ is the finite event set, $\delta : Q \times \Sigma \rightarrow 2^Q$ is nondeterministic transition function, and Q_0 is the set of possible initial state. The language generated by G is denoted by $L(G)$ [102]. Language is defined as the set of all possible trajectories over Σ . Language is a special type of set, and all set operations are applicable on the languages. If $\Sigma = \{a, b, c\}$, we can have:

$$L_1 = \{\varepsilon, a, abb, babbc\},$$

$$L_2 = \{\text{all possible strings of length 3 with event } a\} = \\ \{aaa, aab, aac, aba, abb, abc, aca, acb, acc\},$$

$$\text{or } L_3 = \Sigma^* = \{\varepsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, \dots\}$$

however, there are some operations that apply to just languages, and these include Concatenation, Prefix-closure, Kleen-closure. Table 1 shows some operations that can be applied to languages.

The events are classified into observable events Σ_o and unobservable events Σ_{uo} . We use Σ^* to represent all possible strings over Σ . A string is defined as a finite sequence of events [103]. There are some basic operations that can be made over a string. For example, assume we have two strings $\alpha\beta\gamma$ and $\delta\eta\lambda$, then there are some basic operations that can be done of these strings. First, concatenation of the last two strings will be $\alpha\beta\gamma\delta\eta\lambda$. Second, there is an identity element for any string, and that element is empty string. Empty string is usually represented by " ε ", so $\varepsilon\alpha\beta\gamma =$

$\alpha\beta\gamma\varepsilon = \alpha\beta\gamma$. Finally, if we have the string $s = \alpha\beta\gamma$, α is a prefix of s , denoted by $\alpha \leq s$; β is a substring of s ; γ is a suffix of s .

Table 1. Some operations on languages.

Operation on languages	Definition
Concatenation	Let $L_1, L_2 \subseteq \Sigma^*$, then $L_1L_2 = \{s \in \Sigma^* : s = s_1s_2 \wedge s_1 \in L_1 \wedge s_2 \in L_2\}$.
Prefix-closure	Let $L \subseteq \Sigma^*$, then $\bar{L} = \{s \in \Sigma^* : \exists t \in \Sigma^*)st \in L\}$
Kleen-closure	Let $L \subseteq \Sigma^*$, then $L^* = \{\varepsilon\} \cup L \cup LL \cup LLL \cup \dots$

In discrete event systems, we deal with sets and set notations, so it is very useful to explain various set operations and abbreviations. In metaethical statements, we use symbols instead of words to describe a specific mathematical relation. There are some basic connectives to describe basic logic connectives of sets as shown in Table 2 [104].

Table 2. Logic connectives.

Logic operator	Meaning
\vee	“or” or “disjunction”
\wedge	“and” or “conjunction”
\sim or \neg	“not” or “negation”
\Rightarrow	“if...then”
\Leftrightarrow	“if and only if”
\forall	“for all”
\exists	“there exists”

\in	“belongs to”
\notin	“does not belong to”
$-$	“difference”
\subset	“subset”
$ set $	“cardinality”
$ string $	“length of string”
\cup	“union”
\cap	“intersection”
\times	“product”
2^A	“power set”
\emptyset	“empty set”

There are some properties of empty string " ε " and empty set \emptyset . We can summarize these properties as shown below [1]:

1. The empty string does not belong to the empty set, that is, $\varepsilon \notin \emptyset$.
2. $\{\varepsilon\}$ is a nonempty language that contains just the empty string, that is, $\emptyset \neq \{\varepsilon\}$.
3. $\emptyset^* = \{\varepsilon\}$, and $\{\varepsilon\} = \{\varepsilon\}^*$.
4. If $L = \emptyset$ then $\bar{L} = \emptyset$ ($L = \emptyset \Rightarrow \bar{L} = \emptyset$), also $L = \emptyset \Leftrightarrow \bar{L} = \emptyset$ is true.
5. If $L \neq \emptyset$ then $\varepsilon \in \bar{L}$ ($L \neq \emptyset \Rightarrow \varepsilon \in \bar{L}$), also $(L \neq \emptyset \Leftrightarrow \varepsilon \in \bar{L})$ is true.

Because we deal with sets when we deal with discrete event systems, it is important to mention basic set axioms. There are six axioms of set theory. These axioms can be summarized as follows [104]:

1. The axiom of containment: if all elements in a set A is also elements in set B , then A is a subset of B denoted as $A \subset B$,

$$A \subset B \Leftrightarrow \forall a(a \in A \Rightarrow a \in B).$$

It is important to mention that any set is a subset of itself.

2. The axiom of extension: two sets, A and B , are equal if and only if each both sets have the same elements:

$$A = B \Leftrightarrow \forall x((x \in A \Rightarrow x \in B) \wedge (x \in B \Rightarrow x \in A)).$$

3. The axiom of intersection: for any two sets, A and B , the class of elements that are in belonging to both sets A and B is also a set:

$$\forall(A, B) \exists M (y \in M \Leftrightarrow y \in A \wedge y \in B).$$

4. The axiom of union: for any two sets, A and B , the class of elements that belonging to either A or to B is also a set:

$$\forall(A, B) \exists M (y \in M \Leftrightarrow y \in A \vee y \in B).$$

5. The empty set: the empty set is a set that has no elements. It also called null or void set. We denote that set by \emptyset .

6. Power set axiom: for a set C , there is a special class, the collection of all subsets of the set C .

$$\forall C \exists P(C) (\forall B ((B \in (P(C)) \Leftrightarrow (B \subset C)))$$

Example 1

Assume we have two sets $A = \{a, b\}$, and $B = \{b, c\}$, then:

$$A \cup B = \{a, b, c\},$$

$$A \cap B = \{b\},$$

$$A - B = \{a\},$$

$$A \times B = \{(a, b), (a, c), (b, b), (b, c)\},$$

$$|A \times B| = 4,$$

$$\text{and } 2^A = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}.$$

Another type of operation that can be done over strings and languages is projection or natural projection. Given a large set of events Σ_l and small sets of events Σ_s such that $\Sigma_s \subset \Sigma_l$. Projection on strings is a mapping from large set of events Σ_l to small set of events Σ_s [1].

$$P: \Sigma_l^* \rightarrow \Sigma_s^*$$

where

$$P(\varepsilon) = \varepsilon$$

$$P(\sigma) = \begin{cases} \sigma & \text{if } \sigma \in \Sigma_s \\ \varepsilon & \text{if } \sigma \in \Sigma_l - \Sigma_s \end{cases}$$

$$P(s\sigma) = P(s)P(\sigma)$$

Projection deletes all the events that do not belong to the small event set Σ_s . If $\Sigma_l = \{\alpha, \beta, \mu\}$, $\Sigma_s = \{\alpha, \beta\}$, and $L = \{\mu, \mu\beta\alpha\beta, \beta\mu\beta\alpha\beta\mu\beta\}$, then $P(L) = \{\varepsilon, \beta\alpha\beta, \beta\beta\alpha\beta\beta\}$.

The inverse projection can be defined as $P^{-1}(t) = \{s \in \Sigma_l^* : P(s) = t\}$. There are several properties for projection (P) and inverse projection (P^{-1}) that can be summarized in the following [1]:

1. $P(P^{-1}(L)) = L$
2. $L \subseteq P(P^{-1}(L))$
3. $P(A \cup B) = P(A) \cup P(B)$
4. $P(A \cap B) \subseteq P(A) \cap P(B)$
5. $P^{-1}(A \cup B) = P^{-1}(A) \cup P^{-1}(B)$
6. $P^{-1}(A \cap B) = P^{-1}(A) \cap P^{-1}(B)$
7. $P(AB) = P(A)P(B)$

$$8. P^{-1}(AB) = P^{-1}(A)P^{-1}(B)$$

$$9. A \subseteq B \Rightarrow P(A) \subseteq P(B) \wedge P^{-1}(A) \subseteq P^{-1}(B)$$

We used δ to denote the set of all transitions in G : $\delta = \{(q, \sigma, q') : q' \in \delta(q, \sigma)\}$. The set of observable transitions is denoted by $\delta_o = \{(q, \sigma, q') \in \delta : \sigma \in \Sigma_o\}$. The set of unobservable transitions is denoted by $\delta_{uo} = \{(q, \sigma, q') \in \delta : \sigma \in \Sigma_{uo}\}$. Some observable transitions may be lost in communication. These transitions are denoted by δ_L ($\delta_L \subseteq \delta_o$) [79, 103].

We denote the observation mapping under the communication losses by θ_L . After occurrence of the string s in the system, the agent/supervisor will observe $\theta_L(s)$. Assume the string $s = \sigma_1 \dots \sigma_i \dots \sigma_k$, $\theta_L(s)$ is obtained by replacing σ_i with empty string (ϵ) if the corresponding transition is $(q_i, \sigma_i, \delta(q_i, \sigma_i)) \in \delta_{uo}$, with σ_i if $(q_i, \sigma_i, \delta(q_i, \sigma_i)) \in \delta_o - \delta_L$, and with ϵ or σ_i if $(q_i, \sigma_i, \delta(q_i, \sigma_i)) \in \delta_L$. Since $\theta_L(s)$ is not unique, θ_L is the mapping from $L(G)$ to 2^{Σ^*} [79]:

$$\theta_L: L(G) \rightarrow 2^{\Sigma^*}.$$

We denote the delayed observation with delays bounded by N steps as θ_D^N . For string $s \in L(G)$

$$\theta_D^N(s) = \{s_{-i} : i = 0, 1, \dots, N\},$$

where s_{-i} is the prefix of s with the last i events removed. If a string $s \in L(G)$ occurred, observation delays will change s to one of the strings in $\theta_D^N(s)$, because the last events may not be observed yet. θ_D^N is not a unique [79], that is

$$\theta_D^N: \Sigma^* \rightarrow 2^{\Sigma^*}.$$

We will remove superscript N if it is understood: $\theta_D = \theta_D^N$. With both communication delays and losses, the observation mapping is described by the composition of θ_L and θ_D , denoted as θ_{DL} [79]:

$$\theta_{DL} = \theta_L \circ \theta_D.$$

After observing a string t , the set of all possible states that the system may be in is called state estimate after t , which is defined as follows.

$$E(t) = \{q \in Q: (\exists s \in L(G)) t \in \theta_{DL}(s) \wedge \delta(q_0, s) = q\}$$

To obtain state estimate, we do the following. We first construct automaton G_L to describe the communication losses [79]:

$$G_L = LOSS(G) = (Q, \Sigma_0, \delta_{loss}, Q_0),$$

where $\delta_{loss} = \{(q, \sigma, q'): (q, \sigma, q') \in \delta_o\} \cup \{(q, \varepsilon, q'): (q, \sigma, q') \in \delta_{uo} \cup \delta_L\}$.

From G_L , we can build the observer $G_{L,obs}$ as

$$G_{L,obs} = OBS(G_L) = (X, \Sigma_0, \xi, x_0) = Ac(2^Q, \Sigma_0, \xi, UR(\{Q_0\})).$$

where $Ac(\cdot)$ denotes the accessible part, state $x \in X$ is a subset of Q , and $x_0 = UR(\{Q_0\})$ is the unobservable reach of Q_0 , defined as

$$UR(x) = \{q \in Q: (\exists q' \in x) q \in \delta(q', \varepsilon)\}.$$

The transition function is defined as

$$\xi(x, \sigma) = UR(\{q \in Q: (\exists q' \in x) q \in \delta(q', \sigma)\}).$$

Next, we extend each state $x \in X$ to $y = R(x)$. $R(x)$ denotes the set of states that can be reached within N steps in G , that is,

$$R(x) = \{q \in Q: (\exists q' \in x) (\exists s \in \Sigma^*) |s| \leq N \wedge \delta(q', s) = q\}$$

Finally, the networked observer is defined as

$$G_{DL,obs} = DL(OBS(G_L)) = (Y, \Sigma_0, \zeta, y_0).$$

In $G_{DL,obs}$, the state set Y is defined as follows. Denote $X = \{x_1, x_2, x_3, \dots, x_m\}$, then $Y = \{y_1, y_2, y_3, \dots, y_m\}$ with $y_i = R(x_i)$. The transition function $\zeta: Y \times \Sigma_0 \rightarrow Y$ is defined for $y_i, y_j \in Y$ and $\sigma \in \Sigma_0$, as

$$\zeta = \{(y_i, \sigma, y_j): (x_i, \sigma, x_j) \in \xi\}$$

The networked observer can be used to find state estimates. In fact, it is proven in [79] that

$$E(t) = \zeta(y_0, t)$$

As in [19, 20, 73], we accept the following assumption

[1] The networked discrete event system G is deadlock free [73]

$$(\forall q \in Q)(\exists \sigma \in \Sigma)\delta(q, \sigma)!$$

This means that for any state there is at least one event is defined.

[2] No loops in G that contain only unobservable events [73]

$$\neg(\exists q \in Q)(\exists s \in \Sigma_{uo}^*)s \neq \varepsilon \wedge q \in \delta(q, s).$$

Example 2

We consider the discrete event system shown in Figure 4. The system has five states, and four events. $\Sigma_o = \{\alpha, \beta, \gamma, \mu\}$ and $\Sigma_{uo} = \{\tau\}$. We assume that upper bound on communication delays is $N = 1$, and $\delta_L = \{(q_1, \alpha, q_2)\}$.

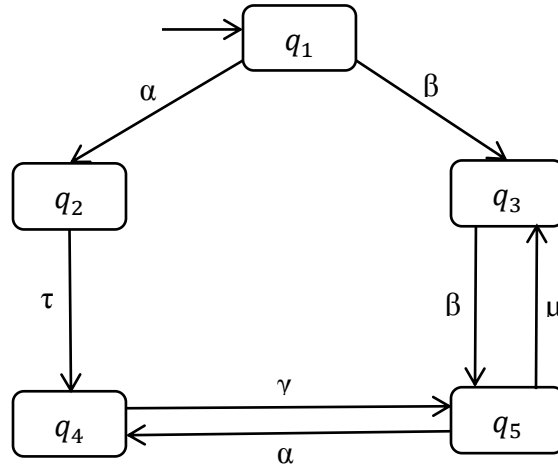


Figure 4. Discrete event system G of Example 2

We first construct $G_L = LOSS(G)$ as shown in Figure 5.

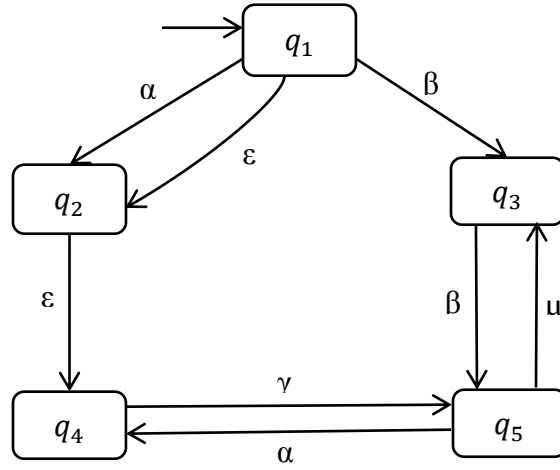


Figure 5. $G_L = \text{LOSS}(G)$ of Example 2

We then construct the observer $G_{L,obs} = \text{OBS}(G_L)$ as shown in Figure 6. In $G_{L,obs}$,

$$X = \{\{q_1, q_2, q_4\}, \{q_2, q_4\}, \{q_3\}, \{q_4\}, \{q_5\}\}.$$

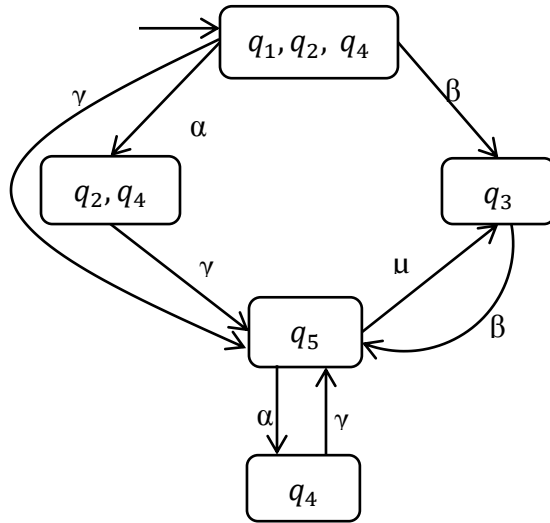


Figure 6. Observer $G_{L,obs} = \text{OBS}(G_L)$ of Example 2

Finally, we construct the networked observer as shown in Figure 7. In $G_{DL,obs}$,

$$Y = \{\{q_1, q_2, q_3, q_4, q_5\}, \{q_2, q_4, q_5\}, \{q_3, q_4, q_5\}, \{q_3, q_5\}, \{q_4, q_5\}\}.$$

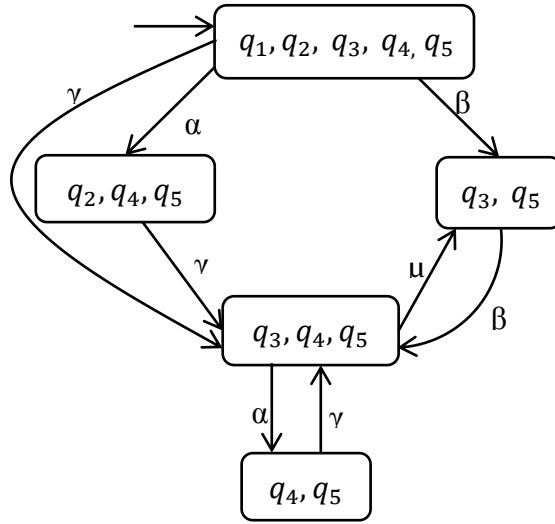


Figure 7. Networked observer
 $G_{DL,obs}=DL(OBS(G_L))$ of Example 2

Form the networked observer, we know that, for example, if $t = \alpha\nu\mu$ is observed, then the state estimate

$$E(t) = \zeta(y_0, t) = \{q_3, q_5\}.$$

3.2 Network Detectability of Discrete Event Systems

Determining the state of a discrete event system is very important, and it is needed in many applications. The importance of the detectability of discrete event systems varies depending on the type of the system. For example, detecting the current state of nuclear reactor is more important than detecting the current state of a printer. The detectability that has been discussed in many papers is non-networked detectability. In other words, there are no delay and loss involved in the discrete event system. In the practical systems, delay or loss of the events or control commands may occur especially when we have networked discrete event systems. Taking the delay and loss in consideration, we need to redefine the four types of detectabilities: detectability, strong detectability, periodic detectability, and strong periodic detectability.

Detectability of discrete event systems was first studied in the mid of 80s in [14, 91]. In these papers, problems like current state and initial state estimation have been introduced and studied. In the 1990, the stability current state detectability was studied by [15]. Many papers have been published after that in [19, 20, 73, 79], and various estimation problems have been discussed. For example, the four types of detectability, generalized detectability, D-detectability, I-detectability are investigated in these published papers.

The delay and loss in the events or control commands happen when we have a networked discrete event system because of the real-time network used to connect the entire system nodes. There are many advantages of using the networked control systems, such as reducing the complexity of the system, increasing the simplicity of the system by making it easy to add/remove nodes, and simplifying the test/diagnose of the system. However, networked discrete event systems introduce delay and loss, so modifications are made to redefine the detectability in networked discrete event systems.

The applications usually define what type of detectability we need to use. For example, defining the current state and subsequent states of the system is required in applications like monitoring the nuclear reactor's state, so that we prevent the reactor to access to undesired or unwanted state.

In this section, we define and investigate detectability of networked discrete event systems, called network detectability. Depending on the requirements of applications, we consider four types of network detectabilities: strong network detectability, strong periodic network detectability, (weak) periodic network detectability, and (weak) network detectability.

We will use the following notations in our work: The set of all possible infinite strings/trajectories of G is denoted by $L^\omega(G)$ [19, 102]. For a string $s \in L^\omega(G)$, we denote the set

of all its prefixes by $Pr(s)$. Also, for any finite string w , we use $|w|$ to denote the length of this string. For any set X , we use $|X|$ to denote the number of elements in X (cardinality).

3.2.1 Definitions of Network Detectabilities

We now define network detectabilities as follows.

Definition 1 (Strong Network Detectability)

A networked nondeterministic discrete event system G is strongly network detectable with respect to θ_{DL} if we can decide, after a limited number of observations, the present state and succeeding states of the system for all trajectories of the system, that is,

$$(\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow |E(t')| = 1).$$

Definition 2 (Network Detectability)

A networked nondeterministic discrete event system G is network detectable with respect to θ_{DL} if we can decide, after a limited number of observations, the present state and succeeding states of the system for some trajectories of the system, that is,

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow |E(t')| = 1).$$

Definition 3 (Strong Periodic Network Detectability)

A networked nondeterministic discrete event system G is strongly periodically network detectable with respect to θ_{DL} if we can periodically decide the present state of the system for all trajectories of the system. That is,

$$(\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in \theta_{DL}(s))(\forall t' \in Pr(t))(\exists t'' \in \Sigma_o^*) \\ (t't'' \in Pr(t) \wedge |t''| < n \wedge |E(t't'')| = 1).$$

Definition 4 (Periodic Network Detectability)

A networked nondeterministic discrete event system G is periodically network detectable with respect to θ_{DL} if we can periodically decide the present state of the system for some trajectories of the system. That is,

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(\exists t'' \in \Sigma_0^*) \\ (t't'' \in Pr(t) \wedge |t''| < n \wedge |E(t't'')| = 1).$$

The strong network detectability is the strong version of network detectability, where the present state and succeeding states can always be determined after a limited number of observations for *all* trajectories of the system. The network detectability, on the other hand, requires the present state and succeeding states to be determined for *some* trajectories of the system after a limited number of observations. The application itself determines the type of detectability needed.

3.2.2 Checking Network Detectabilities

In order to check network detectabilities, we first construct networked observer $G_{DL,obs}$. We then mark the states in $G_{DL,obs}$ that contain a singleton state and denote the set by:

$$Y_m = \{y \in Y : |y| = 1\}.$$

The state in G is known when $G_{DL,obs}$ is in Y_m . Let us denote the set of all loops in $G_{DL,obs}$ as

$$Loop = \{(y, u) \in Y \times \Sigma_0^* : |u| \geq 1 \wedge \zeta(y, u) = y\}$$

Theorem 1

A networked nondeterministic discrete event system G is strongly network detectable with respect to θ_{DL} if and only if in the networked observer $G_{DL,obs}$,

$$(\forall (y, u) \in Loop)(\forall w \in \Sigma_0^*) \zeta(y, w) \in Y_m.$$

In other words, any state reachable from any loop in $G_{DL,obs}$ is in Y_m .

Proof

Note that

$$\begin{aligned} & (\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow |E(t')| = 1) \\ \Leftrightarrow & (\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t' \in Pr(\theta_{DL}(s)))(|t'| > n \Rightarrow |E(t')| = 1) \end{aligned}$$

We first prove the “if” part by showing that if G is not strongly network detectable, then $(\forall (y, u) \in Loop)(\forall w \in \Sigma_0^*) \zeta(y, w) \in Y_m$ is not true.

Suppose that the networked discrete event system G is not strongly detectable with respect to θ_{DL} , then:

$$(\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t' \in Pr(\theta_{DL}(s)))(|t'| > n \wedge |E(t')| \neq 1).$$

Let n be sufficiently large, then, the string t' must go through at least one loop in the networked observer $G_{DL,obs}$. Define first loop by $(y, u) \in Loop$. Clearly, t' will pass y first, that is, $(\exists w \in \Sigma_0^*)(\exists v \in \Sigma_0^*) t' = vw \wedge \zeta(y_0, v) = y$. For such t' , we have $\zeta(y_0, t') = \zeta(y_0, vw) = \zeta(y, w)$. Moreover, $|E(t)| \neq 1 \Rightarrow |\zeta(y, w)| \neq 1 \Rightarrow \zeta(y, w) \notin Y_m$. Hence,

$$(\exists (y, u) \in Loop)(\exists w \in \Sigma_0^*) \zeta(y, w) \notin Y_m,$$

that is, $(\forall (y, u) \in Loop)(\forall w \in \Sigma_0^*) \zeta(y, w) \in Y_m$ is not true.

We next prove the “only if” part by showing that if $(\forall (y, u) \in Loop)(\forall w \in \Sigma_0^*) \zeta(y, w) \in Y_m$ is not true, then G is not strongly network detectable.

Assume $(\forall (y, u) \in Loop)(\forall w \in \Sigma_0^*) \zeta(y, w) \in Y_m$ is not true, that is,

$$(\exists (y, u) \in Loop)(\exists w \in \Sigma_0^*) \zeta(y, w) \notin Y_m.$$

Let v to be any string heading to y from the initial state, that is, $\zeta(y_0, v) = y$. For any $n \in \mathbb{N}$, there exist $s \in \theta_{DL}^{-1}(vu^n w \dots) \cap L^\omega(G)$ and $t' = vu^n w \in Pr(\theta_{DL}(s))$ such that $\zeta(y_0, t') = \zeta(y_0, vu^n w) = \zeta(y, u^n w) = \zeta(y, w) \notin Y_m$. Hence,

$$\begin{aligned} & (\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t' \in Pr(\theta_{DL}(s)))(|t'| > n \wedge \zeta(y_0, t') \notin Y_m) \\ \Rightarrow & (\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t' \in Pr(\theta_{DL}(s)))(|t'| > n \wedge |\zeta(y_0, t')| \neq 1) \end{aligned}$$

$$\Rightarrow (\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t' \in Pr(\theta_{DL}(s)))(|t'| > n \wedge |E(t')| \neq 1)$$

Therefore, the G is not strongly detectable with respect to θ_{DL} . ■

Theorem 2

A networked nondeterministic discrete event system G is network detectable with respect to θ_{DL} if and only if in the networked observer $G_{DL,obs}$,

$$(\exists(y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \in Y_m.$$

In other words, there are loops in $G_{DL,obs}$ which are completely inside Y_m .

Proof

We first prove “if” part by showing that if $(\exists(y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \in Y_m$ is true, then G is network detectable.

Assume that $(\exists(y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \in Y_m$ is true. Let v to be any string heading to y from the initial state, that is, $\zeta(y_0, v) = y$. For such v , there exists $s \in \theta_{DL}^{-1}(vuuu\dots) \cap L^\omega(G)$, $t = vuuu\dots \in \theta_{DL}(s)$, and $n = |v| \in \mathbb{N}$ such that for all $t' \in Pr(t)$, $|t'| > n \Rightarrow t' = vu^jw$, for some $j \in \mathbb{N}$ and $w \in Pr(u)$. Hence, $E(t') = \zeta(y_0, t') = \zeta(y_0, vu^jw) = \zeta(y, u^jw) = \zeta(y, w) \in Y_m$. Therefore,

$$\begin{aligned} & (\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow E(t') \in Y_m \\ & \Rightarrow (\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow |E(t')| = 1). \end{aligned}$$

That is, G is network detectable with respect to θ_{DL} .

We next prove the “only if” part by showing if G is detectable with respect to θ_{DL} , then $(\exists(y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \in Y_m$ is true.

Suppose that G is detectable with respect to θ_{DL} , that is,

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow |E(t')| = 1).$$

Then such t must go through at least one loop in $G_{DL,obs}$. Denote a loop after n transitions by $(y, u) \in Loop$. Let v to the prefix of t that leads to y , that is, $\zeta(y_0, v) = y$. Since $|E(t')| = 1 \Rightarrow |\zeta(y_0, t')| = 1 \Rightarrow \zeta(y_0, t') \in Y_m$, all states in the loop are in Y_m . In other words,

$$(\exists(y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \in Y_m.$$

■

Theorem 3

A networked nondeterministic discrete event system G is strongly periodically network detectable with respect to θ_{DL} if and only if in the networked observer $G_{DL,obs}$,

$$(\forall(y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_m,$$

that is, every loop in $G_{DL,obs}$ must contain at least one state belonging to Y_m .

Proof

We first need to prove the “if” part by showing that if G is not strongly periodically detectable, then $(\forall(y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_m$ is not true.

Suppose that the networked discrete event system G is not strongly periodically network detectable with respect to θ_{DL} , then,

$$\begin{aligned} &(\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\exists t' \in Pr(t))(\forall t'' \in \Sigma_o^*) \\ &(t' t'' \in Pr(t) \wedge |t''| < n \Rightarrow |E(t')| \neq 1). \end{aligned}$$

Take $n = |Y| + 1$. By the above equation,

$$\begin{aligned} &(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\exists t' \in Pr(t))(\forall t'' \in \Sigma_o^*) \\ &(t' t'' \in Pr(t) \wedge |t''| < |Y| + 1 \Rightarrow |E(t')| \neq 1). \end{aligned}$$

Consider the next $n = |Y| + 1$ states after t' in $G_{DL,obs}$ on the path of t'' , since $|E(t')| \neq 1$ all these states do not belong to Y_m . Since the path of t'' is greater than $|Y|$, it must contain a loop.

Denote this loop by $(y, u) \in Loop$. Since all states visited by (y, u) do not belong to Y_m ,

$$(\exists(y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \notin Y_m.$$

That is,

$$(\forall(y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_m$$

is not true.

Next, we prove the “only if” part by showing that if $(\forall(y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_m$ is not true, then G is not strongly periodically network detectable.

Suppose that $(\forall(y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_m$ is not true, that is,

$$(\exists(y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \notin Y_m.$$

Let v to be any string heading to y from the initial state, that is, $\zeta(y_0, v) = y$. In this case, for all $n \in \mathbb{N}$, there exists $s \in \theta_{DL}^{-1}(vuuu\dots) \cap L^\omega(G)$, $t = vuuu\dots \in \theta_{DL}(s)$, and $t' = v$ such that we can let t'' to travel the loop (y, u) sufficient number of times so that the following is true

$$\begin{aligned} & (\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\exists t' \in Pr(t))(\forall t'' \in \Sigma_o^*) \\ & (t't'' \in Pr(t) \wedge |t''| < n \Rightarrow \zeta(y_0, t't'') = \zeta(y, t'') \notin Y_m) \end{aligned}$$

Which implies

$$\begin{aligned} & (\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\exists t' \in Pr(t))(\forall t'' \in \Sigma_o^*) \\ & (t't'' \in Pr(t) \wedge |t''| < n \Rightarrow |E(t')| \neq 1). \end{aligned}$$

In other words, G is not strongly periodically network detectable with respect to θ_{DL} . ■

Theorem 4

A networked nondeterministic discrete event system G is periodically network detectable with respect to θ_{DL} if and only if in the networked observer $G_{DL,obs}$,

$$(\exists(y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_m,$$

That is, there are loops in $G_{DL,obs}$ that include at least one state belonging to Y_m .

Proof

We first prove “only if” part. Suppose that G is periodically detectable with reference to θ_{DL} , that is,

$$\begin{aligned} & (\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(\exists t'' \in \Sigma_o^*) \\ & (t't'' \in Pr(t) \wedge |t''| < n \wedge |E(t')| = 1). \end{aligned}$$

Then t must go through a loop in $G_{DL,obs}$ in which $|E(t')| = 1$ is true for some t'' .

Designate this loop by $(y, u) \in Loop$, then

$$(\exists t'' \in \Sigma_o^*)SP(E(t't'')) \cap T_{spec} = \emptyset \Rightarrow (\exists w \in Pr(u)) \zeta(y, w) \in Y_m.$$

Therefore, $(\exists (y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_m$.

We next prove the “if” part. Suppose that $(\exists (y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_D$ is true. Let v to be any string heading to y from the initial state, that is, $\zeta(y_0, v) = y$. In this case, there exists $n = |vu| \in \mathbb{N}$, $s \in \theta_{DL}^{-1}(vuuu\dots) \cap L^\omega(G)$, and $t = vuuu\dots \in \theta_{DL}(s)$ such that

$$\begin{aligned} & (\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(\exists t'' \in \Sigma_o^*) \\ & (t't'' \in Pr(t) \wedge |t''| < n \wedge \zeta(y_0, t't'') \in Y_m), \end{aligned}$$

where $t't'' = vu^jw$ for some $j \in \mathbb{N}$. Therefore,

$$\begin{aligned} & (\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(\exists t'' \in \Sigma_o^*) \\ & (t't'' \in Pr(t) \wedge |t''| < n \wedge |E(t')| = 1). \end{aligned}$$

In other words, G is periodically network detectable with respect to θ_{DL} . ■

3.2.3 An Algorithm to Check Network Detectabilities of Discrete Event Systems

In summary, we can check network detectabilities using the following algorithm.

Algorithm 1

Input: A networked nondeterministic discrete event system G

An observation mapping θ_{DL} with delays bounded by N .

Output: Network detectable (= yes or no)

Strongly network detectable (= yes or no)

Periodically network detectable (= yes or no)

Strongly periodically network detectable (= yes or no)

Step 1: $G_L = LOSS(G)$;

Step 2: $G_{L,obs} = OBS(G_L)$;

Step 3: $G_{DL,obs} = DL(OBS(G_L))$;

Step 4: $Y_m = \{y \in Y : |y| = 1\}$;

Step 5: $Loop = \{(y, u) \in Y \times \Sigma_0^* : |u| \geq 1 \wedge \zeta(y, u) = y\}$;

Step 6: If $(\forall (y, u) \in Loop)(\forall w \in \Sigma_0^*) \zeta(y, w) \in Y_m$ is true, then

Strongly network detectable = yes;

else

Strongly network detectable = no;

Step 7: If $(\exists (y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \in Y_m$ is true, then

Network detectable = yes;

else

Network detectable = no.

Step 8: If $(\forall (y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_m$ is true, then

Strongly periodically network detectable = yes;

else

Strongly periodically network detectable = no;

Step 9: If $(\exists(y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_m$ is true, then

Periodically network detectable = yes;

else

Periodically network detectable = no.

Because of uncertainties in communication delays and losses, it is rather difficult to determine the state of a system for certain. Therefore, it is rather difficult for a system to be strongly network detectable or network detectable. It is not difficult to see that a loop in $G_{DL,obs}$ is completely inside Y_m only if it is a self loop. For this reason, in networked discrete event systems, strong detectability is equivalent to strong periodic detectability and detectability is equivalent to periodic detectability. The following example illustrates various types of detectability of networked discrete event systems.

3.3 An Illustrative Example

Example 3

To illustrate results for strong network detectability and network detectability, let us first consider the networked discrete event systems shown in Figure 8. Let $\Sigma_o = \{\alpha, \beta, \mu\}$ and $\Sigma_{uo} = \{\lambda\}$. We assume that upper bound on communication delays is $N = 1$, and $\delta_L = \emptyset$

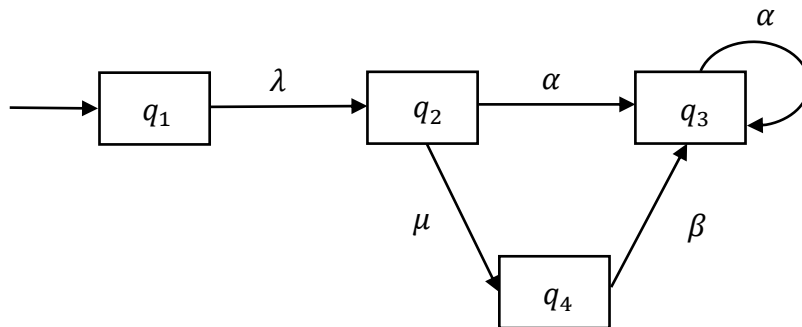


Figure 8. The discrete event system G of Example 3

We construct G_L , observer $G_{L,obs}$, and the networked observer $G_{DL,obs}$ as in Figure 9, 10, and 11 respectively. By definition, $Y_m = \{q_3\}$. There is only one loop in Figure 10. Clearly, condition $(\forall (y, u) \in Loop)(\forall w \in \Sigma_0^*)\zeta(y, w) \in Y_m$ is satisfied. Therefore, the system G shown in Figure 8 is strongly network detectable and hence also strong periodic network detectable.

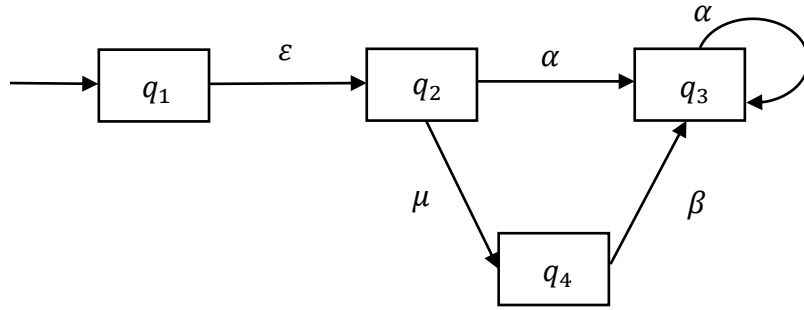


Figure 9. $G_L = LOSS(G)$ of Example 3.

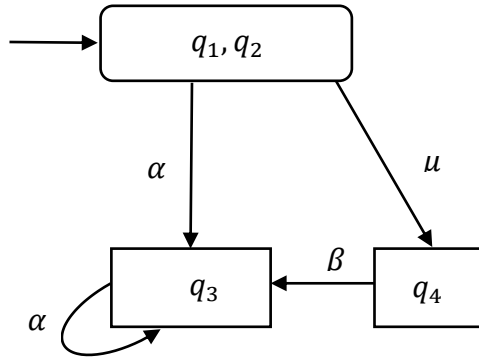


Figure 10. Observer $G_{L,obs}$ of the system in Example 3

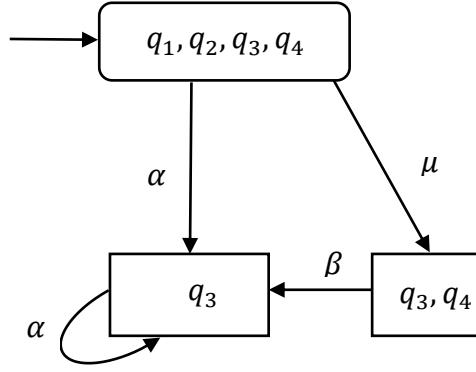


Figure 11. Networked observer $G_{DL,obs}$ of the system in Example 3

Let us now modify the system as in Figure 12. Let $\Sigma_o = \{\alpha, \beta, \tau, \mu\}$ and $\Sigma_{uo} = \{\lambda\}$. Assume that upper bound on delay is $N = 1$, and $\delta_L = \{(q_4, \beta, q_3)\}$.

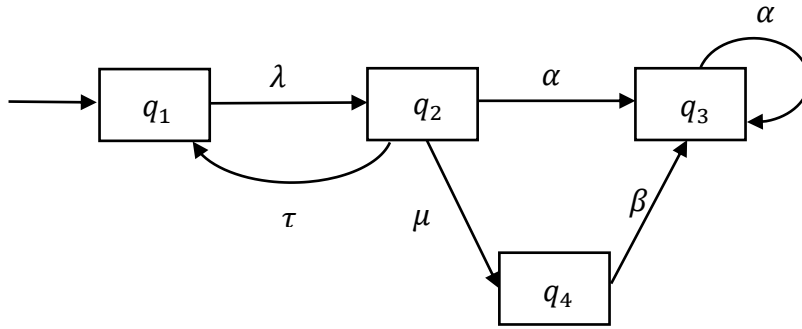


Figure 12. The modified discrete event system G' of Example 3

We construct the observer $G_{L,obs}$ as shown in Figure 13 and the networked observer $G_{DL,obs}$ as shown in Figure 14. By the definition, $Y_m = \{q_3\}$. There are two loops in $G_{DL,obs}$. It is clear that the condition $(\forall (y, u) \in Loop)(\forall w \in \Sigma_o^*) \zeta(y, w) \in Y_m$ is not satisfied. Hence the modified discrete event system G' is not strongly network detectable. However, it is network detectable because the condition $(\exists (y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \in Y_m$ is satisfied.

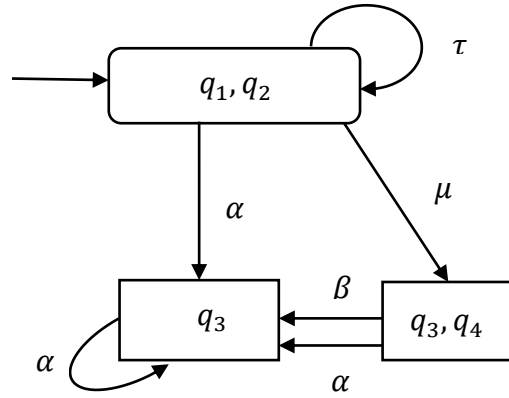


Figure 13. Observer $G'_{L,obs}$ of the modified system in Example 3

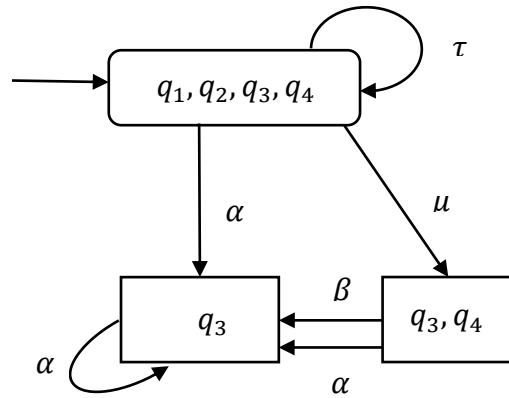


Figure 14. Networked observer $G'_{DL,obs}$ of the modified system in Example 3

CHAPTER 4 D-DETECTABILITY OF NETWORKED DISCRETE EVENT SYSTEMS

Due to uncertainty in communication delays and losses, it is hard to identify the state of a networked discrete event system exactly. Hence, it is more likely that we will use network D-detectability in practice. Network D-detectability can be defined as the ability to distinguish certain pairs of states instead of identifying the current state. To this end, we define the set of all state pairs as

$$T = \{(q, q') : q \in Q \wedge q' \in Q\}.$$

We specify the set of state pairs to be distinguished as a subset of T , that is,

$$T_{spec} \subseteq T.$$

T_{spec} is called a specification. The requirement of network D-detectability is that any state pair in the specification T_{spec} needs to be distinguished after a finite number of observations. D-detectability can be used to define stability [105-107] of discrete event systems by choosing $T_{spec} = (Q - Q_s) \times Q$, where Q_s is the set of stable states [19].

If the state estimate is a subset $Q' \subseteq Q$, the set of indistinguishable state pairs is defined as:

$$SP(Q') = \{(q, q') : q \in Q' \wedge q' \in Q'\}^c.$$

The set of indistinguishable state pairs after observing $s \in \Sigma^*$ is given by:

$$SP(E(\theta_{DL}(s))) = \{(q, q') : q \in E(\theta_{DL}(s)) \wedge q' \in E(\theta_{DL}(s))\}.$$

The following are the definitions of network D-detectabilities in terms of T_{spec} and $SP(E(\theta_{DL}(s)))$.

4.1 Definitions of Networked D-detectabilities.

We now define network D-detectabilities as follows.

Definition 5 (Strong Network D-detectability)

A networked nondeterministic discrete event system G is said to be strongly network D-detectable with respect to θ_{DL} and T_{spec} if all state pairs in T_{spec} are distinguishable all the time, after a finite number of observations, for all trajectories of the system. Formally,

$$(\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow SP(E(t')) \cap T_{spec} = \emptyset).$$

Definition 6 (Network D-detectability)

A networked nondeterministic discrete event system G is said to be network D-detectable with respect to θ_{DL} and T_{spec} if all state pairs in T_{spec} are distinguishable all the time, after a finite number of observations, for some trajectories of the system. Formally,

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow SP(E(t')) \cap T_{spec} = \emptyset).$$

Definition 7 (Strong Periodic Network D-Detectability)

A networked nondeterministic discrete event system G is said to be strongly periodically network D-detectable with respect to θ_{DL} and T_{spec} if all state pairs in T_{spec} are periodically distinguishable for all trajectories of the system. Formally,

$$(\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in \theta_{DL}(s))(\forall t' \in Pr(t))(\exists t'' \in \Sigma_o^*) \\ (t't'' \in Pr(t) \wedge |t''| < n \wedge SP(E(t't'')) \cap T_{spec} = \emptyset).$$

Definition 8 (Periodic Network D-Detectability)

A networked nondeterministic discrete event system G is said to be periodically network D-detectable with respect to θ_{DL} and T_{spec} if all state pairs in T_{spec} are periodically distinguishable for some trajectories of the system. That is,

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(\exists t'' \in \Sigma_o^*) \\ (t't'' \in Pr(t) \wedge |t''| < n \wedge SP(E(t't'')) \cap T_{spec} = \emptyset).$$

4.2 Checking Network D-detectabilities

To check network D-detectabilities, we construct networked observer $G_{DL,obs}$ and mark the states as follows:

$$Y_D = \{y \in Y : SP(y) \cap T_{spec} = \emptyset\}.$$

Theorem 5

A networked nondeterministic discrete event system G is strongly network D-detectable with respect to θ_{DL} and T_{spec} if and only if in the networked observer $G_{DL,obs}$,

$$(\forall (y, u) \in Loop)(\forall w \in \Sigma_0^*) \zeta(y, w) \in Y_D,$$

that is, any state reachable from any loop in $G_{DL,obs}$ is in Y_D .

Proof

Note that

$$\begin{aligned} & (\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow SP(E(t')) \cap T_{spec} = \emptyset) \\ \Leftrightarrow & (\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t' \in Pr(\theta_{DL}(s)))(|t'| > n \Rightarrow SP(E(t')) \cap T_{spec} = \emptyset) \end{aligned}$$

We first prove the “if” part by showing that if G is not strongly D-detectable, then $(\forall (y, u) \in Loop)(\forall w \in \Sigma_0^*) \zeta(y, w) \in Y_D$ is not true.

Suppose that the networked discrete event system G is not strongly D-detectable with respect to θ_{DL} and T_{spec} , then:

$$(\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t' \in Pr(\theta_{DL}(s)))(|t'| > n \wedge SP(E(t')) \cap T_{spec} \neq \emptyset)$$

Let n be sufficiently large. Then, the string t' must go through at least one loop in the networked observer $G_{DL,obs}$. Define first loop by $(y, u) \in Loop$. Clearly, t' will pass y first, that is, $(\exists v \in \Sigma_0^*)(\exists w \in \Sigma_0^*) t' = vw \wedge \zeta(y_0, v) = y$. For such t' , we have $\zeta(y_0, t') = \zeta(y_0, vw) = \zeta(y, w)$. Moreover, $SP(E(t')) \cap T_{spec} \neq \emptyset \Rightarrow \zeta(y, w) \notin Y_D$. Hence,

$$(\exists (y, u) \in Loop)(\exists w \in \Sigma_0^*) \zeta(y, w) \notin Y_D.$$

that is, $(\forall(y, u) \in Loop)(\forall w \in \Sigma_0^*) \zeta(y, w) \in Y_D$ is not true.

Now, let us proof the “only if” part by showing that if $(\forall(y, u) \in Loop)(\forall w \in \Sigma_0^*) \zeta(y, w) \in Y_D$ is not true, then G is not strongly D-detectable.

Assume $(\forall(y, u) \in Loop)(\forall w \in \Sigma_0^*) \zeta(y, w) \in Y_D$ is not true, that is,

$$(\exists(y, u) \in Loop)(\exists w \in \Sigma_0^*) \zeta(y, w) \notin Y_D.$$

Let v to be any string heading to y from the initial state, that is, $\zeta(y_0, v) = y$. For any $n \in \mathbb{N}$, there exists $s \in \theta_{DL}^{-1}(vu^n w \dots) \cap L^\omega(G)$ and $t' = vu^n w \in Pr(\theta_{DL}(s))$ such that $\zeta(y_0, t') = \zeta(y_0, vu^n w) = \zeta(y, u^n w) = \zeta(y, w) \notin Y_D$. Hence,

$$\begin{aligned} & (\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t' \in Pr(\theta_{DL}(s)))(|t'| > n \wedge \zeta(y_0, t') \notin Y_D) \\ \Rightarrow & (\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t' \in Pr(\theta_{DL}(s)))(|t'| > n \wedge SP(\zeta(y_0, t')) \cap T_{spec} \neq \emptyset) \\ \Rightarrow & (\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t' \in Pr(\theta_{DL}(s)))(|t'| > n \wedge SP(E(t')) \cap T_{spec} \neq \emptyset) \end{aligned}$$

therefore, the G is not strongly D-detectable with respect to θ_{DL} and T_{spec} .

■

Theorem 6

A networked nondeterministic discrete event system G is network D-detectable with respect to θ_{DL} and T_{spec} if and only if in the networked observer $G_{DL,obs}$,

$$(\exists(y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \in Y_D,$$

that is, there are loops in $G_{DL,obs}$ which are completely inside Y_D .

Proof

We first prove “if” part by showing that if $(\exists(y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \in Y_D$ is true, then G is network D-detectable.

Assume that $(\exists(y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \in Y_D$ is true. Let v to be any string heading to y from the initial state, that is, $\zeta(y_0, v) = y$. For such v , there exists $s \in$

$\theta_{DL}^{-1}(vuuu\dots) \cap L^\omega(G)$, $t = vuuu\dots \in \theta_{DL}(s)$, and $n = |v| \in \mathbb{N}$ such that for all $t' \in Pr(t)$, $|t'| > n \Rightarrow t' = vu^jw$, for some $j \in \mathbb{N}$ and $w \in Pr(u)$. Hence, $\zeta(y_0, t') = \zeta(y_0, vu^jw) = \zeta(y, u^jw) = \zeta(y, w) \in Y_D$. Therefore,

$$\begin{aligned} & (\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow \zeta(y, w) \in Y_D) \\ \Rightarrow & (\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow SP(\zeta(y, w)) \cap T_{spec} = \emptyset) \\ \Rightarrow & (\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow SP(E(t')) \cap T_{spec} = \emptyset) \end{aligned}$$

Hence, G is network D-detectable with respect to θ_{DL} .

We next prove the “only if” part by showing if G is D-detectable with respect to θ_{DL} , then $(\exists(y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \in Y_D$ is true.

Suppose that G is D-detectable with respect to θ_{DL} , that is,

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow SP(E(t')) \cap T_{spec} = \emptyset)$$

Then such t must go through at least one loop in $G_{DL,obs}$. Denote a loop after n transitions by $(y, u) \in Loop$. Let v to the prefix of t that leads to y , that is, $\zeta(y_0, v) = y$. Since $SP(E(t')) \cap T_{spec} = \emptyset \Rightarrow SP(\zeta(y_0, t')) \cap T_{spec} = \emptyset \Rightarrow \zeta(y_0, t') \in Y_D$, all states in the loop are in Y_D . In other words,

$$(\exists(y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \in Y_D \text{ is true.}$$

■

Theorem 7

A networked nondeterministic discrete event system G is strongly periodically network D-detectable with respect to θ_{DL} and T_{spec} if and only if in the networked observer $G_{DL,obs}$,

$$(\forall(y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_D,$$

that is, every loop in $G_{DL,obs}$ must contain at least one state belonging to Y_D .

Proof

We first need to prove the “if” part by showing that if G is not strongly periodically D-detectable, then $(\forall (y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_D$ is not true.

Suppose that the networked discrete event system G is not strongly periodically network D-detectable with respect to θ_{DL} and T_{spec} , then,

$$(\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\exists t' \in Pr(t))(\forall t'' \in \Sigma_o^*) \\ (t't'' \in Pr(t) \wedge |t''| < n \Rightarrow SP(E(t't'')) \cap T_{spec} \neq \emptyset).$$

Take $n = |Y| + 1$. By the above equation,

$$(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\exists t' \in Pr(t))(\forall t'' \in \Sigma_o^*) \\ (t't'' \in Pr(t) \wedge |t''| < |Y| + 1 \Rightarrow SP(E(t't'')) \cap T_{spec} \neq \emptyset).$$

Consider the next $n = |Y| + 1$ states after t' in $G_{DL,obs}$ on the path of t'' , since $SP(E(t't'')) \cap T_{spec} \neq \emptyset$, all these states do not belong to Y_D . Since the path of t'' is greater than $|Y|$, it must contain a loop. Denote this loop by $(y, u) \in Loop$. Since all states visited by (y, u) do not belong to Y_D ,

$$(\exists (y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \notin Y_D.$$

That is,

$$(\forall (y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_D$$

is not true.

Next, we prove the “only if” part by showing that if $(\forall (y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_D$ is not true, then G is not strongly periodically network D-detectable.

Suppose that $(\forall (y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_D$ is not true, that is,

$$(\exists (y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \notin Y_D.$$

Let v to be any string heading to y from the initial state, that is, $\zeta(y_0, v) = y$. In this case, for all $n \in \mathbb{N}$, there exists $s \in \theta_{DL}^{-1}(vuuu\dots) \cap L^\omega(G)$, $t = vuuu\dots \in \theta_{DL}(s)$, and $t' = v$ such that we can let t'' to travel the loop (y, u) sufficient number of times so that the following is true

$$(\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\exists t' \in Pr(t))(\forall t'' \in \Sigma_o^*) \\ (t't'' \in Pr(t) \wedge |t''| < n \Rightarrow \zeta(y_0, t't'') = \zeta(y, t'') \notin Y_D)$$

Which implies

$$(\forall n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\exists t' \in Pr(t))(\forall t'' \in \Sigma_o^*) \\ (t't'' \in Pr(t) \wedge |t''| < n \Rightarrow SP(E(t't'')) \cap T_{spec} \neq \emptyset).$$

In other words, G is not strongly periodically network D-detectable with respect to θ_{DL} and T_{spec} .

■

Theorem 8

A networked nondeterministic discrete event system G is periodically network D-detectable with respect to θ_{DL} and T_{spec} if and only if in the networked observer $G_{DL,obs}$,

$$(\exists(y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_D,$$

that is, there are loops in $G_{DL,obs}$ that include no less than one state belonging to Y_D .

Proof

We first prove “only if” part. Suppose that G is periodically D-detectable with reference to θ_{DL} and T_{spec} , that is,

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(\exists t'' \in \Sigma_o^*) \\ (t't'' \in Pr(t) \wedge |t''| < n \wedge SP(E(t't'')) \cap T_{spec} = \emptyset).$$

Then t must go through a loop in $G_{DL,obs}$ in which $SP(E(t't'')) \cap T_{spec} = \emptyset$ is true for some t'' . Designate this loop by $(y, u) \in Loop$, then

$$(\exists t'' \in \Sigma_o^*) SP(E(t't'')) \cap T_{spec} = \emptyset \Rightarrow (\exists w \in Pr(u)) \zeta(y, w) \in Y_D.$$

Therefore, $(\exists (y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_D$.

We next prove the “if” part. Suppose that $(\exists (y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_D$ is true. Let v to be any string heading to y from the initial state, that is, $\zeta(y_0, v) = y$. In this case, there exists $n = |vu| \in \mathbb{N}$, $s \in \theta_{DL}^{-1}(vuuu\dots) \cap L^\omega(G)$, and $t = vuuu\dots \in \theta_{DL}(s)$ such that

$$\begin{aligned} & (\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(\exists t'' \in \Sigma_o^*) \\ & (t't'' \in Pr(t) \wedge |t''| < n \wedge \zeta(y_0, t't'') \in Y_D), \end{aligned}$$

where $t't'' = vu^j w$ for some $j \in \mathbb{N}$. Therefore,

$$\begin{aligned} & (\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(\exists t'' \in \Sigma_o^*) \\ & (t't'' \in Pr(t) \wedge |t''| < n \wedge SP(E(t't'')) \cap T_{spec} = \emptyset). \end{aligned}$$

In other words, G is periodically network D-detectable with respect to θ_{DL} and T_{spec} . ■

4.3 An Algorithm to Check Network D-detectabilities of Discrete Event Systems

In summary, we can check network D-detectabilities using the following algorithm.

Algorithm 2

Input: A networked nondeterministic discrete event system G

An observation mapping θ_{DL} with delays bounded by N .

A specification $T_{spec} \subseteq T$.

Output: Network D-detectable (= yes or no)

Strongly network D-detectable (= yes or no)

Periodically network D-detectable (= yes or no)

Strongly periodically network D-detectable (= yes or no)

Step 1: $G_L = LOSS(G)$;

Step 2: $G_{L,obs} = OBS(G_L)$;

Step 3: $G_{DL,obs} = DL(OBS(G_L))$;

Step 4: $Y_D = \{y \in Y : SP(y) \cap T_{spec} = \emptyset\}$;

Step 5: $Loop = \{(y, u) \in Y \times \Sigma_0^* : |u| \geq 1 \wedge \zeta(y, u) = y\}$;

Step 6: If $(\forall (y, u) \in Loop)(\forall w \in \Sigma_0^*) \zeta(y, w) \in Y_D$ is true, then

Strongly network D-detectable = yes;

else

Strongly network D-detectable = no;

Step 7: If $(\exists (y, u) \in Loop)(\forall w \in Pr(u)) \zeta(y, w) \in Y_D$ is true, then

Network D-detectable = yes;

else

Network D-detectable = no.

Step 8: If $(\forall (y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_D$ is true, then

Strongly periodically network D-detectable = yes;

else

Strongly periodically network D-detectable = no;

Step 9: If $(\exists (y, u) \in Loop)(\exists w \in Pr(u)) \zeta(y, w) \in Y_D$ is true, then

Periodically network D-detectable = yes;

else

Periodically network D-detectable = no.

4.4 Illustrative Examples

Example 4

Let us consider the discrete event system shown in Figure 15. The system represents a nuclear reactor with six states, and each state represents the speed of nuclear reaction from very slow (q_1) to very high (q_6). The transition from state to state requires an event α or β to happen. Physically, α and β represent the removing and inserting of the control rods of the reactor to a known position. Assume that $\Sigma_o = \{\alpha_1, \alpha_2, \alpha_4, \alpha_5, \beta_1, \beta_2, \beta_4, \beta_5, \beta_6\}$ and $\Sigma_{uo} = \{\alpha_3, \beta_3\}$ because of sensor failure. Also, assume that upper bound on communication delays is $N = 1$, and $\delta_L = \{(q_3, \beta_2, q_2), (q_2, \alpha_2, q_3)\}$. The specification is given by

$$T_{spec} = \{(q_1, q_5), (q_1, q_6), (q_2, q_5), (q_2, q_6)\}$$

The specification above used to distinguish the safe states $\{q_1, q_2\}$ from the dangerous states $\{q_5, q_6\}$.

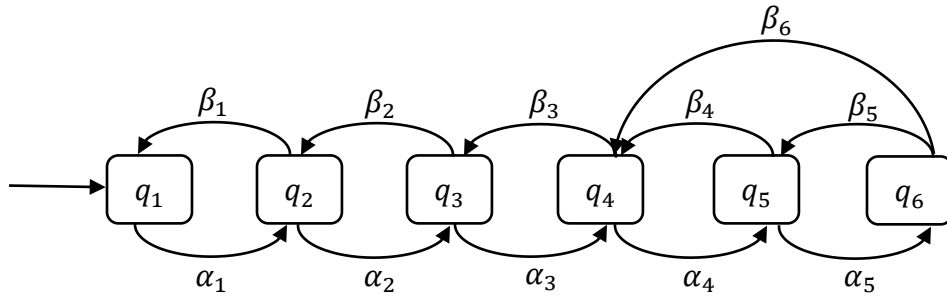


Figure 15. A discrete event system G representing a nuclear reactor

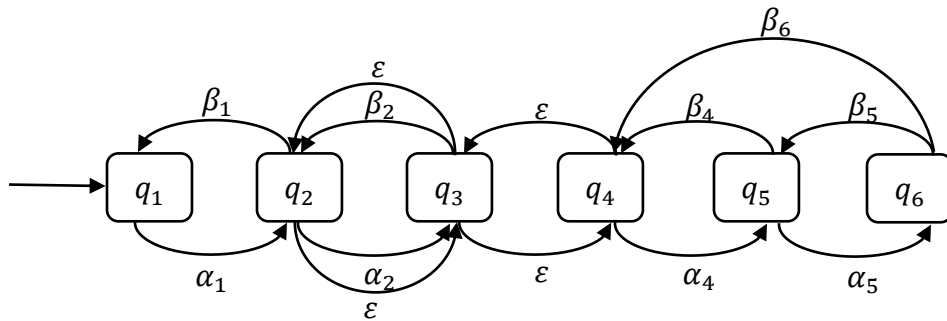


Figure 16. Automaton G_L of the system in Example 4

Using Algorithm 2, we calculate G_L , $G_{L,obs}$, and networked observer $G_{DL,obs}$ as shown in Figures 16, 17, and 18 respectively.

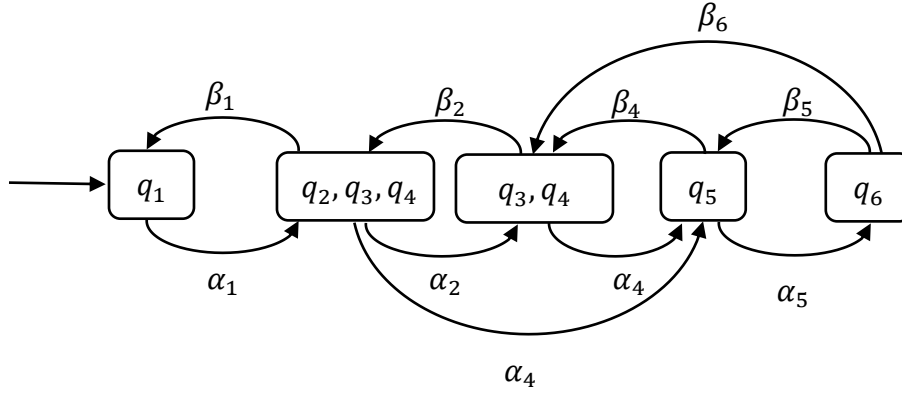


Figure 17. Observer $G_{L,obs}$ of the system in Example 4

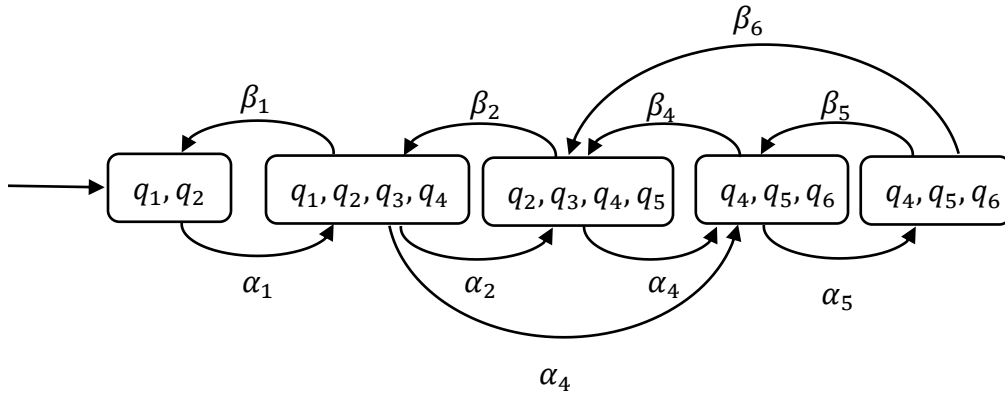


Figure 18. Networked observer $G_{DL,obs}$ of the system in Example 4

The marked states are calculated as

$$Y_D = \{\{q_1, q_2\}, \{q_1, q_2, q_3, q_4\}, \{q_4, q_5, q_6\}, \{q_4, q_5, q_6\}\}$$

By Algorithm 2, we conclude:

Strongly network D-detectable = no;

Network D-detectable = yes;

Strongly periodically network D-detectable = no;

Periodically network D-detectable = yes;

Example 5

Assume that we have a factory with six machines, and this factory is connected to a smart grid to provide it with necessary power to operate. We assume that α_j is the power request event when a machine is switching on with j denoting the total power requested (and not released) so far, β_j is a machine switching off and power release event, μ is a machine break down event, τ is a machine repairing event, and λ is a machine ready to use event (after repair). This factory can be modeled as a discrete event system as in Figure 19.

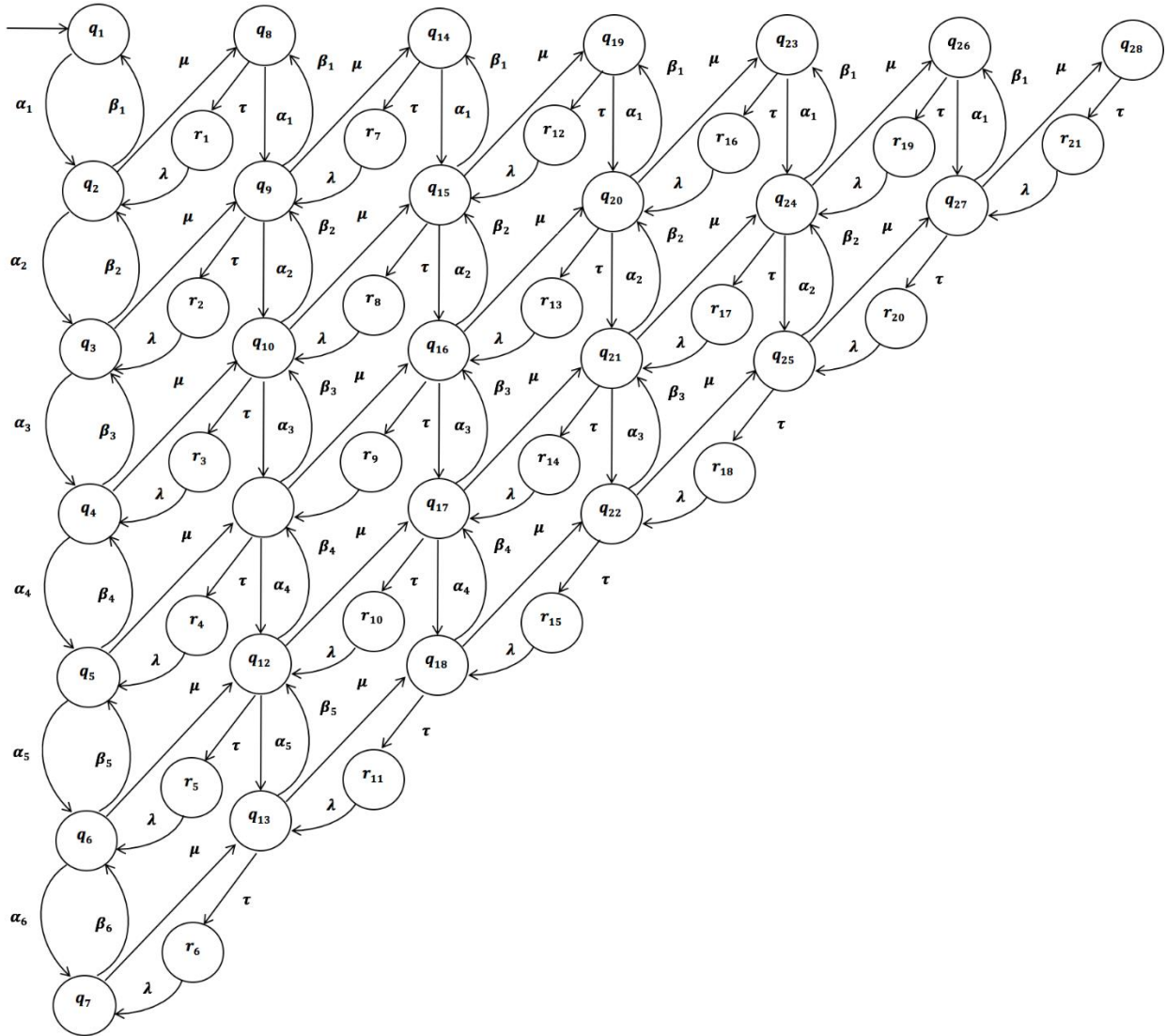


Figure 19. Discrete event system G of the factory

Each q_i state represents a unique state of the system; for example, q_1 represents the state of the system where there are no machines working, and q_5 represents the state of the system where there are four machines working. Similarly, r_i states are transient states where some machines have just been repaired. For example, r_3 is the state of the system where there are one machine working and two being repaired.

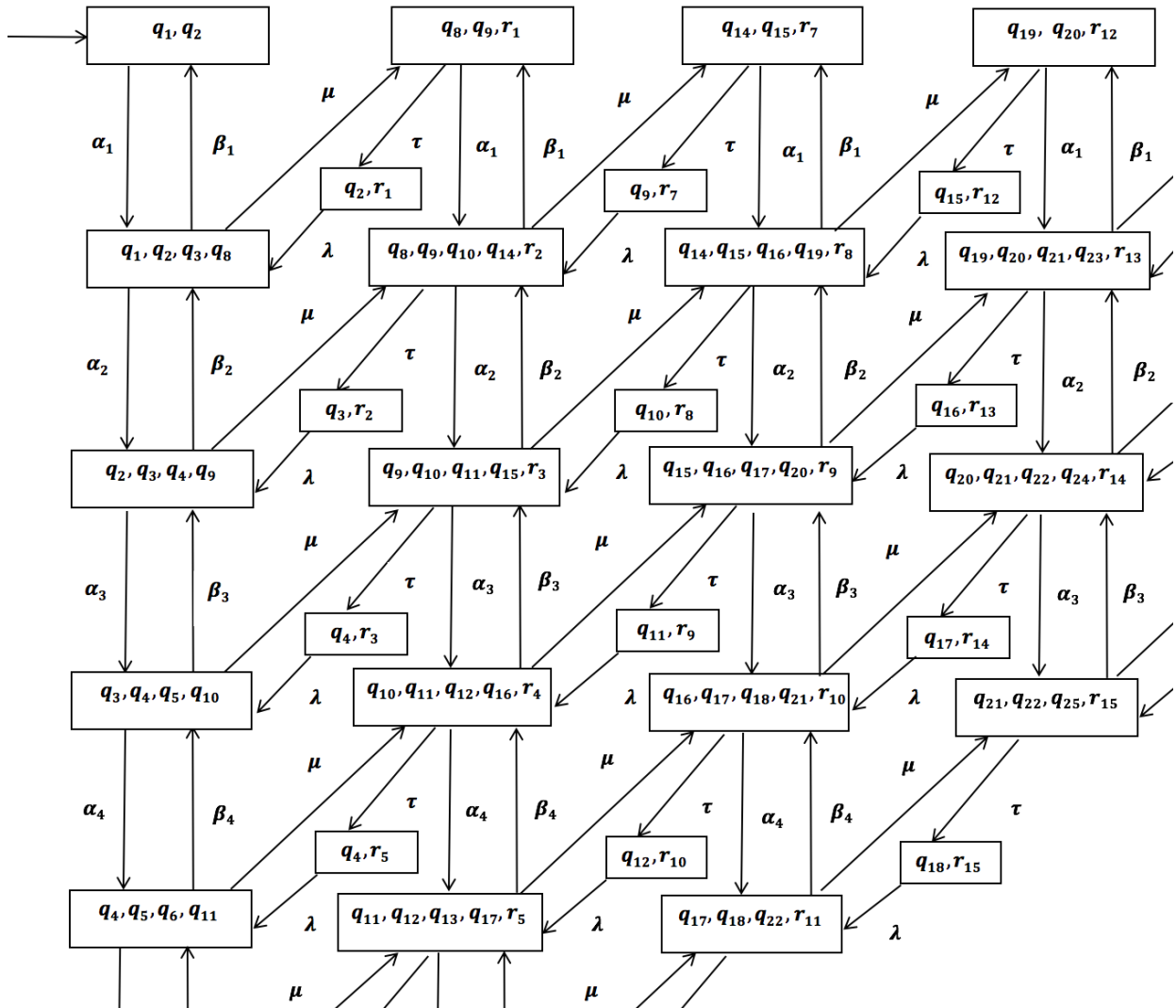


Figure 20. Networked observer $G_{DL,obs}$, for the first few states, with $N = 1$

We assume there is no loss in events, i.e., only the delay can affect the system. Also, assume that the upper bound on observation delays $N = 1$. We construct $G_{DL,obs}$ observer as shown in Figure 20. Note that we only show some states in $G_{DL,obs}$ in order to make it easy to read.

To better schedule the power production, the power supplier would like to know how much power the factory is consuming (which is roughly proportional to the number of machines working). Hence, we group different sets of states as follows:

$Q^0 = \{q_1, q_8, q_{14}, q_{19}, q_{23}, q_{26}, q_{28}, r_1, r_7, r_{12}, r_{16}, r_{19}\}$; no machines is working.

$Q^1 = \{q_2, q_9, q_{15}, q_{20}, q_{24}, q_{27}, r_2, r_8, r_{13}, r_{17}, r_{20}\}$; one machine is working.

$Q^2 = \{q_3, q_{10}, q_{16}, q_{21}, q_{25}, r_3, r_9, r_{14}, r_{18}\}$; two machines are working.

$Q^3 = \{q_4, q_{11}, q_{17}, q_{22}, r_4, r_{10}, r_{15}\}$; three Machines are working.

$Q^4 = \{q_5, q_{12}, q_{18}, r_5, r_{11}\}$; four Machines are working.

$Q^5 = \{q_6, q_{13}, r_6\}$; five machines are working.

$Q^6 = \{q_7\}$; six machines are working.

Due to communications delays, the power supplier may not know the exact power consumption of the factory. However, it would like at least to distinguish states in Q^j from state in Q^{j+3} . Hence, the specification is given by

$$T_{spec} = \{(Q^0 \times Q^3) \cup (Q^0 \times Q^4) \cup (Q^0 \times Q^5) \cup (Q^0 \times Q^6) \cup (Q^1 \times Q^4) \cup (Q^1 \times Q^5) \cup (Q^1 \times Q^6) \cup (Q^2 \times Q^5) \cup (Q^2 \times Q^6) \cup (Q^3 \times Q^6)\}$$

For $N = 1$, we can successfully distinguish all the state pairs specified by T_{spec} . Therefore, the networked discrete event system is strongly network D-detectable. Is this still possible if $N = 2$? Let us construct $G_{DL,obs}$ for $N = 2$ as in Figure 21. We notice that we can distinguish some but not all the state pairs specified by T_{spec} . Therefore, the discrete event system is only network D-detectable.

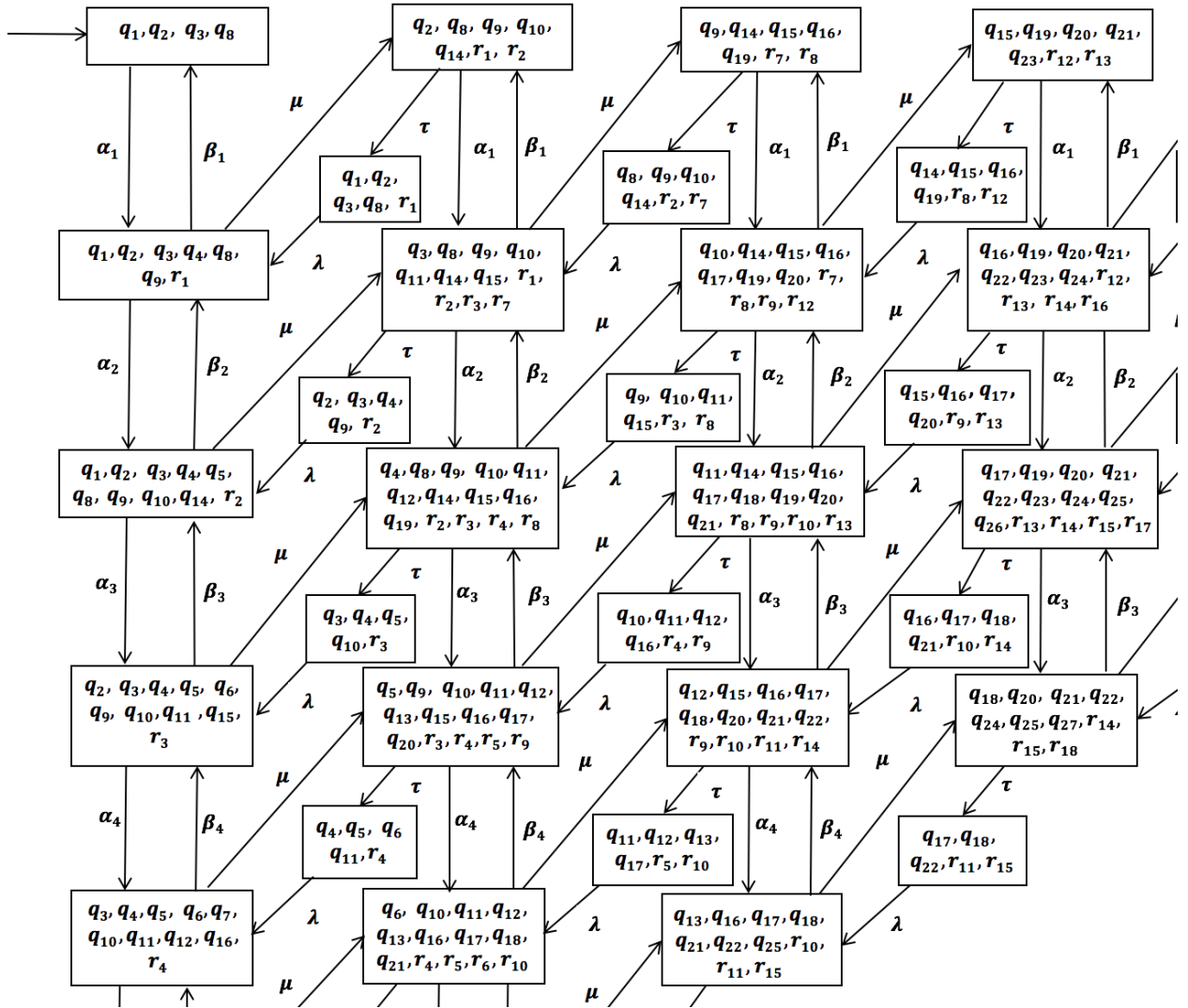


Figure 21. Networked observer $G_{DL,obs}$, for the first few states, with $N = 2$

CHAPTER 5 PROPERTIES OF NETWORK DETECTABILITY

Intuitively, we see that if the upper bound on communication delays N increases, then there are more uncertainties in the state estimate (that is, the state estimate is larger). Let us formally prove this as follows. We use $G_{DL,obs}^N$ to denote the network observer with observation delays bounded by N . Note that communication losses are specified by transitions in δ_L and hence are independent of N .

5.1 Properties

Lemma 1

For a networked discrete event system G , let

$$G_{DL,obs}^N = (Y^N, \Sigma_0, \zeta, y_0^N)$$

where $Y^N = \{y_1^N, y_2^N, \dots, y_m^N\}$ and

$$G_{DL,obs}^K = (Y^K, \Sigma_0, \zeta, y_0^K)$$

where $Y^K = \{y_1^K, y_2^K, \dots, y_m^K\}$. If $N \leq K$, then for $i = 1, 2, \dots, m$,

$$y_i^N \subseteq y_i^K.$$

Proof:

By definition,

$$y_i^N = R^N(x_i) = \{q \in Q : (\exists q' \in x)(\exists s \in \Sigma^*) |s| \leq N \wedge \delta(q', s) = q\}$$

Also

$$y_i^K = R^K(x_i) = \{q \in Q : (\exists q' \in x)(\exists s \in \Sigma^*) |s| \leq K \wedge \delta(q', s) = q\}$$

If $K \geq N$, then $|s| \leq N \Rightarrow |s| \leq K$. Hence

$$(\exists q' \in x)(\exists s \in \Sigma^*) |s| \leq N \wedge \delta(q', s) = q \Rightarrow (\exists q' \in x)(\exists s \in \Sigma^*) |s| \leq K \wedge \delta(q', s) = q$$

Therefore,

$$y_i^N \subseteq y_i^K$$

■

Proposition 1

For a networked discrete event system G , the network observer is same for all $N \geq |Q| - 1$, that is, for $N \geq |Q| - 1$

$$G_{DL,obs}^N = G_{DL,obs}^{|Q|-1},$$

Proof

Denote

$$G_{DL,obs}^N = (Y^N, \Sigma_0, \zeta, y_0^N)$$

where $Y^N = \{y_1^N, y_2^N, \dots, y_m^N\}$ and

$$G_{DL,obs}^{|Q|-1} = (Y^{|Q|-1}, \Sigma_0, \zeta, y_0^{|Q|-1})$$

where $Y^{|Q|-1} = \{y_1^{|Q|-1}, y_2^{|Q|-1}, \dots, y_m^{|Q|-1}\}$. We need to prove that, for $N \geq |Q| - 1$,

$$y_i^N = y_i^{|Q|-1}.$$

Since the total number of states in G is $|Q|$, if a state can be reached from a state in x_i , it can be reached in $|Q| - 1$ steps. Therefore, for $N \geq |Q| - 1$, $y_i^N = R^N(x_i) = R^{|Q|-1}(x_i) = y_i^{|Q|-1}$.

■

Proposition 2

If a network discrete event system G is network detectable when the observation delays are bounded by N , then G is network detectable when the observation delays are bounded by i , for all $i \leq N$.

Proof

By Lemma 1,

if $N \geq i$, then for $j = 1, 2, \dots, m$,

$$y_j^i \subseteq y_j^N$$

This means that y_j^i is a subset of y_j^N , that is, the cardinality $|y_j^N|$ is greater or equal to $|y_j^i|$.

Therefore, if the $G_{DL,obs}^N$ is network detectable, $G_{DL,obs}^i$ is for sure network detectable. ■

The same is true for strong network detectability, network D-detectability, strong network D-detectability, periodic network D-detectability, and strong periodic network D-detectability.

We say that (the graph of) a network discrete event system G is strongly connected if any state in G can be reached from any other state in G . In other words, for any pair of states q_1, q_2 , there exists a path from q_1 to q_2 :

$$(\forall q_1, q_2 \in Q)(\exists s \in \Sigma^*)q_2 \in \delta(q_1, s)$$

Proposition 3

If a network discrete event system G is strongly connected and the bound on communication delays $N \geq 1$, then G is not network detectable. If G is strongly connected and the bound on communication delays $N \geq |Q| - 1$, then G is not network D-detectable.

Proof

In this case, $y = R^N(x) = Q$. In other words, $|y| = |Q| > 1$. The network discrete event system is not detectable. ■

The same is true for strong network detectability, network D-detectability, strong network D-detectability, periodic network D-detectability, and strong periodic network D-detectability.

The following result is for strong network detectability only

Proposition 4

Assume that the upper bound on observation delays $N \geq 1$. Then a networked nondeterministic discrete event system G is strongly network detectable with respect to Θ_{DL} only if all loops in G are self-loops, each containing only one state and there are no transitions leaving the state.

Proof:

Suppose that there exist a loop that contains more than one state or contains one state but has a transition leaving the state. Denote the state in the loop by $q_1 \in Q$. Then

$$(\exists q_1 \in Q)(\exists \sigma \in \Sigma)q_2 \in \delta(q_1, \sigma) \wedge q_2 \neq q_1$$

Denote the corresponding loop in $G_{DL,obs}$ as (y, u) . Since $N \geq 1$, $q_2, q_1 \in y$. In other words, $y \notin Y_m$. Therefore,

$$(\exists (y, u) \in Loop)(\exists \varepsilon \in \Sigma_o^*)\zeta(y, \varepsilon) = y \notin Y_m.$$

By Theorem 1, G is not strongly network detectable with respect to Θ_{DL} . ■

Let us illustrate the above results by following examples.

5.2 Illustrative Examples

Example 6

Let us consider the same system as in Example 1, where $\Sigma_o = \{\alpha, \beta, \gamma, \mu\}$, $\Sigma_{uo} = \{\tau\}$, and $\delta_L = \{(q_1, \alpha, q_2)\}$. In Example 1, we assume that upper bound on communication delays is $N = 1$. Let us now increase the upper bound to $N = 2, 3, 4, \dots$. We note that the networked observer $G_{DL,obs}^N$ is same for all $N \geq 2$, which is shown in Figure 22.

Comparing $G_{DL,obs}^1$ in Figure 7 and $G_{DL,obs}^2$ in Figure 22, it is clear that the conclusion of Lemma 1 holds, that is, for all $i = 1, 2, 3, 4, 5$,

$$y_i^1 \subseteq y_i^2$$

Also, the conclusion of Proposition 1 holds: For all $N \geq |Q| - 1 = 5 - 1 = 4$

$$G_{DL,obs}^N = G_{DL,obs}^4$$

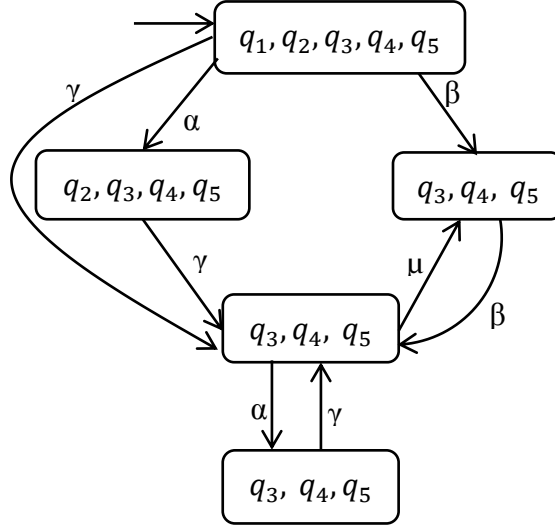


Figure 22. Networked observer $G_{DL,obs}^N$ for $N \geq 2$ of Example 6

The networked discrete event system in Example 5 is not strongly connected. So, let us illustrate Proposition 3 using the following example.

Example 7

Let us consider the networked discrete event system shown in Figure 23, where $\Sigma_o = \{\alpha, \beta, \gamma, \mu\}$, $\Sigma_{uo} = \emptyset$, and $\delta_L = \emptyset$. We construct networked observers $G_{DL,obs}^N$ for $N = 1, 2$ as shown in Figure 19. By Proposition 1, for all $N \geq |Q| - 1 = 3 - 1 = 2$

$$G_{DL,obs}^N = G_{DL,obs}^2.$$

G is not network detectable (strongly network detectable) for $N \geq 2$. This is obvious from Figure 24.

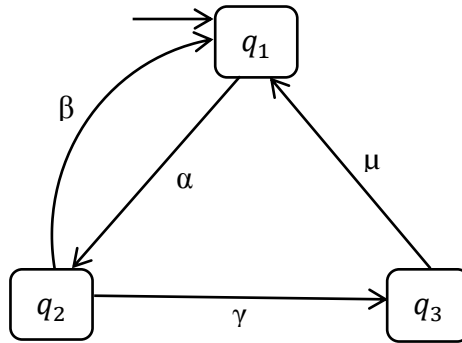


Figure 23. Networked discrete event system G of Example 7

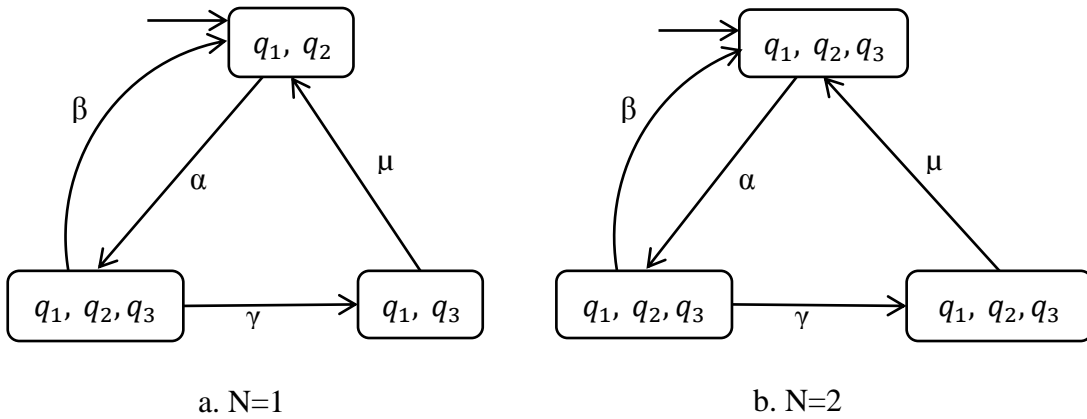


Figure 24. Networked observers $G_{DL,obs}^N$ for $N=1,2$ of Example 7

CHAPTER 6 I-DETECTABILITY OF NETWORKED DISCRETE EVENT SYSTEMS

In our pervious chapters, we talked about state estimation of the current state of the system. current state estimation is important for some types of the system, such as determining the current state of a train. However, in some cases, it is important to know the initial state of the system instead of current state. Initial state detectability, or simply I-detectability, is very important when we need to determine the state of the system after a failure. When a failure happens in a system, it is required to know the initial state of the system that led to that failure to be able the repair it. Like in debugging a software, the programmer needs to know the last instruction executed before the results are not as expected. I-detectability is required whenever we need to determine the state of the system in some past time. checking I-detectability of a system is not always possible because it depends on the system itself. For some systems, I-detectability can be checked for all trajectories of the system, and this called strong I-detectable. However, in other cases, we can only determine I-detectability for some trajectories of the system, and we call it weak I-detectability. Formally, I-detectability is defined as the ability to detect the initial state of the system after finite number of event observations. In [73], Shu and Lin, has investigated I-detectability of discrete event systems. They formally defined weak and strong I-detectabilities as following:

Weak I-Detectability [73]

A nondeterministic discrete event system G is weakly I-detectable with respect to P if we can determine, after a finite event observation, the initial state of the system for some trajectories of the system. That is

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\forall u \in \text{Pr}(s)) |P(u)| > n \Rightarrow |I(Q_0, P(u))| = 1.$$

Strong I-Detectability [73]

A nondeterministic discrete event system G is strongly I-detectable with respect to P if we can determine, after a finite event observation, the initial state of the system for all trajectories of the system. That is

$$(\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall u \in \text{Pr}(s))|P(u)| > n \Rightarrow |I(Q_0, P(u))| = 1.$$

In current state estimation problem, we only deal with the current state of the system, so we only need to remember the present state. however, Initial state estimation is much more complex problem where we need to recall both the current state of the system and some history of the system. The delay and loss in observations will also add more complexity for the system since we deal with uncertainties. In this chapter, we will investigate the effect of delay and loss of observation due to communication. We will extend the I-detectability for discrete event systems to network I-detectability. To investigate network I-detectability, we will expand the discrete event system as shown below.

6.1 Mathematical Background and Network I-observer

We start by using automaton to describe a nondeterministic discrete event system,

$$G = (Q, \Sigma, \delta, Q_0)$$

The initial state estimate after observing $t \in \theta_{DL}(L(G))$ for networked discrete-event system is given by

$$I(Q_0, t) = \{q \in Q_0 : (\exists s \in \Sigma^*) \theta_{DL}(s) = t \wedge \delta(q, s)!\}$$

We define augmented discrete event system G as following:

$$Q^{aug} = (q_C, q_I) \in Q \times Q_0$$

where q_C represents the current state of the system and q_I represents the initial state of the system. observing an event $\alpha \in \Sigma_o$ will only change the current state (q_C) of the system. the initial state

(q_I) will remain unchanged. Hence, the transition function $\delta^{aug}: Q^{aug} \times \Sigma \rightarrow 2^{Q^{aug}}$ of the augmented system will then become

$$\delta^{aug}((q_C, q_I), \alpha) = \{(q'_C, q_I): q'_C \in \delta(q_C, \alpha)\}$$

The above transition function can be extended to string $s \in \Sigma^*$ instead of just event as

$$\delta^{aug}((q_C, q_I), s) = \{(q'_C, q_I): q'_C \in \delta(q_C, s)\}.$$

Note that we use $\delta^{aug}((q_C, q_I), s)!$, which mean that the transition function is defined. that is $\delta^{aug}((q_C, q_I), s) \neq \emptyset$.

At the beginning, the initial state and the current state of the system are equal. That is

$$Q_0^{aug} = \{(q, q): q \in Q_0\}$$

The augmented system will then become

$$G^{aug} = (Q^{aug}, \Sigma, \delta^{aug}, Q_0^{aug}) = Ac(Q \times Q_0, \Sigma, \delta^{aug}, Q_0^{aug}).$$

In the same way, we need to construct augmented automaton G_L^{aug} to describe the communication losses in augmented system:

$$G_L^{aug} = LOSS(G^{aug}) = (Q^{aug}, \Sigma_0, \delta_{loss}^{aug}, Q_0^{aug}),$$

where $\delta_{loss}^{aug} = \{((q_C, q_I), \sigma, (q'_C, q_I)): ((q_C, q_I), \sigma, (q'_C, q_I)) \in \delta_o\} \cup$

$\{((q_C, q_I), \varepsilon, (q'_C, q_I)): ((q_C, q_I), \sigma, (q'_C, q_I)) \in \delta_{uo} \cup \delta_L\}$.

From G_L^{aug} , we can build the augmented observer $G_{L,obs}^{aug}$ as

$$G_{L,obs}^{aug} = OBS(G_L^{aug}) = (X^{aug}, \Sigma_0, \xi^{aug}, x_0^{aug}) = Ac(2^{Q^{aug}}, \Sigma_0, \xi^{aug}, UR^{aug}(\{Q_0^{aug}\})).$$

where $Ac(\cdot)$ denotes the accessible part, state $x^{aug} \in X^{aug}$ is a subset of Q^{aug} , and $x_0^{aug} =$

$UR^{aug}(\{Q_0^{aug}\})$ is the unobservable reach of Q_0^{aug} , defined as

$$UR^{aug}(x^{aug}) = \{(q_C, q_I) \in Q^{aug}: (\exists (q'_C, q_I) \in x^{aug})(q_C, q_I) \in \delta((q'_C, q_I), \varepsilon)\}.$$

The transition function is defined as

$$\xi^{aug}(x^{aug}, \alpha) = UR^{aug}(\{(q_C, q_I) \in Q^{aug}: (\exists(q'_C, q_I) \in x^{aug})(q_C, q_I) \in \delta((q'_C, q_I), \sigma)\}).$$

Next, we extend each state $x^{aug} \in X^{aug}$ to $y^{aug} = R(x^{aug})$. $R(x^{aug})$ denotes the set of states that can be reached within N steps in G^{aug} , that is,

$$R(x^{aug}) = \{(q_C, q_I) \in Q^{aug}: (\exists(q'_C, q_I) \in x^{aug})(\exists s \in \Sigma^*)|s| \leq N \wedge \delta((q'_C, q_I), s) = (q_C, q_I)\}$$

Finally, the augmented networked I-observer $G_{DL,obs}^{aug}$ is defined as

$$G_{DL,obs}^{aug} = DL(OBS(G_L^{aug})) = (Y^{aug}, \Sigma_0, \zeta^{aug}, y_0^{aug}).$$

In $G_{DL,obs}^{aug}$, the state set Y^{aug} is defined as follows. Denote $X^{aug} = \{x_1^{aug}, x_2^{aug}, x_3^{aug}, \dots, x_m^{aug}\}$,

then $Y^{aug} = \{y_1^{aug}, y_2^{aug}, y_3^{aug}, \dots, y_m^{aug}\}$ with $y_i^{aug} = R(x_i^{aug})$. The transition function

$\zeta^{aug}: Y^{aug} \times \Sigma_0 \rightarrow Y^{aug}$ is defined for $y_i^{aug}, y_j^{aug} \in Y^{aug}$ and $\sigma \in \Sigma_0$, as

$$\zeta^{aug} = \{(y_i^{aug}, \sigma, y_j^{aug}): (x_i^{aug}, \sigma, x_j^{aug}) \in \xi^{aug}\}$$

Theorem 9

The initial state estimation after observing $t \in \theta_{DL}(L(G))$ is given by

$$I(Q_0, t) = \{q_I \in Q_0: (\exists q_C \in Q)(q_I, q_C) \in \zeta^{aug}(y_0^{aug}, t)\}$$

Proof

From the definition of the transition function, for $s \in \Sigma^*$, we have

$$q_C \in \delta(q_I, s) \Leftrightarrow (q_I, q_C) \in \delta^{aug}((q_I, q_I), s)$$

After observing $t \in \theta_{DL}(L(G))$, then

$$\begin{aligned} I(Q_0, t) &= \{q_I \in Q_0: (\exists s \in \Sigma^*)\theta_{DL}(s) = t \wedge \delta(q_I, s)!\} \\ &= \{q_I \in Q_0: (\exists q_C \in Q)(\exists s \in \Sigma^*)\theta_{DL}(s) = t \wedge q_C \in \delta(q_I, s)\} \\ &= \{q_I \in Q_0: (\exists q_C \in Q)(\exists s \in \Sigma^*)\theta_{DL}(s) = t \\ &\quad \wedge (q_I, q_C) \in \delta^{aug}((q_I, q_I), s)\} \\ &= \{q_I \in Q_0: (\exists q_C \in Q)(q_I, q_C) \in \zeta^{aug}(y_0^{aug}, t)\} \end{aligned}$$

6.2 Definitions of Network I-Detectabilities

We now define network I-detectabilities as follows.

Definition 9 (Network Weak I-Detectability)

A networked nondeterministic discrete event system G is weakly I-detectable with respect to θ_{DL} if we can determine, after a finite event observation, the initial state of the system for some trajectories of the system. That is

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow |I(Q_0, t')| = 1).$$

Definition 10 (Network Strong I-Detectability)

A networked nondeterministic discrete event system G is strongly I-detectable with respect to θ_{DL} if we can determine, after a finite event observation, the initial state of the system for all trajectories of the system. That is

$$(\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow |I(Q_0, t')| = 1).$$

6.3 Checking Network I-detectabilities

The I-observability of the networked discrete event system will not be affected by the delays of the event observation. In other words, the effect of delays will just delay the observer from sensing the initial state one step depending on the upper delay bound N . therefore, the I-observer used to detect the initial state in [73] can be used to detect and observe the initial state detectability for networked discrete event systems. Only observation losses can affect I-detectabilities. To check network I-detectability of discrete event systems, we have the following two cases.

6.3.1 Checking Network I-detectabilities with Observation Losses

To check network I-detectabilities when we have observation losses, we need to construct augmented networked I-observer $G_{DL,obs}^{aug}$. We then need to mark the states in $G_{DL,obs}^{aug}$ that have the same initial state:

$$Y_I^{aug} = \{y^{aug} \in Y^{aug} : (\exists q \in Q_0)(\forall (q_c, q_I) \in y^{aug})q_I = q\}.$$

The initial state is known when $G_{DL,obs}^{aug}$ is in Y_I^{aug} . Let us denote the set of all loops in $G_{DL,obs}^{aug}$ as

$$Loop = \{(y^{aug}, u) \in Y^{aug} \times \Sigma_0^* : |u| \geq 1 \wedge \zeta^{aug}(y^{aug}, u) = y^{aug}\}$$

Theorem 10

A networked discrete event system G is strongly network I-detectable with respect to θ_{DL} if and only if all loops in $G_{DL,obs}^{aug}$ are entirely inside Y_I^{aug} .

Proof

Note that when the initial state is determined, it is determined thereafter. Hence, for each loop in $G_{DL,obs}^{aug}$, it is either completely in Y_I^{aug} or completely in $Y^{aug} - Y_I^{aug}$. If all loops in $G_{DL,obs}^{aug}$ are completely inside Y_I^{aug} , then the system will then enter Y_I^{aug} after limited number of observations for all trajectories of the system. Once the system is in Y_I^{aug} , it will stay forever and never leave Y_I^{aug} . Therefore, the system is strongly network I-detectable.

If not all loops in $G_{DL,obs}^{aug}$ are completely inside Y_I^{aug} , then the system may stay forever in $Y^{aug} - Y_I^{aug}$. This means we cannot determine the initial state of the system for some trajectories of the system. Therefore, the system is not strongly network I-detectable.

■

Theorem 11

A networked discrete event system G is weakly network I-detectable with respect to θ_{DL} if and only if $Y_I^{aug} \neq \emptyset$ in $G_{DL,obs}^{aug}$.

Proof

If $Y_I^{aug} = \emptyset$, that is, empty set, then the initial state cannot be determined for all trajectory of the system. Hence, the system is not weakly network I-detectable.

If $Y_I^{aug} \neq \emptyset$, that is, not empty set, then the system will reach Y_I^{aug} for some trajectories of the system and the initial state can be determined. Therefore, the system is weakly network I-detectable. ■

Example 8

Let us consider the networked discrete event system in Figure 25. We assume that the upper delay bound $N = 0$ and all events are observable. Also, we assume that observable events are $\{\alpha, \beta\}$ and $\delta_L = \{(q_2, \mu, q_4)\}$. The system initial state is $Q_0 = \{q_1, q_2\}$. We use the procedure above to construct network I-observer. We augment the networked discrete event system by extending states to state pairs. By following the same procedure, we will get the network I-observer shown in Figure 26

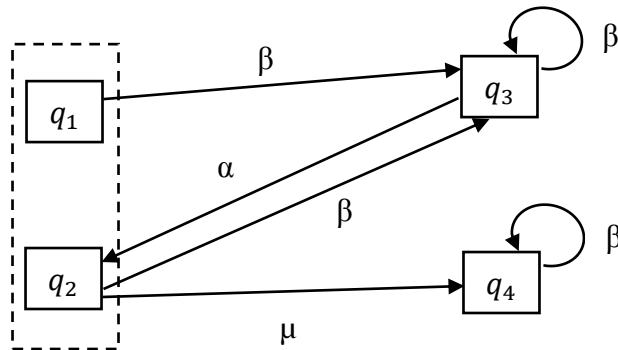


Figure 25. Example 8.

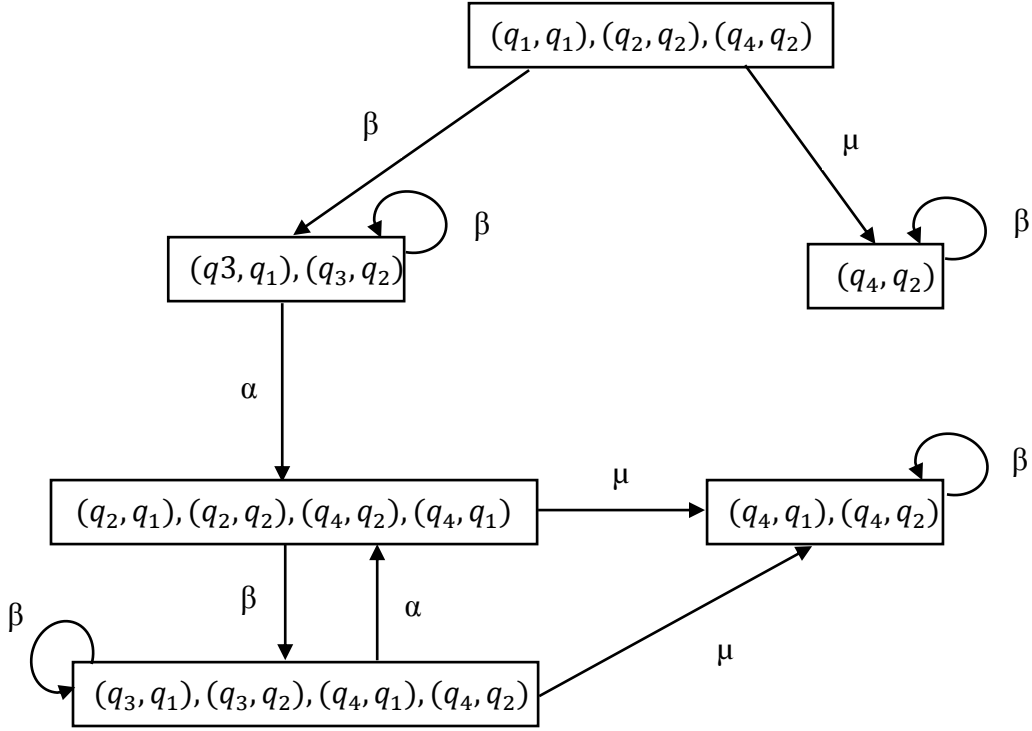


Figure 26. Network I-Observer for Example 8.

6.3.2 Checking Network I-detectabilities without Observation Losses.

The I-detectability of a networked discrete event systems will not be affected by observation delays. In other words, when there are no observation losses, network I-detectabilities are equivalent to I-detectabilities. Hence, the I-observer used for checking I-detectabilities [73] can be used to check network I-detectabilities. This is formally proved as follows.

Theorem 12

Assume that there are no observation losses in the system, that is, $\delta_{loss} = \emptyset$ and $\delta_{loss}^{aug} = \emptyset$.

G is weakly network I-detectable if and only if G is weakly I-detectable.

Proof

To prove that weakly network I-detectability (A) is equivalent to weakly I-detectability (B), we need to prove that $A \Leftrightarrow B$, where, by the definitions, A and B are as follows:

$$A: (\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow |I(Q_0, t')| = 1)$$

$$B: (\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\forall u \in \text{Pr}(s))|P(u)| > n \Rightarrow |I(Q_0, P(u))| = 1$$

$(A \Rightarrow B)$

Assume that A is true, since

$$(\exists t \in \theta_{DL}(u) = \{P(u), P(u_{-1}), P(u_{-2}), \dots, P(u_{-N})\})(|t| > n \Rightarrow |I(Q_0, t)| = 1)$$

In particular, for $t = P(u)$

$|t| > n \Rightarrow |I(Q_0, t)| = 1$, that is,

$$|P(u)| > |I(Q_0, t)| = 1$$

Hence B is true.

$(A \Rightarrow B)$

Assume that B is true, then

$$(\forall u \in \text{Pr}(s))(|P(u)| > n \Rightarrow |I(Q_0, P(u))| = 1)$$

$$(\forall u_{-1} \in \text{Pr}(s))(|P(u_{-1})| > n \Rightarrow |I(Q_0, P(u))| = 1)$$

$$(\forall u_{-2} \in \text{Pr}(s))(|P(u_{-2})| > n \Rightarrow |I(Q_0, P(u))| = 1)$$

$$(\forall u_{-N} \in \text{Pr}(s))(|P(u_{-N})| > n \Rightarrow |I(Q_0, P(u))| = 1)$$

This implies

$$(\forall u \in \text{Pr}(s))(\forall t \in \{P(u), P(u_{-1}), P(u_{-2}), \dots, P(u_{-N})\})(|t| > n \Rightarrow |I(Q_0, t)| = 1)$$

That is, A is true.

If δ_{loss} is equal to \emptyset , then $\theta_{DL}(s) = \{P(s), P(s_{-1}), P(s_{-2}), \dots, P(s_{-N})\}$. This means that the projection will be equal natural. Therefore, A will be a weakly network I-detectable. That proves $(A \Rightarrow B)$.

If $(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\forall u \in \text{Pr}(s))|P(u)| > n \Rightarrow |I(Q_0, P(u))| = 1$ is a weakly I-detectable and if the system introduced to a delay, it will take larger n ($n + N$) for the system to reach $|I(Q_0, P(u))| = 1$.

■

Theorem 13

Assume that there are no observation losses in the system, that is, $\delta_{loss} = \emptyset$ and $\delta_{loss}^{aug} = \emptyset$.

G is strongly network I-detectable if and only if G is strongly I-detectable.

Proof

To prove that strongly network I-detectability (A) is equivalent to strongly I-detectability (B), we need to prove that $A \Leftrightarrow B$, where, by the definitions, A and B are as follows:

$$A: (\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow |I(Q_0, t')| = 1)$$

$$B: (\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall u \in Pr(s))|P(u)| > n \Rightarrow |I(Q_0, P(u))| = 1$$

(A \Rightarrow B)

Assume that A is true, since

$$(\forall t \in \theta_{DL}(u) = \{P(u), P(u_{-1}), P(u_{-2}), \dots, P(u_{-N})\})(|t| > n \Rightarrow |I(Q_0, t)| = 1)$$

In particular, for $t = P(u)$

$|t| > n \Rightarrow |I(Q_0, t)| = 1$, that is,

$$|P(u)| > |I(Q_0, t)| = 1$$

Hence B is true.

(A \Rightarrow B)

Assume that B is true, then

$$(\forall u \in Pr(s))(|P(u)| > n \Rightarrow |I(Q_0, P(u))| = 1)$$

$$(\forall u_{-1} \in Pr(s))(|P(u_{-1})| > n \Rightarrow |I(Q_0, P(u))| = 1)$$

$$(\forall u_{-2} \in Pr(s))(|P(u_{-2})| > n \Rightarrow |I(Q_0, P(u))| = 1)$$

$$(\forall u_{-N} \in Pr(s))(|P(u_{-N})| > n \Rightarrow |I(Q_0, P(u))| = 1)$$

This implies

$$(\forall u \in \text{Pr}(s))(\forall t \in \{P(u), P(u_{-1}), P(u_{-2}), \dots, P(u_{-N})\})(|t| > n \Rightarrow |I(Q_0, t)| = 1)$$

That is, A is true.

If δ_{loss} is equal to \emptyset , then $\theta_{DL}(s) = \{P(s), P(s_{-1}), P(s_{-2}), \dots, P(s_{-N})\}$. This means that the projection will be equal natural. Therefore, A will be a strongly network I-detectable. That proves $(A \Rightarrow B)$.

If $(\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall u \in \text{Pr}(s))|P(u)| > n \Rightarrow |I(Q_0, P(u))| = 1$ is a strongly I-detectable and if the system introduced to a delay, it will take larger n ($n + N$) for the system to reach $|I(Q_0, P(u))| = 1$.

■

Example 9

Let us recall the networked discrete event system in example 8. We assume that the upper delay bound $N = 1$ and all events are observable. We assume that events are observable, and the system initial state $Q_0 = \{q_1, q_2\}$. We construct the network I-observer as in Figure 27. We also construct the I-observer in case of $N = 0$ as in Figure 28

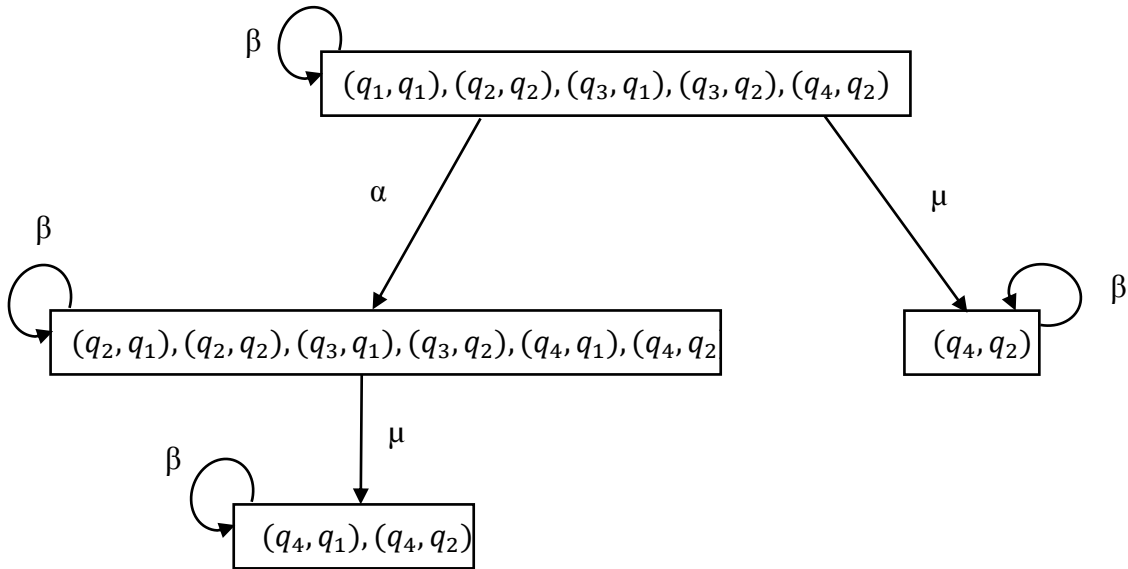


Figure 27. Network I-Observer for Example 9 $N=1$.

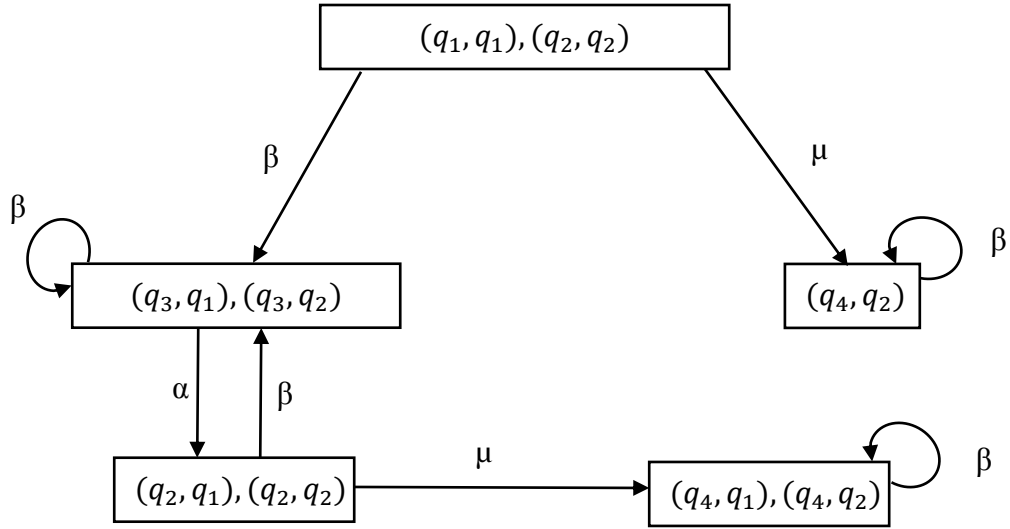


Figure 28. Network I-Observer for Example 9 $N=0$.

Note that in both cases the system is weakly detectable.

6.4 An Algorithm to Check Network I-detectabilities of Discrete Event Systems with Observation Losses.

In summary, if we have a networked discrete event systems with delays only, I-observer introduce in [73] can be used to check network I-detectability. In case of delays and losses of events, we can use the following algorithm to check network I-detectability and strong network I-detectability of networked discrete event systems.

Algorithm 3

Input: A networked nondeterministic discrete event system G

An observation mapping θ_{DL} with delays bounded by N .

Output: Network I-detectable (= yes or no)

Strongly network I-detectable (= yes or no)

Step 1: $G_L^{aug} = LOSS(G^{aug});$

Step 2: $G_{L,obs}^{aug} = OBS(G_L^{aug});$

Step 3: $G_{DL,obs}^{aug} = DL(OBS(G_L^{aug}));$

Step 4: $Y_I^{aug} = \{y^{aug} \in Y^{aug} : (\exists q \in Q_0)(\forall (q_c, q_l) \in y^{aug})q_l = q\}$;

Step 5: $Loop = \{(y^{aug}, u) \in Y^{aug} \times \Sigma_0^* : |u| \geq 1 \wedge \zeta^{aug}(y^{aug}, u) = y^{aug}\}$;

Step 6: If $(\forall (y^{aug}, u) \in Loop)(\forall w \in \Sigma_0^*)\zeta^{aug}(y^{aug}, w) \in Y_I^{aug}$ is true, then

Strongly network I-detectable = yes;

else

Strongly network I-detectable = no;

Step 7: If $(\exists (y^{aug}, u) \in Loop)(\forall w \in Pr(u))\zeta^{aug}(y^{aug}, w) \in Y_I^{aug}$ is true, then

Network I-detectable = yes;

else

Network I-detectable = no.

CHAPTER 7 NETWORK CO-DETECTABILITY

In some applications, the observer is decentralized, so we have a set of local agents each with limited observations. While the agents are not sharing data among themselves, if we can determine the current and subsequent states all trajectories, then we have a system with strong co-detectability. As you may know, every discrete event system is a unique case. Hence, it is hard sometimes to detect the current and subsequent states for some systems for all trajectories of the system. In such case, we may have a discrete event system with a weak co-detectability. In case of multi-agents, determining the current and subsequent states of the system can be achieved with the aid of limited observations of each agent. Co-detectability requires that we can determine the state of the system by at least one agent. Formally, co-detectability can be defined as following:

We assume that I is the index of agents $I = \{1, 2, 3, \dots, n\}$.

P_i is the natural projection for an agent A_i .

$$P_i : \Sigma^* \rightarrow \Sigma_{i,o}^*$$

$$P_i(\sigma) = \begin{cases} \sigma & \text{if } \sigma \in \Sigma_{i,o} \\ \varepsilon & \text{if } \sigma \notin \Sigma_{i,o} \end{cases} \quad P_i(s\sigma) = \begin{cases} P_i(s)\sigma & \text{if } \sigma \in \Sigma_{i,o} \\ P_i(s) & \text{if } \sigma \notin \Sigma_{i,o} \end{cases}$$

and

$R_i(Q', t)$ is the set of all possible states after observing $t \in \Sigma_{i,o}^*$ for an agent A_i .

In this chapter, we investigate network co-detectability of discrete event systems. We assume loss/delay between each agent and system. Therefore, this will introduce uncertainty in detecting the current and subsequent states of the system.

Weak Co-detectability [75]

A discrete event system G is (weakly) co-detectable if, after a finite number of observations, the current state and subsequent states of the system is known to at least one agent for some trajectories of the system. Formally,

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\forall t \in \text{Pr}(s))|P(t)| > n \Rightarrow (\exists i \in I)|R_i(Q_0, P_i((t)))| = 1.$$

Strong Co-detectability [75]

A discrete event system G is strongly co-detectable if, after a finite number of observations, the current state and subsequent states of the system is known to at least one agent for all trajectories of the system. Formally,

$$(\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in \text{Pr}(s))|P(t)| > n \Rightarrow (\exists i \in I)|R_i(Q_0, P_i((t)))| = 1.$$

Weak Periodic Co-detectability [75]

A discrete event system G is (weakly) periodically co-detectable if the current state of the system is known periodically to at least one agent for some trajectories of the system. Formally,

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\forall t \in \text{Pr}(s))(\exists t' \in \Sigma^*)tt' \in \text{Pr}(s) \wedge \\ |P(t')| < n \wedge (\exists i \in I)|R_i(Q_0, P_i((tt')))| = 1.$$

Strong Periodic Co-detectability [75]

A discrete event system G is strongly periodically co-detectable if the current state of the system is known periodically to at least one agent for all trajectories of the system. Formally,

$$(\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in \text{Pr}(s))(\exists t' \in \Sigma^*)tt' \in \text{Pr}(s) \wedge \\ |P(t')| < n \wedge (\exists i \in I)|R_i(Q_0, P_i((tt')))| = 1.$$

7.1 Mathematical Background

The current state of the system will determine by these agents without sharing data. The partial sensing capability for a specific agent A_i is represented as observable events $\Sigma_{i,o}$. Obviously, the observable event of the networked discrete event system will be the union of all observable events of the agents:

$$\Sigma_o = \bigcup_{i \in I} \Sigma_{i,o}$$

We assume each agent A_i has the same upper bound on delay. We denote that as N . This assumption is made for simplicity.

The delay and lose projection for any agent A_i is denoted as

$$\theta_{i,DL} = \theta_{i,L} \circ \theta_{i,D}$$

where

$$\theta_{i,D}: \Sigma^* \rightarrow 2^{\Sigma^*}.$$

and

$$\theta_{i,L}: L(G) \rightarrow 2^{\Sigma_{i,o}^*}$$

We build the network observer for each agent as follows

$$G_{i,L} = LOSS(G) = (Q, \Sigma_{i,o}, \delta_{i,loss}, Q_0),$$

where

$$\delta_{i,loss} = \{(q, \sigma, q') : (q, \sigma, q') \in \delta_{i,o}\} \cup \{(q, \varepsilon, q') : (q, \sigma, q') \in \delta_{i,u0} \cup \delta_{i,L}\}.$$

We used δ to denote the set of all transitions in G : $\delta = \{(q, \sigma, q') : q' \in \delta(q, \sigma)\}$. The set of observable transitions for an agent A_i is denoted by $\delta_{i,o} = \{(q, \sigma, q') \in \delta : \sigma \in \Sigma_{i,o}\}$. The set of unobservable transitions is denoted by $\delta_{i,u0} = \{(q, \sigma, q') \in \delta : \sigma \in \Sigma_{i,u0}\}$. Some observable transitions may be lost in communication. These transitions are denoted by $\delta_{i,L}$ ($\delta_{i,L} \subseteq \delta_{i,o}$) [79, 103].

From $G_{i,L}$, we can build the observer $G_{L,obs}$ as

$$G_{i,L,obs} = OBS(G_{i,L}) = (X_i, \Sigma_{i,o}, \xi_i, x_{i,0}) = Ac(2^Q, \Sigma_{i,o}, \xi_i, UR_i(\{Q_0\})).$$

where $Ac(\cdot)$ denotes the accessible part, state $x_i \in X_i$ is a subset of Q , and $x_{i,0} = UR_i(\{Q_0\})$ is the unobservable reach of Q_0 , defined as

$$UR_i(x) = \{q \in Q : (\exists q' \in x) q \in \delta_{i,loss}(q', \varepsilon)\}.$$

The transition function is defined as

$$\xi_i(x, \sigma) = UR_i(\{q \in Q: (\exists q' \in x)q \in \delta_{i,loss}(q', \sigma)\}).$$

Next, we extend each state $x_i \in X_i$ to $y_i = R(x_i)$. $R(x_i)$ denotes the set of states that can be reached within N steps in G , that is,

$$R(x_i) = \{q \in Q: (\exists q' \in x_i)(\exists s \in \Sigma^*)|s| \leq N \wedge \delta(q', s) = q\}$$

Finally, the networked observer is defined as

$$G_{i,DL,obs} = DL(OBS(G_{i,L})) = (Y_i, \Sigma_{i,o}, \zeta_i, y_{i,0}).$$

In $G_{i,DL,obs}$, the state set Y_i is defined as follows. Denote $X = \{x_1, x_2, x_3, \dots, x_n\}$, then $Y_i = \{y_1, y_2, y_3, \dots, y_n\}$ with $y_j = R(x_j)$. The transition function $\zeta_i: Y_i \times \Sigma_{i,o} \rightarrow Y_i$ is defined for $y_j, y_k \in Y_i$ and $\sigma \in \Sigma_{i,o}$, as

$$\zeta_i = \{(y_j, \sigma, y_k): (x_j, \sigma, x_k) \in \xi_i\}$$

The networked observer can be used to find state estimates. In fact, it is proven by Lin (2014) that

$$E_i(t) = \zeta_i(y_{i,0}, t)$$

We assume that the number of local agents is n . We also assume that there is an imaginary super-agent that observes A . When this agent does not observe anything, this means no other agent can.

$$G_{co,DL,obs} = (Y, \Sigma_o, \zeta^{co}, y_0) = Ac(2^Q \times 2^Q \times 2^Q \times \dots \times 2^Q, \Sigma_o, \zeta^{co}, y_0)$$

where

$$\begin{aligned} y_0 &= (y_0(1), y_0(2), y_0(3), \dots, y_0(n), y_0(n+1)) \\ &= (UR_1(Q_0), UR_2(Q_0), UR_3(Q_0), \dots, UR_z(Q_0), UR(Q_0)) \end{aligned}$$

ζ^{co} is also a vector of $n + 1$ elements.

7.2 Definitions of Network Co-Detectabilities

Now we define network co-detectabilities as follows.

Definition 11 (Network Co-detectability)

A networked discrete event system G is (weakly) network co-detectable if, after a finite number of observations, the current state and subsequent states of the system is known to at least one agent for some trajectories of the system. that is,

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow (\exists i \in I)|E_i(P_i(t'))| = 1)$$

Definition 12 (Strong Network Co-detectability)

A networked discrete event system G is strongly network co-detectable if, after a finite number of observations, the current state and subsequent states of the system is known to at least one agent for all trajectories of the system. that is,

$$(\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in \theta_{DL}(s))(\forall t' \in Pr(t))(|t'| > n \Rightarrow (\exists i \in I)|E_i(P_i(t'))| = 1)$$

Definition 13 (Weak Network Periodic Co-detectability)

A networked discrete event system G is (weakly) periodically network co-detectable if the current state of the system is known periodically to at least one agent for some trajectories of the system.

Formally,

$$(\exists n \in \mathbb{N})(\exists s \in L^\omega(G))(\exists t \in \theta_{DL}(s))(\forall t' \in Pr(t))(\exists t'' \in \Sigma_o^*) \\ (t't'' \in Pr(t) \wedge |t''| < n \wedge (\exists i \in I)|E_i(P_i(t't''))| = 1).$$

Definition 14 (Strong Network Periodic Co-detectability)

A networked discrete event system G is strongly periodically network co-detectable if the current state of the system is known periodically to at least one agent for all trajectories of the system.

Formally,

$$(\exists n \in \mathbb{N})(\forall s \in L^\omega(G))(\forall t \in \theta_{DL}(s))(\forall t' \in Pr(t))(\exists t'' \in \Sigma_o^*) \\ (t't'' \in Pr(t) \wedge |t''| < n \wedge (\exists i \in I)|E_i(P_i(t't''))| = 1).$$

7.3 Checking Network Co-detectabilities

In order to check network co-detectabilities, we first construct networked observer $G_{co,DL,obs}$. We then mark the states in $G_{co,DL,obs}$ that contain a singleton state and denote the set by:

$$Y_m = \{y \in Y : (\exists y(i)) |y(i)| = 1 \wedge (1 \leq i \leq n)\}.$$

The state in G is known when $G_{co,DL,obs}$ is in Y_m . If that is the case, then there is at least one agent that knows the current state of the system. We depend on loops to check co-detectabilities. Therefore, Let us denote the set of all loops in $G_{co,DL,obs}$ as

$$Loop = \{(y, u) \in Y \times \Sigma_0^* : |u| \geq 1 \wedge \zeta^{co}(y, u) = y\}$$

Theorem 14

A networked nondeterministic discrete event system G is strongly network co-detectable with respect of a set of remote agents if and only if in the networked co-observer $G_{co,DL,obs}$,

$$(\forall (y, u) \in Loop) (\forall w \in \Sigma_0^*) \zeta^{co}(y, w) \in Y_m.$$

In other words, any state reachable from any loop in $G_{co,DL,obs}$ is in Y_m .

Theorem 15

A networked nondeterministic discrete event system G is network co-detectable with respect of a set of remote agents if and only if in the networked co-observer $G_{co,DL,obs}$,

$$(\exists (y, u) \in Loop) (\forall w \in Pr(u)) \zeta^{co}(y, w) \in Y_m.$$

In other words, there are loops in $G_{co,DL,obs}$ which are completely inside Y_m .

Theorem 16

A networked nondeterministic discrete event system G is strongly periodically network co-detectable with respect of a set of remote agents if and only if in the networked co-observer $G_{co,DL,obs}$,

$$(\forall (y, u) \in Loop) (\exists w \in Pr(u)) \zeta^{co}(y, w) \in Y_m,$$

that is, every loop in $G_{co,DL,obs}$ must contain at least one state belonging to Y_m .

Theorem 17

A networked nondeterministic discrete event system G is periodically network co-detectable with respect of a set of remote agents if and only if in the networked co-observer $G_{co,DL,obs}$,

$$(\exists(y, u) \in Loop)(\exists w \in Pr(u)) \zeta^{co}(y, w) \in Y_m,$$

That is, there are loops in $G_{co,DL,obs}$ that include at least one state belonging to Y_m .

Proofs for theorems 14, 15, 16, and 17 are similar to the proofs of theorems 1, 2, 3, and 4.

Therefore, proofs are omitted.

7.4 An Algorithm to Check Network Co-detectabilities of Discrete Event Systems

In summary, we can check network co-detectability, strong network co-detectability, periodically network co-detectability, and strongly periodically network co-detectability using the following algorithm

Algorithm 4

Input: A networked nondeterministic discrete event system G

An observation mapping for each agent $\theta_{i,DL}$ with delays bounded by N_i .

Output: Network co-detectable (= yes or no)

Strongly network co-detectable (= yes or no)

Periodically network co-detectable (= yes or no)

Strongly periodically network co-detectable (= yes or no)

Step 1: $G_{i,L} = LOSS(G)$;

Step 2: $G_{i,L,obs} = OBS(G_{i,L})$;

Step 3: $G_{co,DL,obs} = (Y, \Sigma_o, \zeta^{co}, y_0)$;

Step 4: $Y_m = \{y \in Y: |y| = 1\}$;

Step 5: $Y_m = \{y \in Y: (\exists y(i)) |y(i)| = 1 \wedge (1 \leq i \leq n)\}$;

Step 6: If $(\forall (y, u) \in Loop)(\forall w \in \Sigma_0^*) \zeta^{co}(y, w) \in Y_m$ is true, then

Strongly network co-detectable = yes;

else

Strongly network co-detectable = no;

Step 7: If $(\exists (y, u) \in Loop)(\forall w \in Pr(u)) \zeta^{co}(y, w) \in Y_m$ is true, then

Network co-detectable = yes;

else

Network co-detectable = no.

Step 8: If $(\forall (y, u) \in Loop)(\exists w \in Pr(u)) \zeta^{co}(y, w) \in Y_m$ is true, then

Strongly periodically network co-detectable = yes;

else

Strongly periodically network co-detectable = no;

Step 9: If $(\exists (y, u) \in Loop)(\exists w \in Pr(u)) \zeta^{co}(y, w) \in Y_m$ is true, then

Periodically network co-detectable = yes;

else

Periodically network co-detectable = no.

CHAPTER 8 CONCLUSION

In this dissertation, we have defined network detectability, network D-detectability, network I-detectability, and network co-detectability of discrete event systems. We derived necessary and sufficient conditions for network detectability, network D-detectability, network I-detectability, and network co-detectability. We developed algorithms to check all types of network detectabilities. We also discussed and proved some properties of networked discrete event systems. Also, many examples have been given to illustrate different types of network detectabilities.

REFERENCES

- [1] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed.: Springer, 2007.
- [2] F. Lin and W. Wonham, "Decentralized supervisory control of discrete-event systems," *Information Sciences* vol. 44, pp. 199-224, 1988.
- [3] P. Ramadge and W. Wonham, "Supervisory control of a class of discrete event processes," *SIAM journal on control and optimization*, vol. 25, pp. 206-230, 1987.
- [4] R. Cieslak, C. Desclaux, A. S. Fawaz, and P. Varaiya, "Supervisory control of discrete-event processes with partial observations," *IEEE transactions on automatic control*, vol. 33, pp. 249-260, 1988.
- [5] A. F. Vaz and W. M. Wonham, "On supervisor reduction in discrete event systems," *On supervisor reduction in discrete event systems*, vol. 44, pp. 475-491, 1986.
- [6] J. N. Tsitsiklis, "On the control of discrete-event dynamical systems," *Decision and Control, 1987. 26th IEEE Conference on*, vol. 26, pp. 419-422, 1987.
- [7] F. Lin and W. Wonham, "On observability of discrete-event systems," *Information Sciences*, vol. 44, pp. 173-198, 1988.
- [8] J. S. Ostroff and W. M. Wonham, "A temporal logic approach to real time control," *In Decision and Control, 1985 24th IEEE Conference on*, vol. 24, pp. 656-657, 1985.
- [9] K. Rudie and W. Wonham, "Think globally, act locally: decentralized supervisory control," *IEEE Transactions on Automatic Control* vol. 37, pp. 1692-1708, 1992.
- [10] M. Heymann and F. Lin, "On-line control of partially observed discrete event systems," *Discrete Event Dynamic Systems: Theory and Applications* vol. 4, pp. 221-236, 1994.

- [11] S. Chung, S. Lafortune, and F. Lin, "Limited lookahead policies in supervisory control of discrete event systems," *IEEE Transactions on Automatic Control* vol. 37, pp. 1921-1935, 1992.
- [12] F. Lin, "Robust and adaptive supervisory control of discrete event systems," *IEEE Transactions on Automatic Control* vol. 38, pp. 1848-1852, 1993.
- [13] W. M. Wonham, "Towards an Abstract Internal Model Principle," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, pp. 735-740, 1976.
- [14] P. Ramadge, "Observability of discrete event systems," *In 1986 25th IEEE Conference on Decision and Control*, vol. 25, pp. 1108-1112, 1986.
- [15] C. Ozveren and A. Willsky, "Observability of discrete event dynamic systems," *IEEE Transactions on Automatic Control* vol. 35, pp. 797-806, 1990.
- [16] M. E. Merchant, "Production: A dynamic challenge: What's needed is flexibility \hat{A}_i The ability to manufacture efficiently a constantly changing variety of products," *IEEE spectrum*, vol. 20, pp. 36-39, 1983.
- [17] G. A. Frank, D. L. Franke, and W. F. Ingogly, "An architecture design and assessment system," *VLSI Design*, vol. 6, pp. 30-50, 1985.
- [18] W. B. Gevarter, "Expert systems: limited but powerful," *IEEE*, 1983.
- [19] S. Shu and F. Lin, "Generalized detectability for discrete event systems," *Systems & Control Letters* vol. 60, pp. 310-317, 2011.
- [20] S. Shu, F. Lin, and H. Ying, "Detectability of discrete event systems," *IEEE Transactions on Automatic Control*, vol. 52, pp. 2356-2359, 2007.

- [21] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete event systems," *IEEE Transactions on Automatic Control*, vol. 40, pp. 1555-1575, 1995.
- [22] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Failure diagnosis using discrete event models," *IEEE Transactions on Control System Technology*, vol. 4, pp. 105-124, 1996.
- [23] T. S. Yoo and S. Lafortune, "Polynomial-time verification of diagnosability of partially observed discrete-event systems," *IEEE Transactions on automatic control*, vol. 47, pp. 1491-1495, 2002.
- [24] S. Jiang, Z. Huang, V. Chandra, and R. Kumar, "A polynomial algorithm for testing diagnosability of discrete-event systems," *EEE Transactions on Automatic Control*, vol. 46, pp. 1318-1321, 2001.
- [25] F. Lin, "Diagnosability of discrete event systems and its applications," *Discrete Event Dynamic Systems*, vol. 4, pp. 197-212, 1994.
- [26] R. Debouk, S. Lafortune, and D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete event systems," *Decision and Control, 1998. Proceedings of the 37th IEEE conference on*, vol. 4, pp. 3763-3768, 1998.
- [27] D. Thorsley and D. Teneketzis, "Diagnosability of stochastic discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 50, pp. 476-492, 2005.
- [28] M. Sampath, S. Lafortune, and D. Teneketzis, "Active diagnosis of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 43, pp. 908-929, 1998.
- [29] O. Contant, S. Lafortune, and D. Teneketzis, "Diagnosability of discrete event systems with modular structure," *Discrete Event Dynamic Systems*, vol. 16, pp. 9-37, 2006.

- [30] M. V. Moreira, T. C. Jesus, and J. C. Basilio, "Polynomial time verification of decentralized diagnosability of discrete event systems," *IEEE Transactions on Automatic Control*, vol. 56, pp. 1679-1684, 2011.
- [31] W. Qiu and R. Kumar, "Decentralized failure diagnosis of discrete event systems," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 36, pp. 384-395, 2006.
- [32] Y. Pencolé, "Diagnosability analysis of distributed discrete event systems," *In Proceedings of the 16th European Conference on Artificial Intelligence*, pp. 38-42, 2004.
- [33] S. Jiang and R. Kumar, "Failure diagnosis of discrete-event systems with linear-time temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 49, pp. 934-945, 2004.
- [34] A. Ramirez-Trevino, E. Ruiz-Beltran, I. Rivera-Rangel, and E. Lopez-Mellado, "Online fault diagnosis of discrete event systems. A Petri net-based approach," *IEEE Transactions on Automation Science and Engineering*, vol. 4, pp. 31-39, 2007.
- [35] A. Paoli and S. Lafortune, "Safe diagnosability for fault-tolerant supervision of discrete-event systems," *Automatica*, vol. 41, pp. 1335-1347, 2005.
- [36] F. Basile, P. Chiacchio, and G. De Tommasi, "An efficient approach for online diagnosis of discrete event systems," *IEEE Transactions on Automatic Control*, vol. 54, pp. 748-759, 2009.
- [37] S. Haar, A. Benveniste, E. Fabre, and C. Jard, "Partial order diagnosability of discrete event systems using Petri net unfoldings," *In Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 4, pp. 3748-3753, 2003.

- [38] A. Benveniste, E. Fabre, S. Haar, and C. Jard, "Diagnosis of asynchronous discrete-event systems: a net unfolding approach," *IEEE Transactions on Automatic Control*, vol. 48, pp. 714-727, 2003.
- [39] T. Jéron, H. Marchand, S. Pinchinat, and M. O. Cordier, "Supervision patterns in discrete event systems diagnosis," *In Discrete event systems, 2006 8th international workshop on. IEEE*, pp. 262-268, 2006.
- [40] A. Ramírez-Treviño, E. Ruiz-Beltran, I. Rivera-Rangel, and E. Lopez-Mellado, "Diagnosability of discrete event systems: a petri net based approach," *In Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on. IEEE*, vol. 1, pp. 541-546, 2004.
- [41] Y. Wang, T. S. Yoo, and S. Lafortune, "Diagnosis of discrete event systems using decentralized architectures," *Discrete Event Dynamic Systems*, vol. 17, pp. 233-263, 2007.
- [42] S. Jiang, R. Kumar, and H. E. Garcia, "Diagnosis of repeated/intermittent failures in discrete event systems," *IEEE transactions on robotics and automation*, vol. 19, pp. 310-323, 2003.
- [43] S. Lafortune, D. Teneketzis, M. Sampath, R. Sengupta, and K. Sinnamohideen, "Failure diagnosis of dynamic systems: An approach based on discrete event systems," *In American Control Conference, 2001. Proceedings of the 2001. IEEE*, vol. 3, pp. 2058-2071, 2001.
- [44] L. K. Carvalho, M. V. Moreira, and J. C. Basilio, "Generalized robust diagnosability of discrete event systems," *IFAC Proceedings Volumes*, vol. 44, pp. 8737-8742, 2011.
- [45] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of discrete-event systems using labeled Petri nets," *IEEE Transactions on Automation Science and Engineering*, vol. 11, pp. 144-153, 2014.

- [46] T. S. Yoo and S. Lafortune, "NP-completeness of sensor selection problems arising in partially observed discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 47, pp. 1495-1499, 2002.
- [47] G. Lamperti and M. Zanella, "Diagnosis of discrete-event systems from uncertain temporal observations," *Artificial Intelligence*, vol. 137, pp. 91-163, 2002.
- [48] S. Bavishi and E. K. Chong, "Automated fault diagnosis using a discrete event systems framework," *In Intelligent Control, 1994., Proceedings of the 1994 IEEE International Symposium on*, pp. 213-218, 1994.
- [49] M. O. Cordier, L. Travé-Massuyes, and X. Pucel, "Comparing diagnosability in continuous and discrete-event systems," *In Proceedings of the 17th International Workshop on Principles of Diagnosis*, vol. (DX-06), pp. 55-60, 2006.
- [50] J. Chen and R. Kumar, "Polynomial test for stochastic diagnosability of discrete-event systems," *IEEE Transactions on Automation Science and Engineering*, vol. 10, pp. 969-979, 2013.
- [51] S. H. Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in discrete-event systems: Incorporating timing information," *IEEE Transactions on Automatic Control*, vol. 50, pp. 1010-1015, 2005.
- [52] R. Kumar and S. Takai, "Inference-based ambiguity management in decentralized decision-making: Decentralized diagnosis of discrete-event systems," *IEEE Transactions on Automation Science and Engineering*, vol. 6, pp. 479-491, 2009.
- [53] F. Lin and T. W. Lin, "Diagnosability of discrete event systems and its applications to circuit testing," *In Circuits and Systems, 1993., Proceedings of the 36th Midwest Symposium on. IEEE*, pp. 344-347, 1993.

- [54] A. Paoli and S. Lafortune, "Safe diagnosability of discrete event systems," *In Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 3, pp. 2658-2664, 2003.
- [55] J. Zaytoon and S. Lafortune, "Overview of fault diagnosis methods for discrete event systems," *Annual Reviews in Control*, vol. 37, pp. 308-320, 2013.
- [56] J. C. Basilio and S. Lafortune, "Robust codiagnosability of discrete event systems," *In American Control Conference. IEEE*, vol. ACC'09, pp. 2202-2209, 2009.
- [57] J. Lunze and J. Schröder, "State observation and diagnosis of discrete-event systems described by stochastic automata," *Discrete Event Dynamic Systems*, vol. 11, pp. 319-369, 2001.
- [58] E. Kilic, "Diagnosability of fuzzy discrete event systems," *Information Sciences*, vol. 178, pp. 858-870, 2008.
- [59] Y. Pencolé and M. O. Cordier, "A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks," *Artificial Intelligence*, vol. 164, pp. 121-170, 2005.
- [60] Y. L. Chen and G. Provan, "Modeling and diagnosis of timed discrete event systems-a factory automation example," *American Control Conference, 1997. Proceedings of the 1997. IEEE*, vol. 1, pp. 31-36, 1997.
- [61] R. Debouk, S. Lafortune, and D. Teneketzis, "On the effect of communication delays in failure diagnosis of decentralized discrete event systems," *In Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, vol. 3, pp. 2245-2251, 2000.
- [62] S. H. Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in discrete-event systems: Framework and model reduction," *In Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, vol. 4, pp. 3769-3774, 1998.

- [63] F. Liu and D. Qiu, "Diagnosability of fuzzy discrete-event systems: A fuzzy approach," *IEEE Transactions on Fuzzy Systems*, vol. 17, pp. 372-384, 2009.
- [64] S. H. Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in timed discrete-event systems," *In Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 2, pp. 1756-1761, 1999.
- [65] J. Dubreil, P. Darondeau, and H. Marchand, "Supervisory control for opacity," *IEEE Transactions on Automatic Control*, vol. 55, pp. 1089-1100, 2010.
- [66] R. Jacob, J. Lesage, and J. Faure, "Opacity of discrete event systems: models, validation and quantification," *IFAC-PapersOnLine* vol. 48, pp. 174-181, 2015.
- [67] F. Lin, "Opacity of discrete event systems and its applications," *Automatica* vol. 47, pp. 496-503, 2011.
- [68] A. Saboori and C. Hadjicostis, "Notions of security and opacity in discrete event systems," *Proceedings of the 46th IEEE Conference on Decision and Control* pp. 5056-5061, 2007.
- [69] A. Saboori and C. Hadjicostis, "Opacity-enforcing supervisory strategies for secure discrete event systems," *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 889-894, 2008.
- [70] A. Saboori and C. Hadjicostis, "Verification of initial-state opacity in security applications of discrete event systems," *Information Sciences*, vol. 246, pp. 115-132, 2013.
- [71] Ben-Kalefa and F. Lin, "Supervisory control for opacity of discrete event systems," *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on. IEEE*, pp. 1113-1119, 2011.

- [72] W. Wang, S. Lafortune, and F. Lin, "An algorithm for calculating indistinguishable states and clusters in finite-state automata with partially observable transitions," *Systems & Control Letters*, vol. 56, pp. 656-661, 2007.
- [73] S. Shu and F. Lin, "I-Detectability of Discrete-Event Systems," *IEEE Transactions on Automation Science and Engineering*, vol. 10, pp. 187-196, 2013.
- [74] S. Shu and F. Lin, "Delayed Detectability of Discrete Event Systems," *IEEE Transactions on Automatic Control*, vol. 58, pp. 862-875, 2013.
- [75] S. Shu and F. Lin, "Co-Detectability of Multi-Agent Discrete Event Systems," *Proceedings of 2011 Chinese Control and Decision Conference*, pp. 1708-1713, 2011.
- [76] R. Gupta and M. Chow, "Networked control system: Overview and research trends," *IEEE Transactions on Industrial Electronics*, vol. 57, pp. 2527-2535, 2010.
- [77] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of IEEE*, vol. 95, pp. 138-162, 2007.
- [78] S. Balemi, "Input/output discrete event processes and communication delays," *Discrete Event Dynamic Systems*, vol. 4, pp. 41-85, 1994.
- [79] F. Lin, "Control of networked discrete event systems: dealing with communication delays and losses," *SIAM Journal of Control and Optimization* vol. 52, pp. 1276-1298, 2014.
- [80] S. Park and K. Cho, "Supervisory control of discrete event systems with communication delays and partial observations," *Systems & Control Letters* vol. 56, pp. 106-112, 2007.
- [81] K. Rudie, S. Lafortune, and F. Lin, "Minimal communication in a distributed discrete event system," *IEEE Transactions on Automatic Control* vol. 48, pp. 1965-1970, 2003.

- [82] W. Wang, S. Lafortune, and F. Lin, "On the minimization of communication in networked systems with a central station," *Discrete Event Dynamic Systems: Theory and Applications* vol. 18, pp. 415-443, 2008.
- [83] W. Wang, S. Lafortune, and F. Lin, "Minimization of communication of event occurrences in acyclic discrete event systems," *IEEE Transactions on Automatic Control*, vol. 53, pp. 2197-2202, 2008.
- [84] W. Wang, S. Lafortune, and F. Lin, "Minimization of dynamic sensor activation in discrete event systems for the purpose of control," *IEEE Transactions on Automatic Control*, vol. 55, pp. 2447-2461, 2010.
- [85] S. Park and K. Cho, "Decentralized supervisory control of discrete event systems with communication delays based on conjunctive and permissive decision structure," *Automatica*, vol. 43, pp. 738-743, 2007.
- [86] F. Lin, K. Rudie, and S. Lafortune, "Minimal communication for essential transitions in a distributed discrete-event system," *IEEE Transactions on Automatic Control* vol. 52, pp. 1495-1502, 2007.
- [87] L. Carvalho, J. Basilio, and M. Moreira, "Robust diagnosability of discrete event systems subject to intermittent sensor failures," *Proceedings of the 10th International Workshop on Discrete Event Systems* vol. 43, pp. 84-89, 2010.
- [88] W. M. Wonham and P. J. Ramadge, "Modular supervisory control of discrete-event systems," *Mathematics of control, signals and systems* vol. 1, pp. 13-30, 1988.
- [89] P. J. Ramadge, "Control and supervision of discrete event processes," Ph.D Electrical Engineering University of Toronto, 1983.

- [90] W. M. Wonham and P. J. Ramadge, "On the supremal controllable sublanguage of a given language," *SIAM Journal on Control and Optimization*, vol. 25, pp. 637-659, 1987.
- [91] P. E. Caines, R. Greiner, and S. Wang, "Dynamical logic observers for finite automata," *In Decision and Control, 1988., Proceedings of the 27th IEEE Conference on*, pp. 226-233, 1988.
- [92] S. J. Rosenschein, "Formal theories of knowledge in AI and robotics," *New generation computing*, vol. 3, pp. 345-357, 1985.
- [93] S. J. Rosenschein and L. P. Kaelbling, "The synthesis of digital machines with provable epistemic properties," *Proceedings of the 1986 Conference on Theoretical aspects of reasoning about knowledge*, pp. 83-98, 1986.
- [94] P. E. Caines, *Linear stochastic systems* vol. 11. New York: Wiley, 1988.
- [95] G. Tadmor and O. Maimon, "Control of large discrete event systems: constructive algorithms," *IEEE Transactions on Automatic Control*, vol. 34, pp. 1164-1168, 1989.
- [96] L. Tobias and J. Scoggins, "Time-based air-traffic management using expert systems," *IEEE Control Systems Magazine*, vol. 7, pp. 23-29, 1987.
- [97] F. Lin and H. Ying, "Modeling and control of fuzzy discrete event systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, pp. 408-415, 2002.
- [98] F. Lin, H. Ying, X. Luan, R. D. MacArthur, J. A. Cohn, D. Barth-Jones, *et al.*, "Theory for a control architecture of fuzzy discrete event systems for decision making," *In Proceedings of the 44th IEEE Conference on Decision and Control*, 2005.

- [99] S. H. Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in discrete-event systems: framework and model reduction," *IEEE Transactions on Automatic Control*, vol. 48, pp. 1199-1212, 2003.
- [100] S. Jiang and R. Kumar, "Diagnosis of repeated failures for discrete event systems with linear-time temporal-logic specifications," *IEEE Transactions on Automation Science and Engineering*, vol. 3, pp. 47-59, 2006.
- [101] M. Heymann and F. Lin, "Discrete-event control of nondeterministic systems," *IEEE Transactions on Automatic Control*, vol. 43, pp. 3-17, 1998.
- [102] J. G. Thistle and W. M. Wonham, "Control of infinite behavior of finite automata," *SIAM Journal on Control and Optimization*, vol. 32, pp. 1075-1097, 1994.
- [103] N. Ha, "Discrete-event systems abstractions for on-line and real-time hierarchical control," Doctor of Philosophy school of computer science and engineering Nanyang Technological University 2016.
- [104] V. Mitin, D. Romanov, and M. Polis, *Modern advanced mathematics for engineers*. New York: John wiley & sons, INC, 2001.
- [105] Y. Brave, "On stabilization of discrete event processes," in: *Proceedings of CDC*, pp. 2737-2743, 1989.
- [106] C. M. Ozveren and A. S. Willsky, "Output stabilizability of discrete-event dynamic systems," *IEEE Transactions on Automatic Control*, vol. 36, pp. 925-935, 1991.
- [107] Y. Brave and M. Heymann, "Stabilization of discrete-event processes," *International Journal of Control*, vol. 51, pp. 1101-1117, 1990.

ABSTRACT**ON DETECTABILITY OF NETWORKED DISCRETE EVENT SYSTEMS**

by

YAZEED SASI**December 2017****Advisor:** Dr. Feng Lin**Major:** Electrical Engineering**Degree:** Doctor of Philosophy

Detectability of discrete event systems, the ability to determine the current and subsequent states, is very important in supervisory control and many other applications. So far only detectability of non-networked discrete event systems has been defined and investigated. Non-networked discrete event systems assume all the communication to be carried out on time without any delays or losses. The assumption of reliable link is true when the distance of communication is short; however, it is often violated in networked systems. In my dissertation, I investigate the detectability for the networked discrete event systems. Because applications vary, we investigate the four types of the network detectabilities: detectability, strong detectability, periodic detectability, and strong periodic detectability. In addition, I will investigate the network D-detectability, which is the ability to just distinguish certain pairs of states. As in non-networked discrete event systems, I will extend the network detectability to network I-Detectability, and network Co-detectability. Network I-detectability is defined as the ability of determining the initial state of the system after finite numbers of event of observations when the system is subject to communication delays and losses. Network Co-detectability, on the other hand, is defined as the

ability of determining the current state and subsequent states of the system with at least one agent under communication delays losses. In each case, I will define and prove the necessary and sufficient condition for the detectabilities if possible. In some cases, methods to check types of network detectabilities are developed. Examples are also given to illustrate different types of results.

AUTOBIOGRAPHICAL STATEMENT

Yazeed Sasi

Education

Yazeed Sasi received his B.Sc.Eng. degree in electrical engineering from El-Fateh University (currently known as Tripoli University), Tripoli, Libya, in 2005. He received his M.Sc. degree in electrical engineering from Wayne State University, Detroit, USA, in 2010. He is currently a PhD candidate at Wayne State University, Detroit, USA.

From 2005 to 2006, he worked for Waha Oil Company, Tripoli, Libya. He then worked as a teaching assistant in El-Fateh University from 2006 to 2008. His research interests are in the area of discrete event systems and control.

He is a member of Tau Beta Pi - The Engineering Honor Society. He is also a member of Golden Key International Honour Society.

Journal Paper

Y. Sasi, and F. Lin, "Detectability of networked discrete event systems," Conditionally accepted by Discrete Event Dynamic Systems.