Opacity and Security Concepts for Discrete Event Systems, Linear Time-invariant Systems, and Max-Plus Linear Systems

Master of Science (M.Sc.) Thesis

Nathan Maguire

Supervisor: Prof. Oliver Mason
Head of Department: Prof. Stephen Buckley

Department of Mathematics & Statistics
Maynooth University

September 2025

Contents

List of Figures

Abstract

In this thesis, we review concepts in privacy and security for different classes of dynamical systems, with particular emphasis on opacity and attack detection. Opacity has attracted significant attention in recent years due to its role in tackling privacy-related problems within the area of system and control theory. Opacity is an information-flow property that is concerned with a system ability to hide information from an external observer. This property plays a key role in strengthening resilience against attacks and prevents adversaries from determining if their attacks have succeeded.

We begin the thesis by examining opacity in its original setting of discrete event systems (DES) and consider a more recent adaptation for linear time-invariant (LTI) systems. In addition, we also provide some initial thoughts on how opacity might be formulated in the context of max-plus linear systems. Although max-plus systems constitute a subclass of DES, their formal structure resembles that of LTI systems. These models arise in practical setting such as manufacturing systems, communication networks, and railway systems, where synchronization and timing constraints are critical.

To ensure the reliable operation of any system, it is essential to design mechanisms that mitigate the effects of malicious behaviour. This thesis also reviews concepts in security with a focus on attack detection, and discusses the connection between opacity and the notion of undetectable attacks.

Acknowledgments

First and foremost, I would like to sincerely thank my supervisor, Prof. Oliver Mason, for his consistent support, encouragement and guidance throughout the past few years. I have been truly fortunate to work under his supervision; his mentorship not only made my Master's an enjoyable and rewarding experience, but also inspired me to pursue further studies in mathematics. I am confident that many students before me have expressed similar appreciation for Oliver and I know I definitely won't be the last.

I was very fortunate to have the opportunity to share my thoughts and ideas on the thesis with fellow postgraduates during the graduate seminar organised by Dr. Detta Dickinson. This experience greatly improved my presentation skills and enhanced my ability to express academic ideas clearly to my peers.

I am deeply grateful for the opportunity to work within the Mathematics and Statistics department here at Maynooth University. Completing this work was made far more enjoyable and manageable thanks to the support, encouragement, and friendship of the staff in both departments, as well as my fellow departmental tutors. Their help and kindness have been invaluable.

On a more personal level, I would like to thank my Mam and Dad, and my siblings Niamh, Caoimhe and Emily for their support, encouragement, and always believing in me throughout the years. I am especially grateful for their constant presence and for standing by me during my Master's journey.

I would also like to mention some friendships that helped me throughout my Master's. In particular, I am thankful to Ciaran O'Connor and Luke Cowley for checking in and encouraging me during the more challenging times of my Master's. I would also like to thank Alex Finegan, Jonathan Everitt, and Shaun Crabtree, whose support, whether that may have been through shared runs or trips to the gym, reminded me of the importance of taking breaks and maintaining balance.

Chapter 1

Introduction and Layout

In this introductory chapter, we provide a brief description of the main objectives of the thesis. We also give some motivation for the work and outline the structure of the chapters that follow.

1.1 Objective and Motivation

The objective of this thesis is to present a study of recent work on topics related to privacy and security for different classes of systems, with a particular emphasis on opacity and attack detection. In this context, privacy is studied through the lens of opacity. The notion of opacity has attracted increasing attention in recent years due to its role in analysing privacy-related problems within the area of systems and control theory [24], [30] and [11]. Loosely speaking, opacity is an information-flow property that is concerned with a system's ability to hide secret information from an external intruder. A system is opaque if, based on the observable behaviour, an intruder cannot determine with certainty whether the system has engaged in a behaviour it wishes to keep secret. This property is particularly significant in cyberphysical systems such as smart grids, transport networks, and healthcare systems, where sensitive user data may be unintentionally revealed through the system's outputs [2] and [52]. Opacity plays a key role in strengthening resilience against attacks, as attackers often rely on inferring hidden information to carry out effective intrusions. Ensuring opacity helps prevent adversaries from confirming whether their attacks have succeeded [53].

To ensure the reliable operation of any system, it is essential to design and implement security measures that protect against attacks. An important component of such measures is *attack detection*, which enables the system to respond appropriately and mitigate potential damage. High-profile real-world examples of attacks include the Ukraine power grid attack in 2015 [49] and the Iranian oil terminal attack in 2012 [42]. Early and effective detection of such attacks can help reduce their impact.

In this thesis, opacity is examined for multiple system classes. We begin with its original formulation for deterministic automata models of discrete event systems (DES) and then consider a more recent adaptation for linear time-invariant (LTI) systems. To the best of our knowledge, no existing work in the literature addresses how opacity might be formulated in the max-plus linear system setting. At the end of this thesis, we present some initial thoughts on how opacity might be formulated in this setting, and highlight some technical difficulties in the max-plus setting. Max-plus linear systems are essentially a form of DES but formally look similar to LTI systems. Such systems arise in manufacturing systems, communication networks, and railway systems. In particular, the Dutch railway has been effectively studied and modelled using a max-plus linear approach [47].

1.2 Key Questions

At a broad level, the main questions discussed in the thesis are as follows.

1. How is opacity defined and studied in its original setting of discrete-event systems?
2. How is opacity formulated in the context of LTI systems?
3. What is the relationship between opacity and undetectable attacks for LTI systems?
4. How does opacity relate to other system properties for various system classes?
5. In what ways might opacity be formulated and applied to max-plus linear systems?

1.3 Overview

The structure of this thesis is described below.

- In Chapter 2, we discuss opacity for discrete event systems. We describe several notions of opacity from the literature, with particular emphasis on state-based formulations for finite-state automata.
- Chapter 3 reviews the recently introduced notion of opacity for linear time-invariant (LTI) systems. We relate opacity to classical system properties such as controllability, output controllability, reachability, and observability.
- Chapter 4 addresses security and attack detection concepts for LTI systems. We discuss the problem of detectable attacks and the related question of characterising undetectable attacks. We then describe work on the connection between undetectable attacks and opacity for LTI systems.
- In Chapter 5, we survey some fundamental definitions and concepts in max-plus algebra, examine its relation to graph theory, and recall some methods for solving equations in max-plus algebra.
- In Chapter 6, we consider linear systems within the max-plus framework. In particular, we give a brief overview of some control-theoretic properties in the max-plus algebra and offer some initial thoughts of formulating opacity in the context of max-plus linear systems, with particular emphasis on its connection to reachability.
- Concluding remarks and a discussion of possible future work are given in Chapter 7.

Chapter 2

Opacity for discrete-event systems

2.1 Introduction

Opacity will be a major focus of this thesis. In this chapter, we discuss some opacity concepts in the setting of discrete-event systems. This is the setting in which opacity was initially formulated. In later chapters, we will discuss opacity for other system classes.

Discrete-event systems (DES) are dynamical systems where state changes occur in response to events at discrete times. As such, the dynamics of such systems are event-driven. These types of systems naturally arise in many real-world applications. For example, discrete-event systems are used in manufacturing and production applications, which involve machines and conveyors; in healthcare, where DES can be used to simulate patient flow and resource utilization; and cyber-physical systems, such as automated vehicles, robotics, and smart appliances [60], [73], [40].

We will begin the chapter by describing the type of model we use in our discussion of opacity for discrete event systems. We will focus on models given by deterministic finite-state automata. Here, if an event triggers a transition at some state, the state to which the system moves is uniquely determined. However, not all state-event pairs will lead to such a transition. We focus on this simple model to help make the different notions of opacity clear. Here and in future chapters, we are mainly interested in state-based opacity, but we will also

provide a discussion on the language-based approach, as this was how opacity was originally formulated. We will also briefly discuss the problems of verifying and enforcing opacity. Lastly, we will introduce the concepts of controllability and observability for DES. These system properties will play a key role in the next chapter as we explore how they relate to the concept of opacity in a linear system setting.

2.2 Deterministic Finite State Automata and Related Models

As mentioned in the introduction, we will work with deterministic finite state automata (DFSA) models when examining opacity for discrete-event systems [9]. It is important to emphasize that the concept of opacity also arises in different system models. For example, in the works of [71] and [68], opacity is studied in a non-deterministic setting. Opacity can also be defined for stochastic models, such as in [7] and [36]. Now, let's state the definition of a DFSA.

Definition 2.2.1. *(Deterministic finite-state automaton [9])*
A deterministic finite state automaton (DFSA), denoted by G, is the five-tuple,

$$G = (X, E, f, \Phi, X_0)$$

where X is the finite set of states, E is the finite set of events associated with the transitions in G, $f : D \to X$ is the transition function, where $D \subseteq X \times E$; $\Phi : X \to 2^E$ is the feasible event function, and X_0 is the set of possible initial states.

In relation to Definition 2.2.1, we make the following remarks.

- In the literature, the object G is often called the generator (which explains the notation G) or the state machine [51], [62].
- It is important to emphasize that the transition function f is only partially defined. This means that $f(x, e)$ is not necessarily defined for all state-event pairs (x, e).
- G is said to be deterministic as the function f is single-valued. By contrast, the transition function of a nondeterministic automaton is set-valued and maps from a subset of $X \times E$ to the power set 2^X. In this case, given a

feasible state-event pair (x, e), the state to which the system transitions is not uniquely defined in general.

- For each state $x \in X$, $\Phi(x)$ is the set of all events e for which $f(x, e)$ is defined and is called the feasible event set of G at the state x. In some literature, such as [30] and [41], Φ is not included when defining G. We won't explicitly write Φ when discussing an automaton unless the feasible event function is central to the discussion. $\Phi(x)$ is typically included if it is important to distinguish between feasible state-event pairs (x, e) that cause no transition, meaning $f(x, e) = x$, and non-feasible pairs.

- We view the event set E of a DES as an alphabet. A sequence of events that is taken from this alphabet forms a string. A string that consists of no events is called the empty string and we will denote it as ε. E^* is the set of all finite strings formed using elements in E (including ε).

- For a string $s \in E^*$, $|s|$ denotes the length of the string. We also use $\hat{s}$ to denote the prefix-closure of s defined as $\hat{s} = \{p \in E^* | \exists t \in E^* \text{ such that } pt = s\}$. The post-string s/p of s after p is defined as $s/p = t \in E^*$, where $pt = s$.

Next, we describe the evolution of a DFSA G from a given initial state $x_0 \in X_0$. Beginning at x_0, when an event $e \in \Phi(x_0) \subseteq E$ occurs, the system makes a transition to the state $f(x_0, e) \in X$. This process continues with a transition occurring at each state x when an event in $\Phi(x)$ occurs. It is possible to extend the partial definition of f from $X \times E$ to give a, partially defined, function on $X \times E^*$. This can be done in the following recursive manner:

$$f(x, \varepsilon) = x$$

$$f(x, se) = f(f(x, s), e) \text{ for } s \in E^* \text{ and } e \in E. \tag{2.1}$$

Of course, the above recursion is only defined when $f(x, s)$ is defined, and $e \in \Phi(f(x, s))$.

A language $L \subseteq E^*$ is a set of finite-length strings using events in E. The language that is generated by the system G describes how the system behaves and is defined as $\mathcal{L}(G, X_0) = \{s \in E^* | \exists x_0 \in X_0, f(x_0, s) \text{ is defined}\}$. $\mathcal{L}(G, X_0)$ is prefix-closed by its definition.

In the study of opacity, we consider partially observable systems, where we partition our event set E into an observable set E_{obs} and an unobservable event

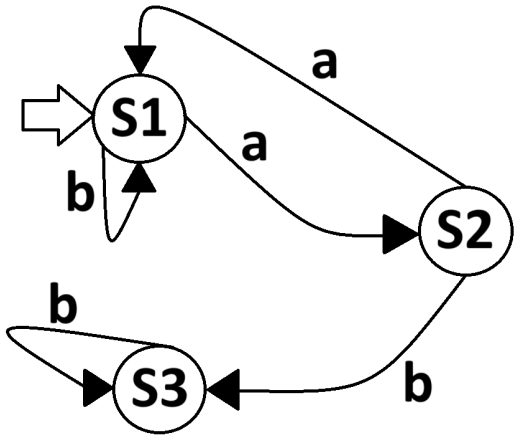
set E_{uo}. Given a string $t = e_1e_2...e_n \in E^*$, its observation is the output of the natural projection function $P : E^* \to E^*_{obs}$. The projection P for any string t is defined as $P(t) = P(e_1e_2...e_n) = P(e_1)P(e_2)...P(e_n)$ such that,

$$P(e_i) = \begin{cases} e_i & \text{if } e_i \in E_{obs} \\ \varepsilon & \text{if } e_i \notin E_{obs} \end{cases}$$

where P removes all unobservable events from a string and preserves only observable ones, in order. Also, for a language L, the inverse projection is defined as $P^{-1}(L) = \{t \in E^* : P(t) \in L\}$.
A labelled graph or state diagram is often a convenient way to represent a DFSA. The nodes are used to represent states, while labelled edges indicate the allowed transitions. We now show how this is done with the following example.

Example 2.2.1. *Consider the deterministic automaton depicted in the state diagram in Figure 2.1.*

Figure 2.1: State diagram for a deterministic automaton.



In this example, we have three states, $X = \{S1, S2, S3\}$, where $S1$ is the initial state. We also have two events labelled a and b, such that $E = \{a, b\}$. From this, the automaton gives us the following transition function table:

Transition Function		
Current State	*Event* $\boldsymbol{a}$	*Event* $\boldsymbol{b}$
S1	*S2*	*S1*
S2	*S1*	*S3*
S3	*undefined*	*S3*

Numerical simulation is a very important tool to investigate the behaviour of finite-state automata. For example, simulations can help us optimize processes, improve efficiency, and make informed decisions [74]. For DES simulation, there are Python software packages such as SimPy available. Note that we are not limited only to Python; for example, discrete event systems can also be simulated using MATLAB using the Simulink package [35]. While our example is very simple and can be analysed manually, we include a sample output below to illustrate the use of SimPy for DFSA simulation. The simulation takes in a string and determines the system's state after each event has occurred. Note that the thesis itself is not based on simulation work; rather, we wish to demonstrate that such simulations can be performed.

```
Sample Output 1:
Please enter the Events: baa
-----------------------------------
We begin at State 1.
From the language b we are in state 1!
From the language a we are in state 2!
From the language a we are in state 1!
-----------------------------------
The simulation is now complete!
```

```
Sample Output 2:
Please enter the Events: aaabb
_____________________________________

We begin at State 1.
From the language a we are in state 2!
From the language a we are in state 1!
From the language a we are in state 2!
From the language b we are in state 3!
From the language b we are in state 3!
_____________________________________

The simulation is now complete!
```

We can also display this information in the notation used in (2.1). Using 'Sample Output 1', we express our simulation in the following way:

$$f(S1, \varepsilon) = S1$$

$$\begin{aligned} f(S1, baa) = f(f(S1, ba), a) &= f(f(f(S1, b), a), a) \\ &= f(f(S1, a), a) \\ &= f(S2, a) = S1. \end{aligned}$$

2.3 Opacity Concepts for Finite State Automata

Introduced in [8], opacity is an information flow property that determines whether a system reveals its 'secrets' to an external observer, often referred to as the intruder. We assume that this intruder has full knowledge of the structure of the system, but can only partially observe the system outputs. Based on their observations of the system, the intruder's goal is to construct an estimate of the system's behaviour and determine the secret information the system wishes to keep hidden. In general, the secret that we wish to keep hidden is said to be opaque if the intruder cannot definitely determine if the secret has occurred. More specifically, we consider a system to be opaque if, for any secret behaviour, there exists at least one other non-secret behaviour that looks exactly the same in the eyes of the intruder [30].

In this section, we recall some of the different notions of opacity that can be found in the literature, such as in [5], [58], and present our own examples to illustrate each concept. For DES models, we can divide the concepts of opacity into two broad families:

1. Language-based Opacity.
2. State-based Opacity.

2.3.1 Language-based opacity - LBO

The first notion of opacity that we consider is language-based opacity (LBO). This type of opacity was first introduced in [4] and [18] and has been formalised in different ways in the literature. Here, we discuss a general definition of language-based opacity from [41]. This definition of LBO involves two sublanguages of the system, $L_s, L_{ns} \subseteq \mathcal{L}(G, X_0)$. It is said that a system is language-based opaque if for any string w in the secret language L_s, there exists a string w' in the non-secret language L_{ns} with the same projection, meaning $P(w) = P(w')$. We now recall its formal definition.

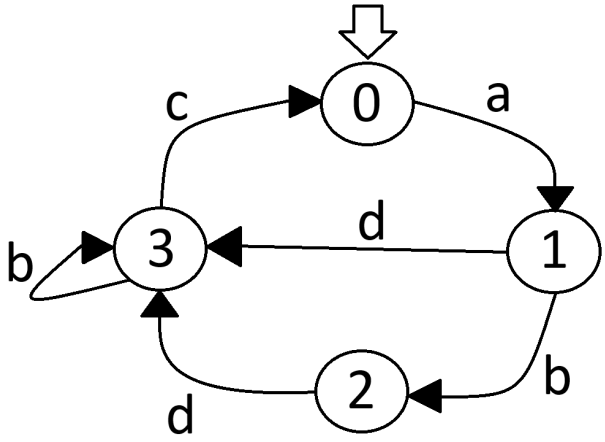
Definition 2.3.1. *(Language-based Opacity [5]). Consider a DFSA G, a projection P, and a secret language and non-secret language L_s and L_{ns} respectively. The system G is language-based opaque if for every string $w \in L_s$, there exists another string $w' \in L_{ns}$ such that $P(w) = P(w')$ or equivalently, $L_s \subseteq P^{-1}[P(L_{ns})]$.*

We illustrate this definition of language-based opacity using the following example.

Example 2.3.1. *Consider the deterministic automaton G that is depicted in the state diagram in Figure 2.2 with $X = \{0, 1, 2, 3\}$, $E = \{a, b, c, d\}$, and $X_0 = \{0\}$. Now let $E_{obs} = \{a, b, c\}$. Then our system G satisfies language-based opacity when $L_s = \{abdc\}$ and $L_{ns} = \{adbc, abdbc\}$. LBO is satisfied due to the fact that whenever the intruder sees the projection $P(L_s) = \{abc\}$, they are not sure if the string abdc or adbc has occurred.*
If we set $L_s = \{abdb\}$ and $L_{ns} = \{abdc, abd, adbbc\}$, then our system does not satisfy LBO as no string in L_{ns} has the same projection as the secret string abdb.

As we mentioned previously, opacity was originally introduced for discrete event systems in [8] using the language-based approach. Although in the rest of this

Figure 2.2: State Diagram System G for Example 2.3.1.



thesis our focus is on state-based approaches for opacity, it is important to note that there is a connection between the two approaches. In [66] and [5], several algorithms are described that can be used to transform a system that satisfies LBO to also satisfy different forms of state-based opacity.

2.3.2 State-based Opacity- SBO

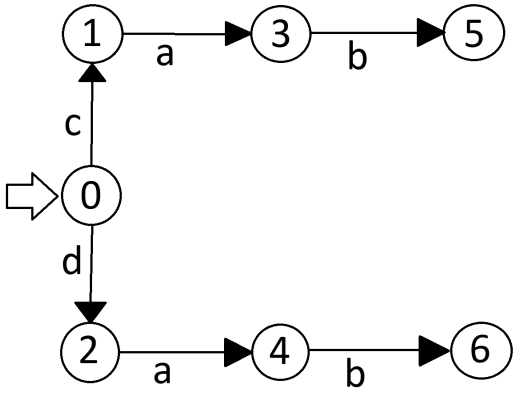
The second notion of opacity we will discuss is a state-based concept. This idea of opacity was first introduced in [8] for Petri nets and has been extended to finite-state automata in [58]. This approach is interested in the intruder's ability to identify if the system is/has been in a given state or set of states we wish to keep secret. Several concepts of state-based opacity have been introduced in the literature [30], [69]. In this thesis, we are interested in the notions of current-state opacity, initial-state opacity, initial-state-final-state opacity, and K-step opacity, which we will discuss below.

Definition 2.3.2. *(Current-State Opacity (CSO) [30]).*
Given a system G, a projection P, a set of states $X_s \subseteq X$, and a set of non-secret states $X_{ns} \subseteq X$, then G is current-state opaque if $\forall x_0 \in X_0$ and $\forall s \in \mathcal{L}(G, x_0)$ such that $f(x_0, s) \in X_s$ there exists $x_0' \in X_0$ and $\bar{s} \in \mathcal{L}(G, x_0')$ such that $f(x_0', \bar{s}) \in X_{ns}$ and $P(s) = P(\bar{s})$.

Loosely speaking, the system G is CSO if for every string of events s that leads to a state we would like to keep secret $x_s \in X_s$, there exists another

string $\bar{s}$ leading to a non-secret state $x_{ns} \in X_{ns}$, such that they have the same projection. This confuses the intruder as they can never assert with certainty that the system's current state belongs to X_s. To demonstrate this, consider the following example.

Figure 2.3: State Diagram of System G for Example 2.3.2



Example 2.3.2. *Consider the DFSA G that is depicted in the state diagram in Figure 2.3. Let us set our secret states and non-secret states as $X_s = \{5\}$ and $X_{ns} = X \backslash X_s$. Also let $X_0 = \{0\}$. If $E_{obs} = \{a, b\}$, then our system G will satisfy current-state opacity. This is because when the strings cab and dab occur, the intruder only observes the string ab such that $P(cab) = P(dab) = ab$. Hence, the intruder does not know with certainty if the system ends in state 5 (the secret state) or state 6 (a non-secret state).*
To give an example of when the system G in Figure 2.3 does not satisfy CSO, consider $E_{obs} = \{c, b\}$, then if the string cab occurs, the intruder knows for certain that the system is in the secret state 5 as the string cab is the only string in G that contains the events c and b.

Example 2.3.2 illustrates the importance of strings being indistinguishable under the projection P in order to satisfy CSO. When the observable event set E_{obs} is chosen so that different strings in the system project to the same observation, the intruder cannot reliably identify whether the system has entered a secret state. However, changing the observable event set can break CSO. In Example 2.3.2 with $E_{obs} = \{c, b\}$, the projection reveals enough information for the intruder to uniquely determine that the system has reached a secret state.

This demonstrates that the observable event set is important for determining whether a system satisfies current-state opacity.

Definition 2.3.3. *(Initial-State Opacity (ISO) [5]).*
Given a system G, a projection P, a set of secret initial states $X_s \subseteq X_0$, and a set of non-secret states $X_{ns} = X_0 \backslash X_s$, G is initial-state opaque if $\forall x_s \in X_s$ and $\forall s \in \mathcal{L}(G, x_s)$, there exists $x_{ns} \in X_{ns}$ and $\bar{s} \in \mathcal{L}(G, x_{ns})$ such that $P(s) = P(\bar{s})$.

A system is initial state opaque if, for every sequence of events s that started from a secret state x_s, there will be another sequence of events $\bar{s}$ that started from a non-secret state x_{ns}, such that s and $\bar{s}$ are observationally equivalent. Hence, an intruder would not be able to determine whether the system started from a state in X_s or not. To illustrate this, consider the example below.

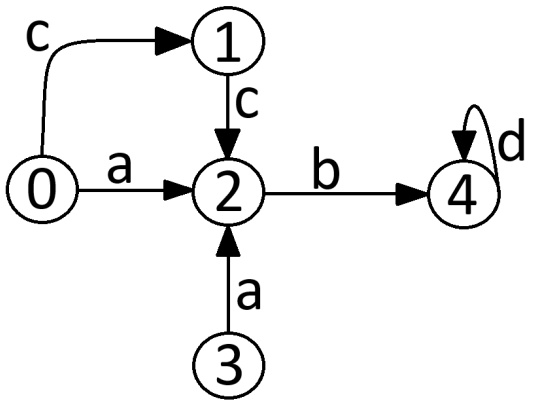
Example 2.3.3. *Consider the automaton G shown in the state diagram in Figure 2.4, with $X_0 = X$. Suppose the set of observable events is $E_{obs} = \{a, b\}$, the set of secret states is $X_s = \{0\}$, and the set of non-secret states is $X_{ns} = X_0 \backslash X_s$. Under these conditions, G satisfies initial-state opacity. This is because for every string that starts at the secret state 0, there exists a string either starting at one of the non-secret initial states 1 or 3 that will give the same projection. For example, consider the string $s = ccbd^*$ starting from the secret state 0. there is a corresponding string $t = cbd^*$ starting for state 1 such that, $P(s) = b = P(t)$. Thus, the observation b does not reveal whether the system started in the secret state or not, satisfying initial-state opacity.*
Let's next consider an example where ISO is not satisfied. If we let $X_s = \{1\}$ and $E_{obs} = \{b, c\}$, then when we see the string cb then the intruder knows the system originated in state 1 as no other initial state can produce the string cb.

Example 2.3.3 illustrates two scenarios in which the system G in Figure 2.4 either satisfies or violates ISO. Notably, the choice of observable events also plays an important role in determining whether ISO holds. As the system may begin at any state, determining what events are observable directly impacted the system's ability to preserve initial-state opacity. We next discuss the concept of initial-and-final-state opacity.

Definition 2.3.4. *(Initial-and-Final-State Opacity (IFO)[66]).*
Given a system G, a projection P, a set of secret pairs $X_{sp} \subseteq X_0 \times X$, and a set of non-secret pairs $X_{nsp} \subseteq X_0 \times X$, G is initial-and-final-state opaque if,

$$\forall (x_0, x_f) \in X_{sp} \ \text{ and } \ \forall s \in \mathcal{L}(G, x_0) \ \text{ such that } \ f(x_0, s) = x_f,$$

Figure 2.4: State Diagram for System G in Example 2.3.3



$$\exists(\bar{x}_0, \bar{x}_f) \in X_{nsp} \;\; \textit{and} \;\; \exists \bar{s} \in \mathcal{L}(G, \bar{x}_0) \;\; \textit{such that} \;\; f(\bar{x}_0, \bar{s}) = \bar{x}_f.$$

$$\textit{and} \;\; P(s) = P(\bar{s}).$$

Given an automaton G, a system satisfies initial-and-final-state opacity if for any string s that starts from the initial state x_0, and ends at state x_f such that the pair $(x_0, x_f) \in X_{sp}$, there exists another string $\bar{s}$ starting from $\bar{x}_0 \in X_0$ and ending at $\bar{x}_f$ with the pair $(\bar{x}_0, x_f) \in X_{nsp}$, such that s and $\bar{s}$ has the same projection.
Note that ISO and CSO are special cases of IFO and can be formalised in this setting. For initial state opacity, we set $X_{sp} = X_s \times X$ and $X_{nsp} = X_{ns} \times X$. Likewise, for current-state opacity, we set $X_{sp} = X_0 \times X_s$ and $X_{nsp} = X_0 \times X_{ns}$. To demonstrate the notion of initial-and-final-state opacity, we use the example below.

Example 2.3.4. *Consider the system G in the state diagram in Figure 2.3. Let our secret pair be $X_{sp} = \{(1,5)\}$ and let $X_{nsp} = \{(2,4),(2,6)\}$. Then, for every string s that starts at state 1 and ends in state 5 there exists a string $\bar{s}$ that started at state 2 and ends at state 6 such that $P(s) = P(\bar{s})$ where $s = ab = \bar{s}$. Hence, the intruder cannot be certain that the system started at state 1 and finished at state 5.*

Except for ISO, the opacity properties discussed so far do not account for the system's behaviour after it exits the secret state. A general problem of interest is ensuring that a previously visited secret state remains unidentifiable by an intruder for a certain number of steps after it has been exited. A property

addressing this issue is called K-step opacity. This was originally introduced in [58]. Here, we recall two notions of K-step opacity and provide an example illustrating each.

Definition 2.3.5. *(K-step (weak) opacity [30])*
Given a system G, a projection P, an integer $K \geq 0$ and sets of secret and non-secret states X_s, X_{ns}, G is K-step (weakly) opaque w.r.t X_s and P (or abbreviated as (X_s, P, K)-(weakly) opaque) if,

$$\forall x_s \in X_0, \forall s \in \mathcal{L}(G, x_s), \quad \textit{and} \quad \forall \bar{s} \in \hat{s} \quad \textit{such that}$$

$$f(x_s, \bar{s}) \in X_s \quad \textit{and} \quad |P(s)/P(\bar{s})| \leq K,$$

$$\exists x_{ns} \in X_0, \exists t \in \mathcal{L}(G, x_{ns}), \quad \textit{and} \quad \exists \bar{t} \in \hat{t} \quad \textit{such that} \quad f(x_{ns}, \bar{t}) \in X_{ns},$$

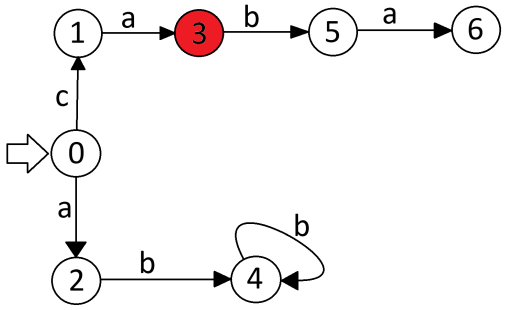
$$P(t) = P(s) \quad \textit{and} \; P(\bar{t}) = P(\bar{s}).$$

A system is (X_s, P, K)-(weakly) opaque if the following holds. For every execution of s of G and for every secret execution $\bar{s}$ prefix of s with an observable difference less than K, there exist two executions t and $\bar{t}$ observationally equivalent respectively to s and $\bar{s}$ such that $\bar{t}$ is not a secret execution. That is, $\bar{t}$ doesn't bring the system to a secret state. From the intruder's point of view, when they observe $P(s)$, they cannot be sure whether a secret state was visited at any point within the last K steps. This is because in our system G, there exists a sequence of events, starting from an initial state that could account for what was seen. We demonstrate this with an example.

Example 2.3.5. *Consider the system G shown in the state diagram in Figure 2.5. If we set $E_{obs} = \{a, b\}$, $X_s = \{3\}$, $X_{ns} = X \backslash X_{ns}$, and $X_0 = \{0\}$. The secret state is shown as a red circle. Here, G is $(X_s, P, 1)$- (weakly) opaque but is not $(X_s, P, 2)$-(weakly) opaque, as if the intruder sees the string aba the only compatible string in the system is caba. Hence after the second 'a' occurs, the intruder will be able to tell that the system was at state 3 two steps before.*

In general, K-step weak opacity is referred to as just *K-step opacity* in the literature. Note that, K-step opacity is a direct extension of CSO, where CSO is equivalent is 0-step opacity [58]. There is a strong version of K-step opacity that was also introduced in [58]. This notion of K-step opacity is referred to as *K-step strong opacity*. A system satisfies K-step strong opacity if it satisfies K-step weak opacity and there exists a trace of the system (sequence of events

Figure 2.5: State Diagram System G for Example 2.3.5.



that are observably equivalent to the actual execution) which does not cross any secret state over the last K steps. We can formalise this with the following definition.

Definition 2.3.6. *(K-step strong opacity [30])*
Given a system G, a projection P and sets of secret and non-secret states X_s, X_{ns}, and an integer $K \geq 0$, G is K-step strongly opaque w.r.t X_s and P (or abbreviated as (X_s, P, K)-strong opaque) if,

$$\forall x_s \in X_0, \forall s \in \mathcal{L}(G, x_s), \exists x_{ns} \in X_0 \text{ and } t \in \mathcal{L}(G, x_{ns}) \text{ such that}$$

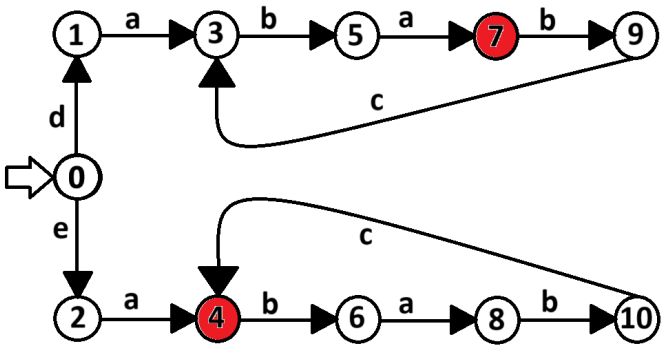
$$P(t) = P(s) \text{ and}$$

$$\forall \bar{t} \in \hat{t}, |P(t)/P(\bar{t})| \leq K$$

$$\Rightarrow \exists x'_{ns} \in X_0 \text{ such that } f(x'_{ns}, \bar{t}) \in X_{ns}.$$

Example 2.3.6. *Consider the DFSA G in Figure 2.6. above. Let $E_{obs} = \{a, b, c\}$, $X_s = \{4, 7\}$ and $X_{ns} = X \backslash X_s$. In Figure 2.6. we indicate our secret states using red circles. It's interesting to note that our system G satisfies (X_s, P, K)-weak opacity for any $K \in \mathbb{N}$. Since events d and e are not observable, for every sequence of events following a visit to state 1, there exists an observationally identical sequence of events starting at state 2.*
Our system G is $(X_s, P, 1)$-strongly opaque. However, our system is not $(X_s, P, 2)$-strong opaque. This is because when the intruder observes the string ababc the system is system is either in state 4 (a secret state) or state 3. This implies that the intruder is currently in a secret state or that it was in the secret state 7 two steps ago. Therefore, from the observation ababc, the system is not $(X_s, P, 2)$-

Figure 2.6: State Diagram System G to Show Strong K-step Opacity in Example 2.3.6.



strong opaque.

2.3.3 Verification of Opacity

Over the past decade or so, extensive work has been done addressing the problem of verifying opacity for discrete event systems. We will not be concerned with verification here, so only very briefly mention some aspects of this work. Verification is concerned with creating methods and algorithms to check if a system satisfies the different notions of opacity. It was shown in [10], [57] and [41] that for many opacity definitions, verification is equivalent to whether or not the system under analysis can admit all possible words constructed on its alphabet. In [41], the author Lin presented algorithms to verify language-based opacity for both deterministic and non-deterministic automata. Lin's algorithm is limited to languages that are regular (meaning languages generated by DFSA) and employs a state-based projection P, which instead, maps the set of states X to sequences of events in E^*. In [59], Saboori and Hadjicostis investigated the verification problem for initial-state opacity in the setting of finite-state deterministic automata. To address this problem, they constructed an initial-state estimator, which constructs all possible initial states of the system that are consistent with the observations of the system's transitions. This approach was later extended in [36] for non-deterministic automata.

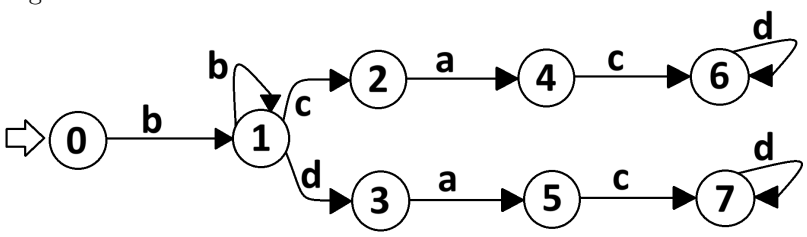
2.4 Enforcement of Opacity

When a system is not opaque, we can use a range of different techniques and algorithms to force a system to satisfy opacity properties. Opacity enforcement does not involve changing the structure of the system. Instead, enforcement of opacity only interferes with the system's output when necessary. Two of the main methods that are used for opacity enforcement are as follows.

1. Deleting events from the output.
2. Adding 'fake' events to change the output.

Other enforcement methods, such as edit functions and delaying observable events, are outlined in [24]. Considering an output trace observed by the intruder, the next event in our system might reveal the secret of the system. The first method aims to avoid this using a device known as a mask, which acts as a filter on the system's observable outputs. The mask selectively removes certain events before they reach the intruder. The mask can restrict the observable outputs of the system in a static or dynamic fashion. A static mask consistently hides a predefined set of events, regardless of system behaviour, while a dynamic mask adapts its actions based on the current system state and the intruder's accumulated knowledge. In the following example, we demonstrate how a mask can be used to enforce CSO in a system.

Figure 2.7: State diagram of system G in Example 2.4.1 to enforce opacity by deleting events.



Example 2.4.1. *Consider the system G in Figure 2.7 where $X_0 = \{0\}$, $X_s = \{4\}$ and $X_{ns} = X/X_{ns}$. Here, we want to make the system G satisfy CSO. If $E_{obs} = E = \{a, b, c, d\}$, then the system is not opaque as bb^*ca leads to the secret state and there are no other strings with the same projection leading to a non-secret state. However, if instead $E_{obs} = \{a\}$ or $E_{obs} = \{b\}$ or even $E_{obs} = \{a, b\}$ then our system satisfies CSO. Thus, we can define static masks*

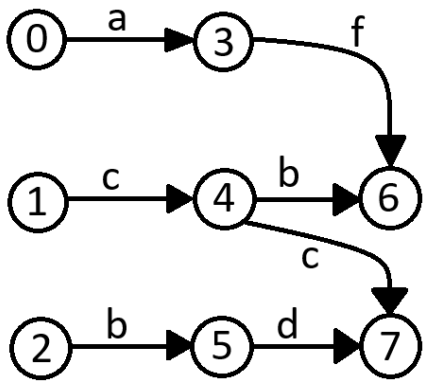
making two events (event c and d) that are permanently unobservable. Although this will make our system opaque, it is also very restrictive. We could hide fewer events, which would make the control less restrictive. In this example, we can be more efficient by using a dynamic mask *that will make an event unobservable only when necessary. Here, if we start from the initial state* 0 *and see the string* bb^**, the intruder knows we are at state* 1 *before splitting off into the two paths. When the event c or d follows, the intruder knows what path we took and reveals the secret state to them. However, we can design a dynamic mask that has the following behaviour.*
The system begins at state 0*, where all the events are observable. When the string* bb^* *occurs, the mask hides the next event to occur (the first occurrence of event c or d is hidden from the intruder) and permits only the events a and b to be observable. Once an event a has been observed, the mask is released and allows all events to be observable again.*

The second enforcement method artificially adds outputs to the set of observed events. This approach uses insertion functions to carry out this task [67]. An insertion function is a monitoring interface at the system's output that changes it by inserting fake events into the observable events. This confuses the intruder as the extra inserted events are indistinguishable from genuine observable events of the system. The intruder, observing the output, which has been altered by the insertion function, cannot tell if the observed string includes inserted events or not. Following the formulation in [31], we defined the insertion function $f_I : E^*_{obs} \times E_{obs} \to E^*_{obs}$ by $f_I(s, e) = es$, where $s \in E^*_{obs}$ and $e \in E_{obs}$. In other words, the insertion function takes an observable string s and inserts the observable event e at the beginning of the string, resulting in the concatenated string es. The following example demonstrates how we can make a system satisfy ISO using this method.

Example 2.4.2. *Consider the system G depicted in Figure 2.8 below, where we would like to enforce initial-state opacity by using an insertion function. Let* $X_0 = \{0, 1, 2\}$ *such that* $X_s = \{2\}$ *and* $X_{ns}\{0, 1\}$ *are the secret and non-secret initial states respectively. Suppose that* $E_{obs} = \{a, b, c\}$*. The system currently does not satisfy ISO, as when the intruder sees the string b, they will know that the system began at the secret state* 2*. Suppose that opacity is enforced by using an insertion function, where* $f_I(b, c) = cb$*. Hence, if the intruder is unaware of the presence or structure of* f_I*, then our system now satisfies ISO. This is because when the string b occurs, after insertion, the intruder will observe the*

string cb, which could have originated from the non-secret initial state 1*. Thus, the intruder cannot be certain whether the system began in a secret state or not.*

Figure 2.8: State diagram for the system G in Example 2.4.2



2.5 Controllability and Observability for DES.

In supervisory control theory (SCT) for discrete-event systems, the notions of controllability and observability play a crucial role in ensuring that a system's behaviour can be guided and monitored to meet desired specifications [9],[63], [51]. The behaviour of such systems is described by their language, and control is exerted by enabling or disabling certain events. When an event is enabled, all transitions labelled by the event are allowed to occur within the system; in contrast, disabled events not permitted to occur within the system. The objective for supervisory control is to restrict the system's behaviour to a desired sublanguage based. Although this is not the main focus of this thesis, the concepts of controllability and observability will play a key role for the system classes in the coming chapters. In particular, we will investigate how these properties relate to opacity.

We first consider controllability. The aim is to enforce a desired behaviour of the system, where this behaviour is represented by a sublanguage $K \subseteq \mathcal{L}(G, X_0)$ of the system's language. The sublanguage K is the control objective. We use the notation $\hat{K}$ to indicate the prefix-closure of the sublanguage K which is defined

as,

$$\hat{K} = \{s \in \Sigma^* | \exists w \in K \text{ such that } sw \in K.\}$$

In other words, $\hat{K}$ is the set of all prefixes of all the strings in K.
The control action is carried out by a supervisor or controller. Typically, the supervisor acts by disabling certain events to ensure the control objective is met. In most practical situations, a supervisor will not have control over all events in E. Let E_C denote the set of controllable events that the supervisor can choose to allow or block, and E_{UC} denote the uncontrollable events that the supervisor cannot disable. We recall the following formal definition of controllability for DES.

Definition 2.5.1. *(Controllability for DES [51])*
Given a DFSA G, a set of controllable events E_C, and a control objective K. We say that K is controllable (with respect to $\mathcal{L}(G, X_0)$ and E_C) if

$$\hat{K}E_{UC} \cap \mathcal{L}(G, X_0) \subseteq \hat{K},$$

where $E_{UC} = E \backslash E_C$.

Loosely speaking, the controllability of K requires that for any prefix $\hat{s}$ of a string in K, if $\hat{s}$ is followed by an uncontrollable event $e \in E_{UC}$ such that $\hat{s}e$ is in $\mathcal{L}(G, X_0)$, then $\hat{s}e$ must also be a prefix of a string in K. It is also known that controllability is preserved under arbitrary unions and consequently the supremal controllable sublanguage of a given language exists and is given by the union of all controllable sublanguages [63].
We next discuss the concept of observability. The intuitive idea is that if we cannot differentiate between two strings based on observations, then the same control action should work for both strings. In terms of the supervisor/controller, this means that if an event e satisfies the control objective for one such string, then it must also do so for the other. The preceding intuition is formalised in the following definition.

Definition 2.5.2. *(Observability for DES [51])*
Given a DFSA G, a set of controllable events E_C, a set of observable events E_{obs}, and a language $K \subseteq \mathcal{L}(G, X_0)$, K is said to be observable (with respect to $\mathcal{L}(G, X_0), E_C, E_{obs}$) if $\forall \hat{s}, \hat{w} \in \hat{K}$ and $\forall e \in E_C$ such that $\hat{s}e \in \mathcal{L}(G, X_0)$, $\hat{w}e \in \hat{K}$ and $P(\hat{s}) = P(\hat{w}), \hat{s}e \in \hat{K}$ holds.

Observability captures the following idea. Suppose that two strings $\hat{s}$ and $\hat{w}$

look the same under observation (same projected string under P), and when a controllable event e occurs after $\hat{w}$, we remain in the desired behaviour $\hat{K}$. Then, we also remain in $\hat{K}$ if the event e occurs after $\hat{s}$. In contrast to controllability, observability is not preserved under unions, and therefore the supremal observable sublanguage of a given language may not exist [63]. In the following examples, we illustrate these two properties.

Example 2.5.1. *Consider the system G depicted in Figure 2.9, which represents a simple automated door control system.*
Here we have three states: the initial state $S0$, where the door is closed, $S1$, where the door is locked, and $S2$, where the door is open. We also have three events of opening, locking, and closing the automated door. We will represent these events as,

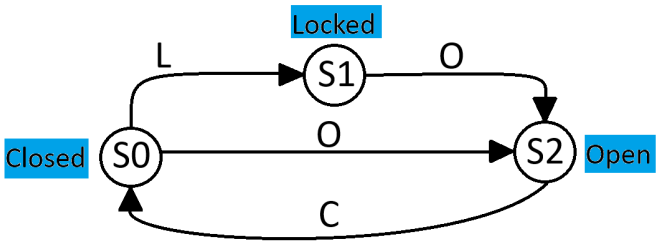
$$E = \{o, c, l\}.$$

We partition our events into controllable and uncontrollable events as,

$$E_C = \{o,\ c\}.$$

$$E_{UC} = \{l\}.$$

Figure 2.9: State Diagram System G depicting the operation of a simple automated door.



Suppose the specification of the system is: 'The door should never lock when the door is closed'. We can define this specification as,

$$K = \{\varepsilon,\ o,\ \ oc,\ oco, \cdots\}.$$

However, when the door is closed and the system is in state $S0$, the event lock can occur, moving the system to state $S1$, even though this is not part of K. This means that K is not controllable.
To see this, first note that K is prefix-closed such that $\hat{K} = K$. Consider the string $s = oc \in \hat{K}$. Then $sl = ocl$ is in $\mathcal{L}(G, X_0)$ but sl is not *in $\hat{K}$, so K is not controllable by Definition 2.5.1.*

Example 2.5.2. *This example now focuses on observability. Consider the same system G in Figure 2.9 with the same controllable and uncontrollable events. Let $E_{obs} = \{o, c\}$, and assume that the specification used to define K' is the following. 'If the door is closed, issue a lock command'. We can define this as,*

$$K' = \{\varepsilon, l, lo, loc, locl, \cdots\}.$$

Here, we have an issue with satisfying the definition of observability. To see this, consider, the strings $\hat{s} = \varepsilon \in \hat{K}'$ and $\hat{w} = l \in \hat{K}'$. Then we have $P(\hat{s}) = P(\hat{w})$. Suppose we consider our controllable event $e = o$. Then we have,

$$\hat{w}e = lo \in \hat{K}$$

and

$$\hat{s}e = o \notin \hat{K}'.$$

Hence, by Definition 2.5.2, our system is not observable.

2.6 Concluding Remarks

In this chapter, we discussed the concept of opacity in its original setting of discrete-event systems. Opacity provided a formal framework to specify and verify whether certain secret states or behaviours can be inferred by an external observer, based on their observations of the system's output. We recalled and formalised various notions of state-based approaches found in the literature. These included initial-state, current-state, initial-and-final-state, and K-step opacity and demonstrated how each of these captured different aspects of protecting secret information from an intruder within a partially observed DES. We briefly mentioned the verification problem associated with LBO and ISO, as discussed in [41] and [59], respectively. We also described some examples of how to enforce opacity by adding/deleting events from the outputs of a system.

Lastly, we described the concept of controllability and observability for discrete event systems, where these concepts will become important to us in the next chapter.

Chapter 3

Opacity Concepts for Linear Systems

3.1 Introduction

In this chapter, we shift our focus from viewing opacity in the DES setting to examining it within the framework of linear systems. This recent line of work is motivated by applications to cyberphysical (CPS) systems that may be modelled by linear time-invariant (LTI) systems. We also make use of tools from control theory to study opacity within this setting. [53], [72].
We will begin the chapter by reviewing LTI systems and their fundamental properties and results. Building on this, we will then review a recently introduced notion of opacity for LTI systems and discuss the concept in terms of reachable sets. Lastly, we will compare opacity with other system properties such as controllability, output controllability, reachability, and observability.

3.2 Linear systems: definition and fundamental results

In this section, we recall some important results and properties for LTI systems that will be used throughout the remainder of the thesis. These concepts are well established in and can be found in works such as [56], [44], and [64].
In this chapter, we will consider discrete-time linear time-invariant (DT-LTI)

systems that take the form:

$$x(t+1) = Ax(t) + Bu(t),$$
$$y(t) = Cx(t), \tag{3.1}$$

where: $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m, y(t) \in \mathbb{R}^p$, and A, B, C, are matrices with $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}$. Time t takes discrete values, where $t \in \mathbb{Z}_+$, the set of nonnegative integers.

The system begins at an initial state $x(0) = x_0 \in X_0$, where $X_0 \subseteq \mathbb{R}^n$ denotes the set of possible initial states. This is often the entire state space $\mathbb{R}^n$. We call $x(t) = [x_1(t), x_2(t), ..., x_n(t)]$ the state vector, and the components $x_i(t)$ are called the state variables. We refer to $u(t)$ as the input vector, where $u(t) = [u_1(t), u_2(t), ..., u_m(t)]$. We use the notation $U(t) = [u(0)^\top, u(1)^\top, \dots u(t)^\top]^\top$ for the vector of inputs up to and including time t. $y(t)$ denotes the output vector, where $y(t) = [y_1(t), y_2(t), ..., y_p(t)]$. Similar to the inputs, we denote the output sequence as $Y(t) = [y(0)^\top, y(1)^\top, \dots, y(t)]^\top$. In the literature, it is quite common to include an extra term in the output equation, leading to $y(t) = Cx(t) + Du(t)$. In this chapter, we shall work with the simpler system model (3.1).

3.2.1 State Equation Solution.

Given an initial state $x(0) = x_0$, and a sequence of inputs $U(t-1)$, we can generate a solution for our system (3.1) in the following way.

At time 1 we have,

$$x(1) = Ax_0 + Bu(0).$$

At time 2,

$$\begin{aligned} x(2) &= A[Ax_0 + Bu(0)] + Bu(1)) \\ &= A^2x_0 + ABu(0) + Bu(1). \end{aligned}$$

at time 3,

$$\begin{aligned} x(3) &= A[A^2x_0 + ABu(0) + Bu(1)] + Bu(2) \\ &= A^3x_0 + A^2Bu(0) + ABu(1) + Bu(2). \end{aligned}$$

$$\vdots$$

From this iteration, it is not difficult to see the pattern and that the state at time t is given by the following expression:

$$x(t) = A^t x_0 + \sum_{i=0}^{t-1} A^{t-i-1} B u(i).$$

It is also clear that for every x_0 and input sequence $U(t-1)$, there is a unique solution defined up to time t. Similarly, the corresponding output $y(t)$ of the system (3.1) can be expressed as

$$y(t) = CA^t x_0 + \sum_{i=0}^{t-1} CA^{t-i-1} B u(i).$$

In terms of notation, we used $x(t, x_0, U(t-1))$ and $y(t, x_0, U(t-1))$ to denote the state and output at time t that originated from the initial state x_0 using the sequence of inputs $U(t-1)$.

3.2.2 Reachability

The system property of reachability refers to the problem of determining whether a system has the ability to transition from one state to another within a finite number of steps. Loosely speaking, a system is considered to be reachable if it can be driven from an initial state to any desired final state by an appropriate choice of inputs. This property concerns the influence of inputs on the state but does not involve the output equation. Reachability in a time interval $[0, t_f]$ is formally defined in the following way.

Definition 3.2.1. *(Reachability [56]).*
The LTI system (3.1) *is said to be reachable on the time interval* $[0, t_f]$ *if given any state* x_f*, there exists an input sequence* $U(t_f - 1)$ *such that, beginning at* 0*, it satisfies* $x(t_f, 0, U(t_f - 1)) = x_f$.

This definition does not require the state to remain at x_f for times $t > t_f$. It also reflects the notion that the inputs can independently influence each of the state variables.
Starting from the initial condition $x_0 = 0$, and applying the input sequence

$U(t_f-1)$, the state at t_f is given by,

$$x(t_f,0,U(t_f-1))=\sum_{i=0}^{t_f-1}A^{t_f-i-1}Bu(i)=[B\ \ AB\ \ A^2B...A^{t_f-1}B]\begin{bmatrix}u(t_f-1)\\u(t_f-2)\\\vdots\\u(0)\end{bmatrix}.$$

We refer to the matrix $[B\ \ AB\ \ A^2B\ \ ...\ \ A^{t_f-1}B]$ as the reachability matrix on $[0,t_f]$. The value $t_f=n$ plays a central role, and the reachability matrix on $[0,n]$ is just denoted $M_{R,n}$. Also, reachability in time n is generally referred to simply as reachability. This is because if a system is reachable in time n, then it is reachable for all times $t\geq n$ also. The following theorem characterises reachability in terms of the rank of $M_{R,n}$.

Theorem 3.2.1. *The LTI system (3.1) is reachable if and only if*

$$rank M_{R,n}=rank[B\ \ AB\ \ A^2B...\ \ A^{n-1}B]=n.$$

In the following example, we illustrate how the rank condition in Theorem 3.2.1 can be applied to verify system reachability.

Example 3.2.1. *Consider the LTI system*

$$x(t+1)=\begin{bmatrix}4&1\\-2&-1\end{bmatrix}x(t)+\begin{bmatrix}1\\2\end{bmatrix}u(t)$$

$$y(t)=\begin{bmatrix}1&0\end{bmatrix}x(t).$$

where,

$$A=\begin{bmatrix}4&1\\-2&-1\end{bmatrix},B=\begin{bmatrix}1\\2\end{bmatrix}\ \ and\ \ C=\begin{bmatrix}1&0\end{bmatrix}.$$

To determine if the system is reachable, we need to check if

$$rank M_{R,2}=rank[B\ \ AB]=n.$$

Here we have that,

$$AB=\begin{bmatrix}4&1\\-2&-1\end{bmatrix}\begin{bmatrix}1\\2\end{bmatrix}=\begin{bmatrix}6\\-4\end{bmatrix}.$$

Hence,

$$M_{R,2} = \begin{bmatrix} 1 & 6 \\ 2 & -4 \end{bmatrix}.$$

Each column of $M_{R,2}$ is linearly independent, so rank $M_{R,2} = 2 = n$. Therefore, by Theorem 3.2.1, the system in this example is reachable.

3.2.3 Controllability

The concept of controllability concerns the ability to drive the system (3.1) from any initial state to the zero state within a finite time interval using an appropriate set of inputs. Controllability is formally defined next.

Definition 3.2.2. *The LTI system (3.1) is controllable on the time interval $[0, t_f]$ if given any initial state, $x(0) = x_0$, there exists an input sequence $U(t_f - 1)$ such that $x(t_f, x_0, U(t_f - 1)) = 0$.*

As with reachability, this definition does not require the system to stay at 0 for any time later than t_f. Also, the case $t_f = n$ is special, and controllability on the interval $[0, n]$ is simply referred to as controllability.
Next, we recall a result that relates controllability to the images of the matrices A^n and $M_{R,n}$ known as *Fuhrmann's rank condition* [44], [45]. We first recall that the image of a general matrix $N \in \mathbb{R}^{l \times p}$ is given by

$$Im(N) = \{Nw | w \in \mathbb{R}^p\}.$$

Theorem 3.2.2. *([44]) The LTI system (3.1) is controllable if and only if,*

$$Im(A^n) \subset Im(M_{R,n})$$

where $M_{R,n}$ is the reachability matrix.

Although controllability and reachability are closely related, it is important to note that for discrete LTI systems, the reachability and controllability properties are not equivalent. If our system (3.1) is reachable, then it implies that our system is controllable. This is true as reachability implies that the image of $M_{R,n}$ equals the state space $\mathbb{R}^n$, so that $Im(A^n) \subset Im(M_{R,n}) = \mathbb{R}^n$. Hence, by Theorem 3.2.2, the system is controllable. The converse doesn't hold however, and we will demonstrate this using a simple example.

Example 3.2.2. *Consider the LTI system:*

$$x(t+1) = \begin{bmatrix} 6 & 24 \\ 8 & 32 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 3 & 4 \end{bmatrix} x(t)$$

where, $A = 0$, $B = \begin{bmatrix} 6 & 24 \\ 8 & 32 \end{bmatrix}$ *and* $C = \begin{bmatrix} 3 & 4 \end{bmatrix}$.

Here $n = 2$. *As rank(B)* $= 1$, *rank*$([B \;\; AB])$ $=$ *rank*$([B \;\; 0])$ < 2. *Hence, the system is not reachable. The system is controllable as* $Im(A^n) \subset Im(M_{R,n})$ *holds trivially because* $Im(A^n) = \{0\}$. *Hence, the system is controllable and not reachable. Note that if A is a full rank matrix, then reachability and controllability are equivalent.*

3.2.4 Observability

We next discuss another system-theoretic property, known as observability. Observability concerns the ability to determine the initial state of a system based on its outputs over time, knowing the inputs. Informally, a system is observable if the system's outputs provide us with enough information to determine the initial state.

Definition 3.2.3. *The LTI system (3.1) is said to be observable on the interval* $[0, t_f]$ *if knowledge of the input sequence* $U(t_f - 1)$ *and output sequence* $Y(t_f)$ *(along with knowing* A, B, C*) is enough to uniquely determine the initial state* x_0.

Unlike controllability and reachability, checking for observability for LTI systems involves examining the outputs in (3.1). Assuming zero input, the outputs of the system can be written in the following way.

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(t_f - 1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{t_f - 1} \end{bmatrix} x_0.$$

Once again, the value $t_f = n$ plays an important role. Observability on $[0, n]$ is referred to as observability, and the matrix,

$$\mathcal{O}_{n-1} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

is referred to as the observability matrix of the system (3.1). The following theorem characterises observability in terms of the observability matrix.

Theorem 3.2.3. *The LTI system (3.1) is observable if and only if,*

$$rank\ \mathcal{O}_{n-1} = rank \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} = n$$

Now we are going to provide an example of using Theorem 3.2.3 to check if an LTI system (3.1) is observable or not.

Example 3.2.3. *Consider the following LTI system,*

$$\begin{aligned} x(t+1) &= \begin{bmatrix} -6 & 0 \\ 0 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 3 \\ 0 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 5 & 6 \end{bmatrix} x(t). \end{aligned}$$

where,

$$A = \begin{bmatrix} -6 & 0 \\ 0 & -2 \end{bmatrix}, B = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \text{ and } C = \begin{bmatrix} 5 & 6 \end{bmatrix}.$$

Using Theorem 3.2.3 to test for observability, we need to check,

$$rank\mathcal{O}_{n-1} = rank \begin{bmatrix} C \\ CA \end{bmatrix} = n$$

where $n = 2$. Here we have that,

$$CA = \begin{bmatrix} 5 & -6 \end{bmatrix} \begin{bmatrix} -6 & 0 \\ 0 & -2 \end{bmatrix} = \begin{bmatrix} -30 & -12 \end{bmatrix}.$$

Hence, we have,

$$\mathcal{O}_{n-1} = \begin{bmatrix} 5 & 6 \\ -30 & -12 \end{bmatrix}$$

and rank $\mathcal{O}_{n-1} = 2 = n$*. Therefore, our system is observable. Note that if we calculated the reachable matrix* $M_{R,n}$ *and checked the reachability condition, we would have,*

$$M_{R,n} = [B \;\; AB] = \begin{bmatrix} -3 & -18 \\ 0 & 0 \end{bmatrix}.$$

where, rank $M_{R,n} = 1 \neq n$*. Hence, our system in this example is observable but not reachable.*

3.3 Opacity for Linear Systems.

We will now introduce the concept of opacity for LTI systems. Here, we are mainly interested in the idea of k-initial state opacity (k-ISO) that was originally presented in [52]. Informally, opacity in this context involves not allowing an intruder who is observing the outputs of the system (3.1) determine whether or not the system started from a secret initial state. To achieve this, for any trajectory starting from a secret initial state, there must exist another trajectory starting from a non-secret state, which looks identical to the intruder. This means when an intruder takes an observation of the outputs of our system, they won't know if the system began at a secret initial state or not. This may be important as knowledge of the system's initial state could enable an intruder to perform targeted attacks on the system [32].

Here, we are going to discuss some fundamental results for k-ISO, and relate them to the system properties discussed previously, as well as the notions of backward-reachable sets, and output controllability. The intruder of the system is assumed to have knowledge of the initial sets of secret and non-secret states, X_s and X_{ns} respectively. Note that the set of initial states X_s and X_{ns} are assumed to be disjoint and to satisfy $X_{ns} = X_0 \backslash X_s$. The intruder also has knowledge of the system model given by A, B, and C. Let $K \subseteq \mathbb{Z}_+$ correspond to the instants of time at which the intruder makes an observation. The intruder's goal is to try to discover, on the basis of observing the outputs of (3.1) at times $k \in K$, whether or not our system started in the set of secret states X_s. The idea of k-initial state opacity is to prevent the intruder from achieving this goal.

Definition 3.3.1. *(Strong k-initial state opacity [52]).*
For the LTI system (3.1)*, given* $X_s, X_{ns} \subseteq X_0$*, and* $k \in K$*,* X_s *is strongly k-initial state opaque (k-ISO) with respect to* X_{ns}*, if for every* $x_s \in X_s$ *and for every sequence of inputs* $U_s(k-1)$*, there exists an* $x_{ns} \in X_{ns}$ *and a sequence of inputs* $U_{ns}(k-1)$ *such that we have,*

$$y(k, x_s, U_s(k-1)) = y(k, x_{ns}, U_{ns}(k-1)).$$

X_s is strongly K-ISO with respect to X_{ns} if X_s is strongly k-ISO with respect to X_{ns}, $\forall\ k \in K$ [53].
Definition 3.3.1 means for any secret state in X_s and sequence of inputs $U_s(k-1)$, when an intruder observes of the output of the system (3.1) at a time $k \in K$, there will be an identical observation from a non-secret initial state by applying a sequence of inputs $U_{ns}(k-1)$. Definition 3.3.1 requires this to hold for every input applied to the initial states in X_s. There is also a weaker version of k-ISO where we relax this condition.

Definition 3.3.2. *(Weak k-initial state opacity [52]).*
For a system (3.1)*, given* $X_s, X_{ns} \subseteq X_0$*, and* $k \in K$*.* X_s *is weakly k-initial state opaque (k-ISO) with respect to* X_{ns}*, if for some* $x_s \in X_s$ *and for some sequence of inputs* $U_s(k-1)$*, there exists an* $x_{ns} \in X_{ns}$ *and a sequence of inputs* $U_{ns}(k-1)$ *such that*

$$y(k, x_s, U_s(k-1)) = y(k, x_{ns}, U_{ns}(k-1)).$$

X_s is weakly K-ISO with respect to X_{ns}, if X_s is weakly k-ISO with respect to X_{ns} $\forall\ k \in K$. In contrast to the strong opacity property, we don't require the outputs to match for every input sequence $U_s(k-1)$ but only require the outputs to match for at least one input sequence $U_s(k-1)$.
These definitions of opacity for LTI systems are related to but different from the familiar definition of observability we have seen in the last section. The observability problem aims to determine the initial state x_0 from its outputs over time. For opacity, the intruder also aims to determine x_0 but only has access to snapshots of the outputs and the set of possible controls. This supports reasoning about intruders with limited observational capabilities. This may be caused by the intruder not wanting to reveal their presence or having limited resources to continuously examine the system's whole behaviour [32].
Staying on the topic of observability, it is important to highlight an aspect of

the definition of K-ISO from [52], which we will revisit later. In this definition, the non-secret state x_{ns} is allowed to be different at different time instants in K. In [32] and [70], they modify this definition of K-ISO and rebrand it as just *initial state opacity*, where x_{ns} is required to be the same across all times when the intruder measures the output. The formulation from [52] described here for opacity in the linear system setting is also different from the definitions of opacity we saw in the DES literature [30], [41], and [24]. In the case of DES, for k-ISO the observation of the entire secret trajectory must coincide with the non-secret trajectory. Here, we only require the secret and non-secret outputs to coincide at time k.

3.4 Comparing k-ISO to Controllability and Observability.

We next explore how the concept of opacity for LTI systems introduced above is related to other system-theoretic properties, such as controllability and observability. First, we will show that if our system is controllable on the time interval $[0, k]$, then there exists a set of secret initial states X_s that satisfies k-ISO with respect to a set of non-secret initial states X_{ns}.

Proposition 3.4.1. *X_s is strongly k-ISO with respect to X_{ns} if our system (3.1) is controllable on the time interval $[0, k]$.*

Proof. Suppose that the LTI system (3.1) is controllable on $[0, k]$. This means that $\forall x_0 \in X_0$, there exists a set of inputs $U(k-1)$ such that $x(k, x_0, U(k-1)) = 0$. Now consider a secret initial $x_s \in X_s$ and an input sequence $U_s(k-1)$. Then

$$x(k, x_s, U_s(k-1)) = A^k x_s + \sum_{i=0}^{k-1} A^{k-i-1} B u_s(i).$$

Pick any non-secret initial state $x_{ns} \in X_{ns}$. As our system is controllable on $[0, k]$ there exists an input sequence $U(k-1)$ such that

$$x(k, x_{ns} - x_s, U(k-1)) = 0.$$

However,

$$x(k, x_{ns} - x_s, U(k-1)) = A^k(x_{ns} - x_s) + \sum_{i=0}^{k-1} A^{k-i-1} B u(i).$$

It follows that if we define $u_{ns}(i) = u(i) + u_s(i)$ for $0 \leq i \leq k-1$, that

$$\begin{aligned} x(k, x_{ns}, U_{ns}(k-1)) &= A^k x_{ns} + \sum_{i=0}^{k-1} A^{k-i-1} B u_{ns}(i) \\ &= A^k x_s + \sum_{i=0}^{k-1} A^{k-i-1} B u_s(i) \\ &= x(k, x_s, U_s(k-1)). \end{aligned}$$

It now follows immediately that

$$y_s(k, x_s, U_s(k-1)) = Cx(k, x_s, U_s(k-1)) = Cx(k, x_{ns}, U_{ns}(k-1)) = y_{ns}(k, x_{ns}, U_{ns}(k-1)).$$

From Definition 3.3.1, X_s is strongly k-ISO with respect to X_{ns}. □

Now we are going to show that the converse does not hold and that a system that is uncontrollable on $[0, k]$ can still satisfy k-ISO.

Example 3.4.1. *Consider the set of secret initial states* $X_s = \{[1 \;\; 1]^\top\}$ *and the non-secret initial states* $X_{ns} = X_0 \backslash X_{ns}$ *for the LTI system*

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 1 & 1 \end{bmatrix} x(t). \end{aligned}$$

where, $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $C = \begin{bmatrix} 1 & 1 \end{bmatrix}$.

We first show that this system is uncontrollable on $[0, 2]$. *To see this, note that*

$$A^2 = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \text{ and } [B \quad AB] = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}.$$

It is now straightforward to see that for any initial state x_0 *with a non-zero second component, there is no input sequence* $u(0)$, $u(1)$ *that will result in* $x(2, x_0, U(1)) = 0$.

We next show that the system satisfies k*-ISO for* $k = 0, 1, 2$. *For* $X_s = \{\begin{bmatrix} 1 \\ 1 \end{bmatrix}\}$ *with the non-secret state* $x_{ns} \in X_{ns}$, *where* $x_{ns} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$, X_s *is strongly* k*-initial*

state opaque as,
for $k = 0$*,*

$$y(0, x_s, \cdot) = Cx_s = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 2$$

and,

$$y(0, x_{ns}, \cdot) = Cx_{ns} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = 2,$$

where the 'dot' in the third argument indicates that the input has no influence on the value of the output at time 0*. Hence,* $y(0, x_s, U_s(-1)) = y(0, x_{ns}, U_{ns}(-1))$*. For* $k = 1$ *we have,*

$$x(1, x_s, U_s(0)) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_s(0) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} u_s(0) \\ 0 \end{bmatrix} = \begin{bmatrix} 2 + u_s(0) \\ 1 \end{bmatrix}$$

and,

$$y(1, x_s, U_s(0)) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 2 + u_s(0) \\ 1 \end{bmatrix} = 3 + u_s(0).$$

The corresponding non-secret state is

$$x(1, x_{ns}, U_{ns}(0)) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} [u_{ns}(0)] = \begin{bmatrix} 2 + u_{ns}(0) \\ 0 \end{bmatrix}$$

so,

$$y(1, x_{ns}, U_{ns}(0)) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 2 + u_{ns}(0) \\ 0 \end{bmatrix} = 2 + u_{ns}(0).$$

Hence, k-ISO holds for $k = 1$ *as we can set* $u_{ns}(0) = u_s(0) + 1$ *to make* $[3 + u_s(0)] = [2 + u_{ns}(0)]$*.*
Now for $k = 2$*,*

$$x(2, x_s, U_s(1)) = A^2 x_s + ABu_s(0) + Bu_s(1) = \begin{bmatrix} 3 \\ 1 \end{bmatrix} + \begin{bmatrix} u_s(0) \\ 0 \end{bmatrix} + \begin{bmatrix} u_s(1) \\ 0 \end{bmatrix} = \begin{bmatrix} 3 + u_s(0) + u_s(1) \\ 1 \end{bmatrix}.$$

Then,

$$y(2, x_s, U_s(1)) = 4 + u_s(0) + u_s(1)$$

For the corresponding non-secret state,

$$x(2, x_{ns}, U_{ns}(1)) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} u_{ns}(0) \\ 0 \end{bmatrix} + \begin{bmatrix} u_{ns}(1) \\ 0 \end{bmatrix} = \begin{bmatrix} 1 + u_{ns}(0) + u_{ns}(1) \\ 1 \end{bmatrix}$$

and the output is

$$y(2, x_{ns}, U_{ns}(1)) = 2 + u_{ns}(0) + u_{ns}(1).$$

Again, k-ISO holds for $k = 2$ as we can set $u_{ns}(0) + u_{ns}(1) = (u_s(0) + u_s(1)) + 2$ to make $[4 + u_s(0) + u_s(1)] = [2 + u_{ns}(0) + u_{ns}(1)]$. So, this system satisfies k-ISO for $k = 0, 1, 2$, but it is not controllable on $[0, 2]$.

Returning to the relation between observability and K-ISO, we want to show that with the application in mind, you would expect that with $K = \{0, 1, 2, ..., k\}$, if the system is observable on $[0, k]$, then it should not satisfy $K-$ISO. In fact, it is desirable that this would hold for any choice of X_s and X_{ns}. Our example below shows that this is not necessarily the case. This highlights the issue with the definition K-ISO from [52] that we mentioned previously.

Example 3.4.2. *Consider the system studied in Example 3.4.1,*

$$x(t+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 1 \end{bmatrix} x(t).$$

We have shown previously that this system satisfies K-ISO for $K = \{0, 1, 2\}$. However, this system is observable on $[0, 2]$. To see this, we apply Theorem 3.2.3.
The observability matrix is given by

$$\mathcal{O}_1 = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}.$$

so that

$$rank(\mathcal{O}_1) = 2 = n.$$

Therefore, the system is observable on $[0, 2]$. This means that the outputs measured at times $0, 1, 2$ uniquely determine the initial state, despite the system

satisfying the definition of K-ISO for $K = \{0, 1, 2\}$. This arises as the definition of K-ISO allows the input sequence used for the non-secret initial state to be different for each time in K.

3.5 Opacity and Reachable States

In the model of opacity used here, the intruder has knowledge of the system model as well as the sets of secret and non-secret initial states. Moreover, the intruder does not have knowledge of the exact control sequence that is applied in the time interval $[0, k]$, but is only aware of input sequences that could be applied to the system. Opacity fundamentally depends on the outputs that can result from a given initial state. Of course, this depends on what state can be reached from a given initial state. With this in mind, it is natural to consider how opacity relates to the reachability properties of a system. Here, we are going to discuss k-ISO in terms of the states that are reachable at time k. This will include establishing conditions for k-ISO to hold in terms of reachable sets and discussing the idea of backward reachable sets and their relation to k-ISO. To do this, we are going to introduce some notation that will help us establish this connection.

Let $\mathcal{U}_{k-1}$ be the set of all input sequences $U(k-1)$. Recall that $x(k, x_0, U(k-1))$ is the state at time k with initial condition x_0 and input sequence $U(k-1)$. We write $R(X_0, k)$ for the set of states that are reachable in k-steps, starting at time 0 from the set X_0. That is,

$$R(X_0, k) = \{x(k, x_0, U(k-1)) : x_0 \in X_0, U(k-1) \in \mathcal{U}_{k-1}\}.$$

For opacity, we are particularly interested in the reachable sets for secret and non-secret initial states $R(X_s, k)$ and $R(X_{ns}, k)$. With this notation, the set of reachable outputs at time k can be written as

$$Y(X_0, k) = \{y : y = Cx, x \in R(X_0, k)\}.$$

The next two theorems characterises k-ISO in terms of reachable outputs defined above with respect to X_s and X_{ns}.

Theorem 3.5.1. *([54]) Consider the LTI system (3.1) with the sets of secret states X_s and non-secret states X_{ns}. The following statements hold.*

1. *X_s is strongly k-ISO with respect to X_{ns} if and only if $Y(X_s, k) \subseteq Y(X_{ns}, k)$.*

2. *X_s is strongly K-ISO with respect to X_{ns} if and only if $Y(X_s, k) \subseteq Y(X_{ns}, k)$ for all $k \in K$.*

It is immediate that $R(X_s, k) \subseteq R(X_{ns}, k)$ is sufficient for X_s to be strongly k-ISO with respect to X_{ns}. However, this is not a necessary condition. The following simple example illustrates this.

Example 3.5.1. *Consider the following LTI system:*

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x(t) \\ y(t) &= \begin{bmatrix} 3 & 3 \end{bmatrix} x(t). \end{aligned}$$

If we consider $X_s = \{[0 \quad 3]^\top\}$ such that $X_{ns} = X_0/X$, where $x_{ns} = [3 \quad 0]^\top \in X_{ns}$. As the A matrix is the identity I_2 and B is zero, then we would have $R(X_s, k) = [0 \quad 3]^\top$ and $Y(X_s, k) = 9$. As $[3 \quad 0]^\top \in X_{ns}$, we have, $R(x_{ns}, k) = [3 \quad 0]^\top$ and $Y(x_{ns}, k) = 9$. Therefore, we have that $Y(X_s, k) \subseteq Y(X_{ns}, k)$ and establishing k-ISO for our system. This shows that $R(X_s, k) \subseteq R(X_{ns}, k)$ is not a necessary condition as here $R(X_s, k) \nsubseteq R(X_{ns}, k)$.

We also have a similar result for weak k-ISO.

Theorem 3.5.2. *([54]) Consider the LTI system (3.1) with the sets of secret states X_s and non-secret states X_{ns}. The following two statements hold.*

1. *X_s is weakly k-ISO with respect to X_{ns} if and only if $Y(X_s, k) \cap Y(X_{ns}, k) \neq \emptyset$.*

2. *X_s is weakly K-ISO with respect to X_{ns} if and only if $Y(X_s, k) \cap Y(X_{ns}, k) \neq \emptyset$, for all $k \in K$.*

Having $R(X_s, k) \cap R(X_{ns}, k) \neq \emptyset$ is a sufficient condition for Theorem 3.5.2 to hold. Again, having this condition $R(X_s, k) \cap R(X_{ns}, k) \neq \emptyset$ is not necessary. We demonstrate this using the following example.

Example 3.5.2. *Consider the system we used in Example 3.5.1, with $X_s = \{[0 \quad 1]^\top, [2 \quad 3]^\top\}$ and $X_{ns} = X_0 \backslash X_s$. Then, $R(X_s, k) = \{[0 \quad 1]^\top, [2 \quad 3]^\top\}$. If we consider the non-secret initial state $x_{ns} = [1 \quad 0]^\top$ such that $R(x_{ns}, k) = [1 \quad 0]^\top$. Then, $Y(X_s, k) = \{3, 15\}$, and clearly $3 \in Y(X_{ns}, k)$. Hence, we have $Y(X_s, k) \cap Y(X_{ns}, k) \neq \emptyset$ but $R(X_s, k) \cap R(X_{ns}, k) = \emptyset$.*

3.6 k-ISO and backwards reachable sets.

In this section, we describe the concept of backward reachable sets for the system (3.1) and explain its connection to k-ISO. Before doing so, recall that the inverse (or pre) image of a set $\mathcal{Y}$ in $\mathbb{R}^p$ under the linear mapping defined by the matrix C is

$$C^{-1}\mathcal{Y} = \{x \in \mathbb{R}^n : Cx \in \mathcal{Y}\}.$$

Definition 3.6.1. *(Backward reachable set [53]) Given a set of states $X_f \subseteq \mathbb{R}^n$. The backward reachable set in k-steps,denoted by $Pre_k(X_f)$, is given by*

$$Pre_k(X_f) = \{x_0 \in \mathbb{R}^n : \exists U(k-1) \in \mathcal{U}_{k-1} \text{ with } x(k, x_0, U(k-1)) \in X_f\}.$$

Informally, the backward reachable set $Pre_k(X_f)$, is the set of initial states from which you can reach a state in X_f at time k by applying a set of inputs $U(k-1) \in \mathcal{U}_{k-1}$.
We are now going to discuss characterising k-ISO in terms of backward reachable sets. In the following result, the thesis adds a slight modification to the one in [53], where it was shown that k-ISO for the system (3.1) is equivalent to the two conditions below.

1. $(Pre_k[C^{-1}(Y(X_s, k))]) \cap R(X_{ns}, k) = \emptyset$
2. $X_s \subseteq Pre_k[C^{-1}(Y(X_{ns}, k))]$

In the following result, we are going to show that we only need the second condition to hold for the system (3.1) to satisfy k-ISO.

Theorem 3.6.1. *X_s satisfies k-ISO with respect to X_{ns} if and only if*

$$X_s \subseteq Pre_k[C^{-1}(Y(X_{ns}, k))].$$

Proof. (If): First let's assume that X_s is k-ISO with respect to X_{ns}. Then by Theorem 3.5.1, we have $Y(X_s, k) \subseteq Y(X_{ns}, k)$. This implies that $R(X_s, k) \subseteq C^{-1}(Y(X_{ns}, k))$. From this, we have,

$$X_s \subseteq Pre_k[C^{-1}(Y(X_{ns}, k))].$$

(Only if): Now let's suppose that $X_s \subseteq Pre_k[C^{-1}(Y(X_{ns}, k))]$. Take $x_s \in X_s$. Then $x_s \in Pre_k[C^{-1}(Y(X_{ns}, k))]$. Hence, there exists an input sequence

$U_s(k-1)$ such that,

$$x(k, x_s, U_s(k-1)) \in C^{-1}(Y(X_{ns}, k)).$$

Then this implies that,

$$Cx(k, x_s, U_s(k-1)) \in Y(X_{ns}, k).$$

Hence we have,

$$Cx(k, x_s, U_s(k-1)) = Cx(k, x_{ns}, U_{ns}(k-1)) \text{ for some } x_{ns} \in X_{ns}, U_{ns}(k-1) \in \mathcal{U}_{k-1}.$$

Now we need to show that this holds for every input sequence applied from x_s. Consider any input sequence $\bar{U}(k-1)$. The linearity of the system (3.1) implies that

$$\begin{aligned} Cx(k, x_s, \bar{U}(k-1)) &= Cx(k, x_s, U_s(k-1)) + Cx(k, 0, \bar{U}(k-1) - U_s(k-1)) \\ &= Cx(k, x_{ns}, U_{ns}(k-1)) + Cx(k, 0, \bar{U}(k-1) - U_s(k-1)) \\ &= Cx(k, x_{ns}, U_{ns}(k-1) + \bar{U}(k-1) - U_s(k-1)). \end{aligned}$$

Hence, we have,

$$Cx(k, x_s, \bar{U}(k-1)) = Cx(k, x_{ns}, U_{ns}(k-1) + \bar{U}(k-1) - U_s(k-1)).$$

As $\bar{U}(k-1)$ is arbitrary, it follows that X_s is k-ISO with respect to X_{ns}. . □

3.7 Output Controllability

A state of the system (3.1) is said to be controllable on the time interval $[0, t_f]$ if we can find an input sequence that transfers the state to 0 in finite time t_f. Now we are going to discuss a similar concept of output controllability that was presented in the works of [52] [54], and [53]. Informally, a state is output controllable on an interval $[0, t_f]$ if we can choose an input sequence such that the output at time t_f is zero.

Definition 3.7.1. *(Output Controllability [53])*
A state x of the system (3.1) is output controllable on the interval $[0, t_f]$ if there exists an input sequence $U(t_f - 1)$ that transfers the system from $x_0 = x$ to

$x(t, x_0, U(t_f - 1))$, *so that* $y(t, x_0, U(t_f - 1)) = 0$.

Note from this definition, we don't require our system's output to remain at zero for time greater than t_f, but only require the output to be zero at that exact time instant t_f. System controllability implies that output controllability also holds for our system, but the converse does not hold. In the next result, we are going to show that controllable implies output controllability.

Theorem 3.7.1. *A state* $x_0 \in \mathbb{R}^n$ *is output controllable on the interval* $[0, t_f]$ *if it is controllable on* $[0, t_f]$.

Proof. Suppose the state x_0 is controllable on the interval $[0, t_f]$. Then there exists a control sequence $U(t_f - 1)$ which drives the state x_0 to $x(t_f, x_0, U(t_f - 1)) = 0$. From our system model (3.1), the output is given by $y(t) = Cx(t)$. Hence, the output corresponding to $x(t_f, x_0, U(t_f - 1))$ is,

$$y(t_f, x_0, U(t_f - 1)) = Cx(t_f, x_0, U(t_f - 1)) = C \times 0 = 0$$

Therefore, x_0 is also output controllable. □

Theorem 3.7.1 simply formalises the observation that if we can control the state to 0 by appropriately selecting inputs, then the output is immediately also controlled to 0 as the state to output map is linear. The following two results describe the relationship between strong k-ISO and output controllability.

Theorem 3.7.2. *([53]) Consider the LTI system* (3.1) *with secret and non-secret states given by* X_s *and* X_{ns} *respectively. Suppose* X_s *is strongly* k*-ISO with respect to* X_{ns}*. Then, there exists a state that is output controllable on the interval* $[0, k]$.

Proof. Let X_s be k-ISO with respect to X_{ns}. From the definition of k-ISO, this implies that for $x_s \in X_s$, $U_s(k-1) \in \mathcal{U}_{k-1}$, there exists $x_{ns} \in X_{ns}$, $U_{ns}(k-1) \in \mathcal{U}_{k-1}$ such that $y(k, x_s, U_s(k-1)) = y(k, x_{ns}, U_{ns}(k-1))$. As X_s and X_{ns} are disjoint, $x_s \neq x_{ns}$. Now, set $x_0 = x_s - x_{ns}$ and $u(j) = u_s(j) - u_{ns}(j)$ for $0 \leq j \leq k-1$. Then consider,

$$CA^k x_s + \sum_{j=0}^{k-1} CA^{k-j-1} B u_s(j) = CA^k x_{ns} + \sum_{j=0}^{k-1} CA^{k-j-1} B u_{ns}(j) \text{ (From } k\text{-ISO)},$$

which implies

$$CA^k x_s - CA^k x_{ns} + \sum_{j=0}^{k-1} CA^{k-j-1} B u_s(j) - \sum_{j=0}^{k-1} CA^{k-j-1} B u_{ns}(j) = 0,$$

Hence by linearity,

$$CA^k(x_s - x_{ns}) + \sum_{j=0}^{k-1} CA^{k-j-1} B(u_s(j) - u_{ns}(j)) = 0.$$

Thus, $y(k, x_0, U(k-1)) = 0$ showing that $x_0 = x_s - x_{ns}$ is output controllable. □

In Theorem 3.7.2, if k-ISO holds for the secret initial state $x_s \in X_s$ with respect to the non-secret initial secret $x_{ns} \in X_{ns}$ with appropriate control sequences $U_s(k-1)$ and $U_{ns}(k-1)$, then the set of inputs $u(i) = u_s(i) - u_{ns}(i)$, where $i = 0, 1, 2, ..., k-1$ will achieve output controllability of the state $x_0 = x_s - x_{ns}$. The next result gives a partial converse to Theorem 3.7.2, and describes conditions under which output controllability implies k-ISO.

Theorem 3.7.3. *([53]) Consider the system (3.1). Let X_{oc} denote the set of states that are output controllable on $[0, k]$, and assume that $X_{oc}\backslash\{0\}$ is non-empty. Let X_s, X_{ns} be sets of secret and non-secret initial states such that every $x_s \in X_s$ can be written as $x + x_{ns}$, where $x \in X_{oc}\backslash\{0\}$, and $x_{ns} \in X_{ns}$. Then X_s is strongly k-ISO with respect to X_{ns}.*

Proof. Let $x_s \in X_s$ be given. Then we know that there exists $x_{ns} \in X_{ns}$ and $x \in X_{oc}\backslash\{0\}$ such that $x_s = x + x_{ns}$.
Output controllability ensures that there exists some input sequence $U(k-1)$ such that

$$CA^k x + \sum_{j=0}^{k-1} CA^{k-j-1} B u(j) = 0. \tag{3.2}$$

For any control input $U_s(k-1)$, the output at time k, starting from $x_s \in X_s$ is given by,

$$y(k, x_s, U_s(k-1)) = CA^k x_s + \sum_{j=0}^{k-1} CA^{k-j-1} B u_s(j)$$

The output at time k starting from $x_{ns} \in X_{ns}$ with the input sequence $\{U_s(k-$

$1) - U(k-1)\}$ is given by,

$$y(k, x_{ns}, U_s(k-1) - U(k-1)) = CA^k x_{ns} + \sum_{j=0}^{k-1} CA^{k-j-1} B(u_s(j) - u(j)).$$

Now, using equation (3.2), we see that

$$\begin{aligned} y(k, x_{ns}, U_s(k-1) - U(k-1)) &= CA^k x_{ns} + \sum_{j=0}^{k-1} CA^{k-j-1} B(u_s(j) - u(j)) \\ &= CA^k (x_s - x) + \sum_{j=0}^{k-1} CA^{k-j-1} B(u_s(j) - u(j)). \end{aligned}$$

$$= CA^k x_s + \sum_{j=0}^{k-1} CA^{k-j-1} B u_s(j) - (CA^k x + \sum_{j=0}^{k-1} CA^{k-j-1} B u(j))$$

$$= CA^k x_s + \sum_{j=0}^{k-1} CA^{k-j-1} B u_s(j) - 0$$

$$= CA^k x_s + \sum_{j=0}^{k-1} CA^{k-j-1} B u_s(j) = y(k, x_s, U_s(k-1))$$

.

Hence, $y(k, x_s, U_s(k-1)) = y(k, x_{ns}, U_{ns}(k-1))$ where $U_{ns}(k-1) = U_s(k-1) - U(k-1)$. As x_s and $U_s(k-1)$ were arbitrary, this shows that X_s is k-ISO with respect to X_{ns} as claimed. □

3.8 Concluding Remarks

In this chapter, we discussed the concept of opacity in the setting of LTI systems. In particular, we looked at the version of k-ISO introduced recently in the works of [52] and linked it to several system properties such as controllability, observability, and output controllability. We also examined k-ISO using backward reachable sets and provided a slight extension of a result in [53]. In the next chapter, we examine attack detection and security for LTI systems and examine the link between opacity and undetectable attacks.

Chapter 4

Security and attack detection for linear systems

4.1 Introduction

Privacy breaches in cyber-physical systems(CPS) can be used maliciously to cause damage to system components and users. Real-world examples of this include the Ukraine power grid attack in 2015 [49] and the Iranian oil terminal attack in 2012 [42]. To help prevent these attacks, extensive research has been conducted on questions related to privacy and security of cyber-physical systems [34] [27] [39]. The concept of opacity discussed in the last chapter is one approach that has been taken to protect privacy. Recall that the goal is to ensure that an intruder or malicious actor cannot definitely determine if the system started from a 'secret' initial state based on the system's output. In this chapter, we turn our attention to the question of security. We will discuss recent work on modeling attacks on linear systems, as well as results and methods to help an external monitor identify if an attack has occurred or not.

We begin by describing the attack model for linear systems and outlining some preliminary results related to it. There are two key actors in the model we study: the *attacker* and the *detector* (following the terminology from Chen et al [11]). The attacker seeks to disrupt the system's operation by injecting an additional attack signal. The aim of the detector is to identify when this happens using input and output measurements. We then discuss the problem of detecting attacks and the related question of characterising undetectable attacks. We

finish the chapter by investigating the connection between undetectable attacks and opacity for linear systems.

4.2 Basic Definitions and Attack Model

To begin, we consider the system model,

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + \bar{B}a(t), \\ y(t) &= Cx(t) + Du(t) + \bar{D}a(t). \end{aligned} \tag{4.1}$$

As before, $t \in \mathbb{Z}_+$ is our time index, the system state $x(t) \in \mathbb{R}^n$, the output $y(t) \in \mathbb{R}^p$, and the input $u(t) \in \mathbb{R}^m$. The signal $a(t) \in \mathbb{R}^s$ is the unknown attack. We make the following remarks to clarify the key assumptions of the attack model we will study here (4.1).

- The detector knows the matrices A, B, C, D, as well as the outputs $y(t)$ and inputs $u(t)$. They do not know $\bar{B}$ and $\bar{D}$ or the attack signal $a(t)$. Also, they do not have full state information: in particular, the detector does not in general know the initial state x_0. The detector's goal is to use the observed input-output behaviour of the system to detect whether an attack $a(t)$ has occurred or not.
- The attacker also knows the system matrices A, B, C, D and the attack matrices $\bar{B}$ and $\bar{D}$. In fact, they design $\bar{B}$ and $\bar{D}$. The matrices $\bar{B}$ and $\bar{D}$ describe the capabilities of the attacker to impact the system's behaviour. The attacker's goal is to inject a non-zero signal $a(t) \neq 0$ to disrupt the system (4.1). Clearly, the attacker wants this attack to be undetected.
- A non-zero attack signal $a(t)$ is *undetectable* if there exist two initial states, which generate the same output sequence, where one is subject to the attack $a(t)$, and the other is not.

The inputs $u(t)$ are known to the detector and they know how the inputs $u(t)$ contribute to the outputs $y(t)$ in the absence of an attack. As in [11], we will work with a modified version of the system (4.1) where we ignore the known inputs $u(t)$. Formally, we consider the system,

$$\bar{\Sigma}: \quad \begin{aligned} x(t+1) &= Ax(t) + \bar{B}a(t) \\ y(t) &= Cx(t) + \bar{D}a(t). \end{aligned} \tag{4.2}$$

We use the notation $\bar{\Sigma} = (A, \bar{B}, C, \bar{D})$ to represent the system model (4.2) and $\Sigma = (A, B, C, D)$ to denote the system with no attack:

$$\begin{aligned} \Sigma: \quad x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t). \end{aligned} \tag{4.3}$$

In Lemma 4.2.1, we justify working with the simpler system (4.2) rather than (4.1).
Consider the attack sequence,

$$E(T) = [a(0)^\top, a(1)^\top, ..., a(T)^\top]^\top,$$

and the corresponding output sequence with initial state x_0,

$$Y(T, x_0, E(T)) = [y(0)^\top, y(1)^\top, ..., y(T)^\top]^\top$$

where $T \geq n-1$. We consider an attack to have occurred on $\bar{\Sigma}$ when $E(T) \neq 0$. Let $Y_1(T, x_0, E(T))$ and $Y(T, x_0, E(T))$ denote the output sequences produced by applying the attack inputs to the initial state x_0 for the systems (4.1) and (4.2) respectively. We can express both output sequences as:

$$Y_1(T, x_0, E(T)) = \mathcal{O}_T x_0 + \mathcal{M}_T U(T) + \bar{\mathcal{M}}_T E(T)$$

and

$$Y(T, x_0, E(T)) = \mathcal{O}_T x_0 + \bar{\mathcal{M}}_T E(T).$$

$\mathcal{O}_T$ is the extended observability matrix,

$$\mathcal{O}_T = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^T \end{bmatrix} \tag{4.4}$$

and $\mathcal{M}_T$ is the input-output matrix of our system, which is given by

$$\mathcal{M}_T = \begin{bmatrix} D & 0 & 0 & ... & 0 \\ CB & D & 0 & ... & 0 \\ CAB & CB & D & ... & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ CA^{T-1}B & CA^{T-2}B & ... & CB & D \end{bmatrix} \tag{4.5}$$

The matrix $\bar{\mathcal{M}}_T$ is defined analogously.

Key Assumptions: Throughout this chapter, we assume that the systems Σ, $\bar{\Sigma}$ are observable, so that the rank of $\mathcal{O}_{n-1}$ is n. Also, we assume that $T \geq n-1$. The next result formally justifies working with the simpler system model (4.2) when considering the problem of attack detection.

Lemma 4.2.1. *An attack sequence $E(T)$ is undetectable for the system (4.1) if and only if it is undetectable for the system (4.2).*

Proof. Consider a non-zero attack sequence $E(T)$. Then $E(T)$ is undetectable for the system (4.1) if and only if there exist two initial states x_0, x_0' such that,

$$\mathcal{O}_T x_0 + \mathcal{M}_T U(T) + \bar{\mathcal{M}}_T E(T) = \mathcal{O}_T x_0' + \mathcal{M}_T U(T),$$

where $U(T)$ is the *known* input sequence. However, by linearity,

$$\begin{aligned} \mathcal{O}_T x_0 + \mathcal{M}_T U(T) + \bar{\mathcal{M}}_T E(T) &= \mathcal{O}_T x_0' + \mathcal{M}_T U(T) \\ \Leftrightarrow \mathcal{O}_T x_0 + \bar{\mathcal{M}}_T E(T) &= \mathcal{O}_T x_0' \end{aligned}$$

which is equivalent to $E(T)$ being undetectable for the system (4.2). □

From now on, we will work with (4.2) as our core system model when considering attack detection.

Another important aspect of the problem concerns *side information* on the initial state. This describes what the detector knows about x_0, and can limit the attacker's ability to remain undetected. The more precise this information, the more difficult it is for an attacker to remain undetected. The side initial state information is given by

$$y_\Gamma = \Gamma x_0 \tag{4.6}$$

where $y_\Gamma \in \mathbb{R}^q$ and $\Gamma \in \mathbb{R}^{q \times n}$. Γ is called the side information matrix. If the matrix Γ has full column rank, y_Γ uniquely determines x_0. On the other

extreme, when Γ is the zero matrix, y_Γ gives us no information about x_0. It is important to note for later that the side information y_Γ cannot be modified by the attacker.
For a system $\bar{\Sigma}$ with side information matrix Γ, an attack $E(T) \neq 0$ is undetectable if there exist initial states x_0, x_0', such that

$$\mathcal{O}_T x_0 + \bar{\mathcal{M}}_T E(T) = \mathcal{O}_T x_0'$$

and $\Gamma x_0 = \Gamma x_0'$. In addition, the initial states must now be consistent in terms of the side initial state information.

4.2.1 The weakly unobservable subspace (WUS).

We now describe the *weakly unobservable subspace*. This will play an important role in studying attack detection and its connection to opacity [11], [33], and [34]. Later in the chapter, we will introduce the strong property of uniform k-ISO, and show that it can be characterised using the WUS. We first define the sequence of weakly unobservable subspaces, and show that this sequence converges in finitely many steps to a fixed subspace. This subspace is the weakly unobservable subspace.

Definition 4.2.1. *(Sequence of weakly unobservable subspaces [64]) For the system* (4.3)*, the sequence of weakly unobservable subspaces* $\mathcal{V}_0$, $\mathcal{V}_1$ $\mathcal{V}_2, \ldots$ *is defined as follows. For* $j = 0, 1, 2, \ldots,$

$\mathcal{V}_j = \{x_0 \in \mathbb{R}^n | \textit{there exists an input sequence U(j) such that } Y(j, x_0, U(j)) = 0\}.$

We can easily show that $\mathcal{V}_j$ is a subspace for all $j \geq 0$. For $j \geq 0$, $\mathcal{V}_j$ satisfies the three properties necessary to define a subspace.

1. If $x_0 = 0$, the output trajectory corresponding to the zero input sequence, $U(j) = 0$, is easily seen to be $Y(j, x_0, U(j)) = \bar{\mathcal{M}}_j U(j) = 0$.

2. **Closed under addition**: If x_1 and x_2 are in $\mathcal{V}_j$, there exist input sequences $U_1(j)$ and $U_2(j)$ such that the corresponding outputs satisfy $Y_1(j, x_1, U_1(j)) = Y_2(j, x_2, U_2(j)) = 0$. The output sequence corresponding to the initial condition $x_1 + x_2$ and the input sequence $U_1(j) + U_2(j)$

is then

$$\begin{aligned} Y(j, x_1 + x_2, U_1(j) + U_2(j)) &= \mathcal{O}_j(x_1 + x_2) + \mathcal{M}_j(U_1(j) + U_2(j)) \\ &= \mathcal{O}_j(x_1) + \mathcal{M}_j E_1(j) + \mathcal{O}_j(x_2) + \mathcal{M}_j E_2(j) \\ &= Y_1(j, x_1, E_1(j)) + Y_2(j, x_2, E_2(j)) = 0. \end{aligned}$$

Hence, $x_1 + x_2 \in \mathcal{V}_j$.

3. **Closed under scalar multiplication**: Now let $x_0 \in \mathcal{V}_j$ and $\alpha \in \mathbb{R}$ be given. Choose $U(j)$ such that the corresponding output sequence $Y(j, x_0, U(j)) = 0$. Then by linearity,

$$\begin{aligned} Y(j, \alpha x_0, \alpha U(j)) &= \mathcal{O}_j(\alpha x_0) + \mathcal{M}_j(\alpha U(j)) \\ &= \alpha[\mathcal{O}_j x_0 + \mathcal{M}_j(U(j))] = \alpha \times 0 = 0. \end{aligned}$$

Hence, $\alpha x_0 \in \mathcal{V}_j$.

We will sometimes refer to the subspace $\mathcal{V}_j$ as the j^{th} weakly unobservable subspace.

We show below that the sequence of subspaces $\mathcal{V}_j$ forms a chain that satisfies the inclusion relation, $\mathcal{V}_0 \supseteq \mathcal{V}_1 \supseteq \mathcal{V}_2 \supseteq \cdots$. The next lemma is important for this and will also be used later in the chapter when we relate attack detection to the concept of *uniform opacity*.

Lemma 4.2.2. *$x_0 \in \mathcal{V}_{j+1}$ if and only if there is an input $u(0) \in \mathbb{R}^s$ such that*

$$\begin{bmatrix} A \\ C \end{bmatrix} x_0 + \begin{bmatrix} B \\ D \end{bmatrix} u(0) \in \mathcal{V}_j \times \{0\}.$$

where $\{0\} \in \mathbb{R}^p$.

Proof. (If): Suppose that $x_0 \in \mathcal{V}_{j+1}$. Then, by Definition 4.2.1, there exists an input sequence $U(j+1) = [u(0), u(1), ..., u(j+1)]$ such that the output $Y(j+1, x_0, U(j+1)) = 0$. In particular,

$$y(0) = Cx_0 + Du(0) = 0,$$

$$x(1) = Ax_0 + Bu(0).$$

Since the rest of the sequence $[u(1), ..., u(j+1)]$ also ensures that the outputs $y(1) = \cdots = y(j+1) = 0$, and starts from the state $x(1)$, it follows that

$x(1) \in \mathcal{V}_j$ and that,

$$\begin{bmatrix} A \\ C \end{bmatrix} x_0 + \begin{bmatrix} B \\ D \end{bmatrix} u(0) = \begin{bmatrix} x(1) \\ y(0) \end{bmatrix} \in \mathcal{V}_j \times \{0\}.$$

(Only if): Suppose there exists a $u(0) \in \mathbb{R}^s$ such that,

$$\begin{bmatrix} A \\ C \end{bmatrix} x_0 + \begin{bmatrix} B \\ D \end{bmatrix} u(0) \in \mathcal{V}_j \times \{0\}.$$

Then if we apply $u(0)$ as an input at time 0,

$$x(1) = Ax_0 + Bu(0),$$

$$y(0) = Cx_0 + Du(0) = 0.$$

By the assumption on x_0, $u(0)$, it follows that $x(1) \in \mathcal{V}_j$. Hence there exists an input sequence $[u(1), ..., u(j+1)]$ such that starting from $x(1)$, the corresponding output sequence $[y(1), \ldots, y(j+1)]$ is zero. Then, inserting the input $u(0)$ to define $U(j+1) = [u(0), u(1), ..., u(j+1)]$, $Y(j+1, x_0, U(j+1)) = 0$. This implies that $x_0 \in \mathcal{V}_{j+1}$. □

Lemma 4.2.2 implies the following recurrence relation for the sequence $\{\mathcal{V}_j\}$.

$$\mathcal{V}_{j+1} = \begin{bmatrix} A \\ C \end{bmatrix}^{-1} (\mathcal{V}_j \times \{0\} + \operatorname{Im} \begin{bmatrix} B \\ D \end{bmatrix}) \tag{4.7}$$

We will use this to show in Proposition 4.2.1 that there is some integer k such that $\mathcal{V}_k = \mathcal{V}_{k+1}$. Thus, the inclusion chain for $\mathcal{V}_j$ has the form,

$$\mathcal{V}_0 \supset \mathcal{V}_1 \supset \mathcal{V}_2 \supset ... \supset \mathcal{V}_k = \mathcal{V}_{k+1} = \mathcal{V}_{k+2} = ..., \text{for some integer } \ k \leq n-1. \tag{4.8}$$

Proposition 4.2.1. *Let $\mathcal{V}_j$, for $j = 0, 1, 2, ...$ be the sequence of weakly unobservable subspaces for the system (4.3). Then there exists $k \leq n$ such that $\mathcal{V}_k = \mathcal{V}_{k+1}$. Furthermore, $\mathcal{V}_k = \mathcal{V}_j$ for all $j \geq k$.*

Proof. First, we prove that the sequence $\{\mathcal{V}_j\}_{j \in \mathbb{N}}$ is not increasing with respect to set inclusion. To show this, we use induction.
Base case: Suppose that $x_0 \in \mathcal{V}_1$. By definition, there exists an input sequence $U(1)$ such that $Y(1, x_0, U(1)) = 0$. This means that the outputs $y(0) = 0$ and

$y(1) = 0$. In particular, $y(0) = 0$, so $x_0 \in \mathcal{V}_0$. Hence,

$$\mathcal{V}_0 \supseteq \mathcal{V}_1.$$

Induction Step: Next suppose that

$$\mathcal{V}_{j-1} \supseteq \mathcal{V}_j$$

for some $j \geq 1$. This implies that

$$(\mathcal{V}_{j-1} \times \{0\} + \text{Im} \begin{bmatrix} B \\ D \end{bmatrix}) \supseteq (\mathcal{V}_j \times \{0\} + \text{Im} \begin{bmatrix} B \\ D \end{bmatrix})$$

.
Then, using (4.7) it follows that

$$\mathcal{V}_j = \begin{bmatrix} A \\ C \end{bmatrix}^{-1} (\mathcal{V}_{j-1} \times \{0\} + \text{Im} \begin{bmatrix} B \\ D \end{bmatrix}) \supseteq \begin{bmatrix} A \\ C \end{bmatrix}^{-1} (\mathcal{V}_j \times \{0\} + \text{Im} \begin{bmatrix} B \\ D \end{bmatrix}) = \mathcal{V}_{j+1}$$

Note that the same argument shows that if $\mathcal{V}_j = \mathcal{V}_{j-1}$, then $\mathcal{V}_{j+1} = \mathcal{V}_j = \mathcal{V}_{j-1}$. Hence, $\{\mathcal{V}_j\}_{j \in \mathbb{N}}$ is a descending chain of subspaces of $\mathbb{R}^n$. This means that for each $j \geq 1$, either $\mathcal{V}_j = \mathcal{V}_{j-1}$ or the dimension of $\mathcal{V}_j$ is strictly less than the dimension of $\mathcal{V}_{j-1}$. Since $\mathbb{R}^n$ is finite-dimensional, it follows that this sequence must terminate in finite time.
It is not hard to see that if $\mathcal{V}_0 = \mathbb{R}^n$, then $\mathcal{V}_j = \mathbb{R}^n$ for all $j \geq 0$, so the conclusion that $k \leq n-1$ is trivial in this case. Otherwise, the dimension of $\mathcal{V}_0$ is at most $n-1$, and again we can conclude that there exists $k \leq n-1$ such that $\mathcal{V}_k = \mathcal{V}_{k+1}$. Finally, as noted above, it now follows from the recurrence (4.7) that:

$$\mathcal{V}_{k+2} = \mathcal{V}_{k+1} = \mathcal{V}_k.$$

Iterating, this implies that $\mathcal{V}_j = \mathcal{V}_k$ for all $j \geq k$. □

Proposition 4.2.1 shows that the WUS sequence converges within k steps to a fixed subspace $\mathcal{V}_k$. Note that if x_0 is in this subspace $\mathcal{V}_k$, then it is in $\mathcal{V}_j$ for all $j \geq 0$. This subspace is typically just referred to as the weakly unobservable subspace of the system Σ, which we formally define below.

Definition 4.2.2. *(Weakly unobservable subspace [64], [65]): The weakly unobservable subspace of a system Σ, $\mathcal{V}(\Sigma)$, is the subspace of all $x_0 \in \mathbb{R}^n$ such*

that for every $j \geq 0$, there exists an input sequence $U(j)$ such that the output trajectory satisfies $Y(j, x_0, U(j)) = 0$.

The subspace $\mathcal{V}(\Sigma)$ of the system is fundamentally connected to undetectable attacks. In particular, we shall see later in this chapter that if $\mathcal{V}(\Sigma) \neq 0$ then there exists an undetectable attack $E(T)$ for some related attack system $\bar{\Sigma}$ [34], [48].

4.3 Dynamic Attack Detection

We now return to considering attacks over the time window $0, 1, \ldots, T$. A dynamic attack detector analyses the system output over this window and the side initial information y_Γ of Σ and determines if an attack has happened or not. Formally, we define the detector as a mapping ψ, which takes the output sequence and side initial information as inputs, and outputs one of the values 'Attack', 'No Attack'. Mathematically,

$$\psi : \mathbb{R}^{p(T+1)} \times \mathbb{R}^q \to \{\text{Attack, No Attack}\}.$$

We assume that the detector ψ has full knowledge of the matrices A and C but does not know the matrices $\bar{B}$ and $\bar{D}$ in (4.2). The attack detector ψ also doesn't know the initial state x_0 but knows the matrix Γ as specified in (4.6). An attack is detectable by ψ if ψ takes the value 'Attack' when it occurs. We now introduce two desirable properties for the mapping ψ: namely, consistency and soundness for attack detectors [11].

Definition 4.3.1. *(Consistent attack detector [11])*
An attack detector ψ is said to be consistent if $\psi(\mathcal{O}_T x_0, \Gamma x_0) =$ No Attack for all $x_0 \in \mathbb{R}^n$.

The detector ψ is consistent if, for any possible initial state $x_0 \in \mathbb{R}^n$, it declares 'No Attack' when given the output sequence $\mathcal{O}_T x_0$ and the side information Γx_0 corresponding to what would be seen under normal (i.e. attack-free) conditions. A consistent detector ensures that we do not mistakenly identify genuine system behaviour as malicious interference. Consistent attack detectors do not create false alarms.

Definition 4.3.2. *(Sound attack detector [11])*

A consistent attack detector ψ is sound if $\psi(Y(T, x_0, E(T)), y_\Gamma) = $ No Attack for some $Y(T, x_0, E(T))$ and y_Γ,then, for any other consistent detector $\bar{\psi}$, $\bar{\psi}(Y(T, x_0, E(T)), y_\Gamma) = $ No Attack.

A sound attack detector is consistent by definition. Therefore, it doesn't create false alarms. Moreover, if a detector is sound, then it detects all attacks that can be detected by a consistent detector. It is worth noting that in [11], [33], an alternative definition of undetectable attacks is suggested, where an attack is deemed undetectable if for every consistent detector, there is some initial state for which the detector will return *No Attack*.
Lemma 4.3.1 provides a necessary and sufficient condition for an attack $E(T)$ to be undetectable in the case where there is no side information (i.e. when $\Gamma = 0$).

Lemma 4.3.1. *([48]): Consider the system given by* (4.2) *and assume that* $\Gamma = 0$. *An attack $E(T)$ is undetectable if and only if*

$$\mathcal{O}_T x_0 + \bar{\mathcal{M}}_T E(T) = \mathcal{O}_T x_0'$$

for some initial states x_0 and $x_0' \in \mathbb{R}^n$.

Lemma 4.3.1 states that an attack $E(T)$ is undetectable if and only if the output sequence produced by this attack starting from x_0 is equal to an output sequence produced by no attack starting from another initial state x_0'. The following example shows how Lemma 4.3.1 can be used to construct an undetectable attack on a system.

Example 4.3.1. *Consider the following system,*

$$x(t+1) = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} a(t)$$

$$y(t) = \begin{bmatrix} 1 & -2 \end{bmatrix} x(t) + 2a(t)$$

where $A = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \bar{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & -2 \end{bmatrix}, \bar{D} = 2$. Note that $n = 2$ in this case and that,

$$\bar{\mathcal{M}}_1 = \begin{bmatrix} D & 0 \\ CB & D \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ -1 & 2 \end{bmatrix} \text{ and } \mathcal{O}_1 = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & -2 \\ 3 & -2. \end{bmatrix}$$

Hence, Lemma 4.3.1 implies that the condition for an attack to be undetectable is

$$\mathcal{O}_1 x_0 + \bar{\mathcal{M}}_1 E(1) = \mathcal{O}_1 x_0'$$

for some x_0 and x_0' in $\mathbb{R}^2$. We can rewrite this as

$$\mathcal{O}_1(x_0' - x_0) = \bar{\mathcal{M}}_1 E(1),$$

or equivalently

$$\begin{bmatrix} C \\ CA \end{bmatrix} (x_0' - x_0) = \begin{bmatrix} D & 0 \\ CB & D \end{bmatrix} \begin{bmatrix} a(0) \\ a(1) \end{bmatrix}.$$

If we let $z = x_0' - x_0$, the condition for an undetectable attack is that there is a solution to

$$\begin{bmatrix} 1 & -2 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} a(0) \\ a(1) \end{bmatrix}$$

Multiplying out both sides of this yields

$$\begin{bmatrix} z_1 - 2z_2 \\ 3z_1 - 2z_2 \end{bmatrix} = \begin{bmatrix} 2a(0) \\ -a(0) + 2a(1) \end{bmatrix}.$$

This system has many solutions. For instance, let $z_1 = 1$ and $z_2 = 0$, so that we need to solve

$$\begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 2a(0) \\ -a(0) + 2a(1) \end{bmatrix}.$$

This is solved by $a(0) = 0.5$ and $a(1) = 1.75$. Therefore, by Lemma 4.3.1 the attack $E(1) = [0.5 \quad 1.75]^\top$ is undetectable for our system in this example.
For the initial conditions $x_0 = [1 \;\; 2]^\top$ and $x_0' = [2 \;\; 2]^\top$, we have,

$$\begin{aligned} \mathcal{O}_1 x_0 + \bar{\mathcal{M}}_1 E(1) &= \mathcal{O}_1 x_0' \\ \begin{bmatrix} 1 & -2 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 0.5 \\ 1.75 \end{bmatrix} &= \begin{bmatrix} 1 & -2 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} \\ \begin{bmatrix} -3 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 3 \end{bmatrix} &= \begin{bmatrix} -2 \\ 2 \end{bmatrix} \\ \begin{bmatrix} -2 \\ 2 \end{bmatrix} &= \begin{bmatrix} -2 \\ 2 \end{bmatrix} \end{aligned}$$

Corollary 4.3.1. *(No initial state information $\Gamma = 0$ [11]) Consider the system given by* (4.2) *and assume that $\Gamma = 0$. An attack $E(T)$ is undetectable if and*

only if there exist $x_0, x_0' \in \mathbb{R}^n$ *such that* $\bar{\mathcal{M}}_T E(T) = -\mathcal{O}_T\theta$, *where* $\theta = x_0 - x_0'$ *is in* $\mathcal{V}(\bar{\Sigma})$.

Proof. This corollary follows from Lemma 4.3.1. Suppose an attack $E(T)$ is undetectable to our system $\bar{\Sigma}$. Then this is equivalent to,

$$\mathcal{O}_T x_0 + \bar{\mathcal{M}}_T E(T) = \mathcal{O}_T x_0'. \text{ (from Lemma 4.3.1)}$$

Then we have,

$$\bar{\mathcal{M}}_T E(T) = \mathcal{O}_T x_0' - \mathcal{O}_T x_0.$$

By linearity, we can write,

$$\bar{\mathcal{M}}_T E(T) = -\mathcal{O}_T[x_0 - x_0'].$$

As $\theta = x_0 - x_0'$, $\bar{\mathcal{M}}_T E(T) = \mathcal{O}_T\theta$. Clearly, as $T \geq n-1$, $\theta \in \mathcal{V}(\bar{\Sigma})$. □

Now we are going to discuss two of the main results in the paper by Chen et al [11]. The first result describes necessary conditions for an attack to be undetectable when the attack detector has side initial state information y_Γ. We will use $\mathcal{N}(\Gamma)$ to denote the null space of Γ:

$$\mathcal{N}(\Gamma) = \{x \in \mathbb{R}^n : \Gamma x = 0\}$$

The next Lemma will help us prove Theorem 4.3.1 below.

Lemma 4.3.2. *An attack* $E(T)$ *against the system* $\bar{\Sigma}$ *is undetectable if and only if* $\mathcal{O}_T x_0 + \bar{\mathcal{M}}_T E(T) = \mathcal{O}_T x_0'$ *and* $\Gamma x_0 = \Gamma x_0'$ *for initial states* $x_0, x_0' \in \mathbb{R}^n$.

Theorem 4.3.1. *(Undetectable Attacks With Side Initial State Information) An attack* $E(T)$ *is undetectable if and only if there exists* $\theta \in \mathcal{N}(\Gamma) \cap \mathcal{V}(\bar{\Sigma})$ *for which* $\bar{\mathcal{M}}_T E(T) = -\mathcal{O}_T\theta$.

Proof. (if): Suppose a system $\bar{\Sigma} = (A, \bar{B}, C, \bar{D})$ is equipped with an attack detector that has side information matrix Γ. Let $x_0 \in \mathbb{R}^n$ and $E(T)$ be an attack such that $\bar{\mathcal{M}}_T E(T) = \mathcal{O}_T\theta$ for $\theta \in \mathcal{N}(\Gamma) \cap \mathcal{V}(\bar{\Sigma})$. Consider $x_0' = x_0 - \theta$ so $\theta = -x_0' + x_0$. As $\bar{\mathcal{M}}_T E(T) = -\mathcal{O}_T\theta$, this implies

$$\bar{\mathcal{M}}_T E(T) = -\mathcal{O}_T(-x_0' + x_0)$$

which can be rearranged to give

$$\bar{\mathcal{M}}_T E(T) + \mathcal{O}_T x_0 = \mathcal{O}_T x_0'$$

In addition, $\theta \in N(\Gamma)$ (as $\theta \in N(\Gamma) \cap \mathcal{V}(\bar{\Sigma})$) so that $\Gamma x_0' = \Gamma(x_0 - \theta) = \Gamma(x_0) - \Gamma(\theta) = \Gamma x_0$. Thus, for any x_0, there exists x_0' such that $\bar{\mathcal{M}}_T E(T) + \mathcal{O}_T x_0 = \mathcal{O}_T x_0'$ and $\Gamma x_0 = \Gamma x_0'$. By Lemma 4.3.2, the attack $E(T)$ is an undetectable attack.
(Only if): Now assume that $E(T)$ is an undetectable attack. From Lemma 4.3.2, we know $\bar{\mathcal{M}}_T E(T) = -\mathcal{O}_T(x_0 - x_0')$ holds and $\Gamma x_0 = \Gamma x_0'$. Note that, $\Gamma x_0 = \Gamma x_0'$ implies that $\Gamma\theta = 0$ for $\theta = x_0 - x_0'$. It follows that $\theta \in \mathcal{N}(\Gamma)$. As $\bar{\mathcal{M}}_T E(T) + \mathcal{O}_T \theta = 0$, it follows that $\theta \in \mathcal{N}(\Gamma) \cap \mathcal{V}(\Sigma)$.

□

An attack $E(T)$ is undetectable over the time interval $\{0, 1, \ldots, T\}$ if and only if the attack's contribution to the output, $\mathcal{M}_T E_T$, exactly cancels out the output that would result from the system starting from some initial state θ that lies in both the null space of the side information matrix Γ and the weakly unobservable subspace $\mathcal{V}(\Sigma)$. We refer to θ as the attack-induced state.

4.3.1 Detector design

After introducing and characterising undetectable attacks, the authors of [11] considered the problem of designing *detectors*. We next discuss their approach to this problem. Specifically, we describe their design of a consistent detector that can detect all attacks that do not belong to the set of undetectable attacks. To begin, we choose a positive integer l. At every time k, the detector records the new output $y(k)$ and decides if the system was attacked in the time period up to and including time k. The detector only uses the most recent l measurements: $y(k-l+1), \ldots, y(k-1), y(k)$ for each decision. The authors argue that this can be more efficient than scanning the entire history of measurements $y(0), y(1), \ldots, y(k)$ to detect attacks.
Define $\bar{Y}(k)$ as the l-length window of output measurements ending at time k, where $k \geq l-1$ such that we have,

$$\bar{Y}(k) = [y(k-l+1)^\top, y(k-l+2)^\top =, y(k-l+3)^\top, ..., y(k)^\top]. \tag{4.9}$$

The aim is to design an attack detector that makes a decision at each time

$k \geq l-1$, and outputs either 'Attack' or 'No Attack'. The input to the detector is denoted by $\hat{Y}(k)$ and is defined in the following way.

$$\hat{Y}(k) = \begin{cases} \begin{bmatrix} y_\Gamma \\ \bar{Y}(k) \end{bmatrix} & \text{for } k = l-1 \\ \bar{Y}(k) & \text{for } k = l, l+1... \end{cases} \tag{4.10}$$

Note that $\hat{Y}(l-1)$ includes the side information on the initial state, while $\hat{Y}(k)$ for $k \geq l$, only includes the l most recent output measurements. The definition for $k = l-1$ is different as that is the only value of k where the initial state x_0 appears explicitly in the measurement window $\bar{Y}(k)$.
The image space of the following matrix $\mathcal{K}(k)$ is important in defining the detector and characterising its behaviour.

$$\mathcal{K}(k) = \begin{cases} \begin{bmatrix} \Gamma \\ \mathcal{O}_{l-1} \end{bmatrix} & \text{for} \quad k = l-1 \\ \mathcal{O}_{l-1} & \text{for} \quad k = l, l+1... \end{cases} \tag{4.11}$$

To construct the detector, the authors define the mapping ψ by

$$\psi(\hat{Y}(k)) = \begin{cases} \text{No Attack}, & \text{if } \hat{Y}(k) \in \text{Im}(\mathcal{K}(k)), \\ \text{Attack} & \textit{Otherwise.} \end{cases} \tag{4.12}$$

Note that $\psi(\hat{Y}(k))$ returns 'No Attack' when the input $\hat{Y}(k)$ lies in the image of $\mathcal{K}(k)$.
Finally, for a time $T \geq l-1$, the overall window-based detector returns 'No Attack' for the time interval $0, 1, ..., T$ if

$$\psi(\hat{Y}(l-1)) = \psi(\hat{Y}(l)) = ... = \psi(\hat{Y}(T)) = \text{No Attack}.$$

Theorem 4.3.2 characterises when this detector will return 'No Attack', and shows that it is both consistent and sound. Here, we present a sharper formulation of the result in [11]. We begin with the following lemma that is useful for the proof of theorem.

Lemma 4.3.3. *Consider the system $\bar{\Sigma}$ and let $l \geq n+1$, $k \geq l-1$. Suppose that*

$$\bar{Y}(k) = \mathcal{O}_{l-1}x, \ \bar{Y}(k+1) = \mathcal{O}_{l-1}x'$$

for x, x' in $\mathbb{R}^n$. Then $x' = Ax$.

Proof. $\bar{\Sigma}$ is observable, which implies that $\mathcal{O}_{n-1}$ has rank n. Moreover, as $l \geq n+1$, $l-2 \geq n-1$ and we can conclude that $\mathcal{O}_{l-2}$ also has rank n. We are given that

$$\bar{Y}(k) = \begin{bmatrix} y(k-l+1) \\ y(k-l+2) \\ \vdots \\ y(k) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{l-1} \end{bmatrix} x, \quad \bar{Y}(k+1) = \begin{bmatrix} y(k-l+2) \\ y(k-l+3) \\ \vdots \\ y(k+1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{l-1} \end{bmatrix} x'.$$

Choosing the last $l-2$ entries from the first of these, and the first $l-2$ from the second, we see that

$$\begin{bmatrix} y(k-l+2) \\ y(k-l+3) \\ \vdots \\ y(k) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{l-2} \end{bmatrix} Ax = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{l-2} \end{bmatrix} x'.$$

Thus, $\mathcal{O}_{l-2}Ax = \mathcal{O}_{l-2}x'$. As $\mathcal{O}_{l-2}$ has full column rank ($= n$), this implies that $Ax = x'$ as claimed. □

Theorem 4.3.2. *([11]) Consider the system $\bar{\Sigma}$ and let $T \geq l \geq n+1$, where n is the dimension of the system state space. Then $\psi(\hat{Y}(l-1)) = \psi(\hat{Y}(l)) = ... = \psi(\hat{Y}(T))$ = No Attack if and only if there is some $x \in \mathbb{R}^n$ with $Y(T) = \mathcal{O}_T x$ and $y_\Gamma = \Gamma x$.*

Proof. First, suppose that there exists $x \in \mathbb{R}^n$ with $Y(T) = \mathcal{O}_T x$ and $y_\Gamma = \Gamma x$. We need to show that $\hat{Y}(k)$ lies in the image of $\mathcal{K}(k)$ for $l-1 \leq k \leq T$.
We will separately consider the cases $k = l-1$ and $k > l-1$.
Using the definition of $\hat{Y}(k)$ in (4.10) we have,

$$\hat{Y}(l-1) = \begin{bmatrix} y_\Gamma \\ \bar{Y}(l-1) \end{bmatrix}.$$

By assumption, we have that $Y(T) = \mathcal{O}_T x$. Also, $\bar{Y}(l-1) = Y(l-1)$, so we can rewrite $\hat{Y}(l-1)$ as,

$$\hat{Y}(l-1) = \begin{bmatrix} \Gamma \\ \mathcal{O}_{l-1} \end{bmatrix} x. \tag{4.13}$$

This shows that $\hat{Y}(l-1)$ lies in the image of $\mathcal{K}(l-1)$, so $\psi(\hat{Y}(l-1)) =$ No Attack.
Now, consider the case $l \leq k \leq T$. From the definition of $\hat{Y}(k)$ in (4.10), $\hat{Y}(k) = \bar{Y}(k)$. Hence

$$\hat{Y}(k) = \bar{Y}(k) = \begin{bmatrix} y(k-l+1) \\ y(k-1+2) \\ \vdots \\ y(k) \end{bmatrix} = \begin{bmatrix} CA^{k-l+1} \\ CA^{k-l+2} \\ \vdots \\ CA^{k} \end{bmatrix} x = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{l-1} \end{bmatrix} A^{k-l+1}x,$$

for $l \leq k \leq T$. Hence, $\hat{Y}(k) = \mathcal{O}_{l-1}A^{k-l+1}x$. From (4.11), $\mathcal{K}(k) = \mathcal{O}_{l-1}$ and therefore,

$$\hat{Y}(k) = \mathcal{K}(k)A^{k-l+1}x$$

lies in the image of $\mathcal{K}(k)$ and $\psi(\hat{Y}(k)) =$ No Attack for $l \leq k \leq T$ also.
Conversely, suppose that,

$$\psi(\hat{Y}(l-1)) = \psi(\hat{Y}(l)) = ... = \psi(\hat{Y}(T)) = \text{ No Attack.}$$

We want to show that there exists $x \in \mathbb{R}^n$ with $Y(T) = \mathcal{O}_T x$, $y_\Gamma = \Gamma x$.
First note that since $\psi(\hat{Y}(l-1)) =$ No Attack, $\hat{Y}(l-1)$ lies in the image of $\mathcal{K}(l-1)$. Thus, there exists $x \in \mathbb{R}^n$ such that

$$\hat{Y}(l-1) = \mathcal{K}(l-1)x.$$

From (4.11), we can rewrite this as,

$$\hat{Y}(l-1) = \begin{bmatrix} \Gamma \\ \mathcal{O}_{l-1} \end{bmatrix} x, \tag{4.14}$$

so that $y_\Gamma = \Gamma x$, $Y(l-1) = \bar{Y}(l-1) = \mathcal{O}_{l-1}x$.
Now for $l \leq k \leq T$, $\psi(\hat{Y}(k)) = \psi(\bar{Y}(k)) =$ No Attack. This implies that $\bar{Y}(k)$ is in the image of $\mathcal{K}(k) = \mathcal{O}_{l-1}$. It follows from Lemma 4.3.3, using a very simple inductive argument, that $\bar{Y}(k) = \mathcal{O}_{l-1}A^{k-l+1}x$ for $l \leq k \leq T$. In particular, recalling the definition of $\bar{Y}(k)$, this means that $y(k) = CA^k x$, and it is easy to

see that this implies that

$$Y(T) = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(T) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^T \end{bmatrix} x = \mathcal{O}_T x.$$

This completes the proof. □

Note that the detector described in Theorem 4.3.2 satisfies the properties of consistency and soundness, as defined in Definition 4.3.1 and Definition 4.3.2, respectively. Consistency follows immediately as the theorem shows that if $Y(T) = \mathcal{O}_T x$, $y_\Gamma = \Gamma x$ for some x in $\mathbb{R}^n$, the detector will return 'No Attack'. Soundness follows from the converse direction. If our detector returns 'No Attack', the theorem implies that there exists some $x \in \mathbb{R}^n$ such that $Y(T) = \mathcal{O}_T x$ and $y_\Gamma = \Gamma x$. It now follows immediately that any consistent detector also returns 'No Attack'.

4.4 Opacity and Attack Detection

In the last chapter, we discussed opacity for linear systems. We now describe recent results that illustrate the relationship between opacity and attack detectability. As we highlighted previously in the thesis, opacity and attack detectability are important aspects for any system. These concepts allow secrets to remain private and malicious attack signals to be detected. It is reasonable to expect some sort of 'trade-off' between these aspects of a system. Informally, if a system is opaque, it is likely to be vulnerable to undetectable attacks. In this section, we present some results from [34] and [32] that make this observation more formal.

Here, we are going to use a slightly different definition of k-ISO compared to the one presented in Chapter 3. This form of k-ISO was presented in [34] and [32] and throughout this thesis, we refer to it as *uniform k-ISO*, to avoid confusion.

Definition 4.4.1. *(Uniform k-ISO [32]).*
For the system (4.3)*, given $X_s, X_{ns} \subseteq X_0$, X_s is uniform k-ISO with respect to X_{ns} if for all $k \geq 0$, for every $x_s \in X_s$ and for every sequence of inputs $U_s(k)$,*

there exists an $x_{ns} \in X_{ns}$ and a sequence of inputs $U_{ns}(k)$ such that we have,

$$Y(k, x_s, U_s(k)) = Y(k, x_{ns}, U_{ns}(k)).$$

In contrast to the earlier definition of k-ISO, uniform k-ISO requires that for every $x_s \in X_s$ and input sequence $U_s(k)$, there exist $x_{ns} \in X_{ns}$ and a single input sequence $U_{ns}(k)$ such that the corresponding output satisfies $y(j, x_s, u_s(j)) = y(j, x_{ns}, u_{ns}(j))$ for $0 \leq j \leq k$. In particular, uniform opacity and K-ISO for the set $K = \{0, \ldots, k\}$ are distinct properties. Under uniform opacity, the non-secret state $x_{ns} \in X_{ns}$ is the same for all time instants up to k. In contrast, K-ISO allows different non-secret states, x_{ns}, to be used at different time instants.
Uniform opacity can be characterised in terms of the weakly unobservable subspace $\mathcal{V}(\Sigma)$ as seen in the following lemma.

Lemma 4.4.1. *([33]) Consider the system Σ (4.3). There exist disjoint sets of secret states X_s and non-secret states X_{ns} such that X_s satisfies uniform k-ISO with respect to X_{ns} if and only if $\mathcal{V}_k(\Sigma) \neq \{0\}$.*

Proof. Suppose that for the system Σ, there exists a set of secret states X_s that satisfies uniform k-ISO with respect to the (disjoint) set of non-secret states X_{ns}. Equivalently, for every $x_s \in X_s$ and every input sequence $U_s(k)$, there exists an $x_{ns} \in X_{ns}$ and an input sequence $U_{ns}(k)$ such that

$$\mathcal{O}_k x_s + \mathcal{M}_k U_s(k) = \mathcal{O}_k x_{ns} + \mathcal{M}_k U_{ns}(k).$$

By linearity, this is equivalent to,

$$\mathcal{O}_k(x_s - x_{ns}) + \mathcal{M}_k(U_s(k) - U_{ns}(k)) = 0.$$

It follows that $\mathcal{V}_k(\Sigma) \neq 0$, as $x_s - x_{ns} \in \mathcal{V}_k(\Sigma)$. □

Lemma 4.4.1 highlights how opacity relates to the k^{th} weakly unobservable subspace $\mathcal{V}_k(\Sigma)$ for linear systems. It shows us that a non-zero $\mathcal{V}_k(\Sigma)$ is essential for a system Σ to satisfy uniform k-ISO with respect to the disjoint secret and non-secret sets. Note that if $k \geq n-1$, then there exist disjoint sets X_s, X_{ns} such that X_s satisfies k-ISO with respect to X_{ns} if and only if the WUS $\mathcal{V}(\Sigma) \neq \{0\}$. From now on, we assume $k \geq n-1$. The following result from [32] illustrates the relationship between opacity and undetectable attacks. To establish this

relationship, the authors in [34] proceed in the following manner. First, they show that if Σ given by (4.3) is uniformly k-ISO with respect to some sets X_s, X_{ns}, then this implies the existence of an undetectable attack $E(k) \neq 0$ for $\bar{\Sigma} = \Sigma$. Then, they show that every system $\bar{\Sigma}$ satisfying the condition,

$$\text{Im} \begin{bmatrix} \bar{B} \\ \bar{D} \end{bmatrix} \supseteq \text{Im} \begin{bmatrix} B \\ D \end{bmatrix}$$

admits an undetectable attack. Our approach to proving the second fact differs from theirs and makes use of the recurrence in (4.7), whereas they use properties of the Kronecker product to establish this.

Theorem 4.4.1. *([34]) Consider the LTI system Σ (4.3), and let X_s, X_{ns} denote disjoint sets of secret and non-secret states, and suppose $k \geq n-1$. Suppose that X_s satisfies uniform k-ISO with respect to X_{ns} for Σ. Then, there exists an attacked system $\bar{\Sigma}$ (4.2) such that $\bar{\Sigma}$ admits an undetectable attack $E(k)$. Moreover, any system $\bar{\Sigma}$ with*

$$Im \begin{bmatrix} \bar{B} \\ \bar{D} \end{bmatrix} \supseteq Im \begin{bmatrix} B \\ D \end{bmatrix}.$$

admits an undetectable attack.

Proof. As X_s is uniformly k-ISO with respect to X_{ns} for Σ and $k \geq n-1$, it follows from Lemma 4.4.1 that the weakly unobservable subspace $\mathcal{V}(\Sigma) \neq 0$. If we set $\bar{\Sigma} = \Sigma$, then clearly $\mathcal{V}(\bar{\Sigma}) \neq \{0\}$.
Choose a non-zero vector $\bar{x}_0 \in \mathcal{V}(\bar{\Sigma})$. Corollary 4.3.1 implies that there exists an attack $E(k)$ satisfying

$$\mathcal{M}_k E(k) = -\mathcal{O}_k \bar{x}_0. \tag{4.15}$$

Since Σ is observable, so is $\bar{\Sigma}$. As $k \geq n-1$, it follows that $\mathcal{O}_k$ has rank n. Thus, $\bar{x}_0 \neq 0$ implies that

$$\mathcal{O}_k \bar{x}_0 \neq 0.$$

Hence, using this fact and (4.15), we can conclude that $E(k) \neq 0$.
Next, suppose that X_s is k-ISO with respect to X_{ns} for Σ, and that the system $\bar{\Sigma}$ satisfies,

$$\text{Im} \begin{bmatrix} \bar{B} \\ \bar{D} \end{bmatrix} \supseteq \text{Im} \begin{bmatrix} B \\ D \end{bmatrix}. \tag{4.16}$$

As X_s is k-ISO with respect to X_{ns} for Σ, we have that $\mathcal{V}(\Sigma) \neq \{0\}$. Recall the recurrence for the weakly unobservable subspaces given in (4.7)

$$\mathcal{V}_{j+1} = \begin{bmatrix} A \\ C \end{bmatrix}^{-1} \left(\mathcal{V}_j \times \{0\} + \operatorname{Im} \begin{bmatrix} B \\ D \end{bmatrix} \right).$$

It is not hard to see that $\mathcal{V}_0(\Sigma) \subseteq \mathcal{V}_0(\bar{\Sigma})$ follows from the assumption (4.16). Further, it follows from (4.16) and the above recurrence that $\mathcal{V}_1(\Sigma) \subseteq \mathcal{V}_1(\bar{\Sigma})$, and a simple inductive argument shows that $\mathcal{V}_k(\Sigma) \subseteq \mathcal{V}_k(\bar{\Sigma})$. It now follows immediately that $\mathcal{V}_k(\bar{\Sigma}) \neq \{0\}$. As Σ, $\bar{\Sigma}$ share the same A, C matrices, the k-step observability matrix of the $\bar{\Sigma}$ is the same as for Σ, namely $\mathcal{O}_k$. As Σ is observable and $k \geq n-1$, $\mathcal{O}_k$ has rank n.
Similarly to the argument above, choose a non-zero $\bar{x}_0$ in $\mathcal{V}_k(\bar{\Sigma})$. We know that there exists $E(k)$ with $\bar{\mathcal{M}}_k E(k) = -\mathcal{O}_k \bar{x}_0$ As above, we can conclude that $-\mathcal{O}_k \bar{x}_0 \neq 0$ and hence that $E(k) \neq 0$ and is an undetectable attack for the system Σ. This completes the proof. □

Corollary 4.4.1. *([33]) Consider the LTI-system Σ, attack system $\bar{\Sigma}$ and let $[B^\top \quad D^\top]^\top$ have full column rank. Then, there exist sets X_s, X_{ns} such that X_s is uniformly k-ISO with respect to X_{ns} for Σ if and only if there exists an undetectable attack $E(T) \neq 0$.*

When the pair of sets X_s, X_{ns} exists, the set X_s is informally referred to as an *opaque set*. Theorem 4.4.1 shows us that the existence of opaque sets always implies the existence of an attack system $\bar{\Sigma}$ with undetectable attack inputs. Further, Corollary 4.4.1 shows that if the matrix $[B \;\; D]^\top$ has full column rank, then the existence of undetectable attacks implies the existence of opaque sets. These two results highlight that we can't have opacity without making our system vulnerable to undetectable attacks. In the following example, we illustrate these ideas.

Example 4.4.1. *Consider the following LTI system,*

$$\begin{aligned} \Sigma: \quad x(t+1) &= \begin{bmatrix} 4 & 1 \\ -2 & -1 \end{bmatrix} x(t) + \begin{bmatrix} -1 \\ 2 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t). \end{aligned}$$

where, $A = \begin{bmatrix} 4 & 1 \\ -2 & -1 \end{bmatrix}, B = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix}$ *and* $D = 0$.

Choose an initial state $x_0 = [0 \quad x_2]^\top$*, where* $x_2 \in \mathbb{R}$ *is non-zero, and the input sequence* $U(2) = [x_2 \; x_2 \; x_2]^\top$*. Then clearly* $y(0) = Cx_0 = 0$ *and*

$$\begin{aligned} x(1, x_0, u(0)) &= \begin{bmatrix} 4 & 1 \\ -2 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ x_2 \end{bmatrix} + \begin{bmatrix} -1 \\ 2 \end{bmatrix} x_2 \\ &= \begin{bmatrix} x_2 \\ -x_2 \end{bmatrix} + \begin{bmatrix} -x_2 \\ 2x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ x_2 \end{bmatrix}, \end{aligned}$$

so $y(1) = 0$ *also. Moreover, it is easy to see that* $x(2, x_0, U(1)) = x_0$ *and hence* $y(2) = 0$ *as well. This shows that the subspace* $\mathcal{V}_2(\Sigma) \neq \{0\}$*. Lemma 4.4.1 implies that there exist sets* X_s*,* X_{ns} *such that* X_s *is uniformly* 2*-ISO for* Σ *and Theorem 4.4.1 then implies that an undetectable attack can be made on the system.*

4.5 Concluding Remarks.

In this chapter, we discussed various security and attack detection concepts for LTI systems. We dealt with the notion of undetectable attacks and examined these for the cases with and without side initial state information. We also looked in some detail at a recently proposed design for a detector, and showed that this detector was both consistent and sound. Lastly, we looked at the fundamental trade-off between undetectable attacks and opacity. We presented key results from [34], giving a different proof for Theorem 4.4.1, which showed that we cannot have an opaque system without making our system vulnerable to undetectable attacks. We used the idea of weakly unobservable subspaces to show this.

Chapter 5

Max-plus Algebra

5.1 Introduction

In this chapter, we discuss some key concepts of max-plus algebra that will provide the foundation for discussing possible extensions of the notion of opacity to max-plus linear systems in the next chapter. Max-plus algebra is an algebraic structure with two operations. The conventional operations of addition and multiplication are replaced by maximization and addition, respectively. This algebraic structure forms a semiring [22] with elements that are typically given by the real numbers together with the element $-\infty$, which acts as the neutral element for maximization.

From a historical perspective, max-plus algebra and its applications are relatively recent developments. Early results can be traced to publications such as [12], [14]. One of the first major milestones in the field came with the publication of the book [13] in 1979. Many studies, such as [29] and [23], have explored max-plus algebra due to its effectiveness in modelling and analysing systems where synchronization and timing constraints are critical. In particular, max-plus algebra has a close relation to discrete event systems. Although discrete event systems often lead to a non-linear description in conventional algebra, there exists a subclass of DES for which the model becomes 'linear' when it is formulated in max-plus algebra. Some examples of these systems include manufacturing systems, communication networks, and railway systems [38]. In particular, The Dutch railway has been effectively studied and modelled using a max-plus approach [47].

This chapter begins with some fundamental definitions and concepts of max-plus algebra, followed by examining its connection to graph theory. Since this algebraic framework forms a semiring without additive inverses, we also discuss methods for solving affine equations over max-plus algebra. Before introducing formal definitions, we will begin with an illustrative example of a railway network, inspired by the works of [28].

5.1.1 Motivational Example

Consider the simple railway network between two stations illustrated in Figure 5.1. The stations are labeled $S1$ and $S2$. As shown in the figure, one track runs from $S1$ to $S2$ with a travel time of 5 time units. A second track connects $S2$ to $S1$ with a travel time of 8 time units. These two tracks form a circuit between the two stations: trains departing from $S1$ travel to $S2$, then return to $S1$ along the other track. Similarly, trains that start at $S2$ will, after visiting $S1$, return to $S2$. We also have a track that loops at $S2$. This track may connect the station to suburban areas of a given city. A trip around this track lasts 3 units of time.
Suppose we wish to design a timetable for $S1$ and $S2$ that satisfies the following criteria:

1. The frequency of the trains (i.e., the number of departures per unit of time) must be as high as possible.

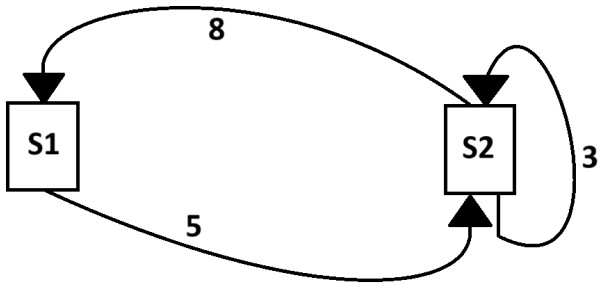
2. The frequency must be identical across all three tracks.

3. For $S2$, trains arriving at this station must wait for each other to allow a transfer of passengers.

4. Trains should depart as soon as possible.

To keep the model relatively simple, we consider a setup that works with three trains, with one train assigned to each track. Moreover, we also consider that the changeover of passengers at $S2$ is instantaneous.
Let x_1 denote the departure times of $S1$ and similarly, x_2 denote the departure times of the two trains in $S2$. Together, the departure times can be written as a vector $x \in \mathbb{R}^2$. The first departure time vector will be given by $x(0)$ and then the trains leave at the time instants given by the two elements of the vector $x(1)$ and so on. The k^{th} departure times are denoted by $x(k-1)$. Based on the

Figure 5.1: Graph of Railway network.



journey times, we can express x_1 and x_2 as,

$$\begin{aligned} x_1(k+1) &= \max(x_1(k), x_2(k)+8), \\ x_2(k+1) &= \max(x_1(k)+5, x_2(k)+3). \end{aligned} \tag{5.1}$$

Given initial departure times $x(0)$, all future departures are uniquely determined by (5.1). For example, if we set $x_1(0) = x_2(0) = 0$, then the sequence $x(k)$, for $k = 0, 1, 2, \ldots$, is given by

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 8 \\ 5 \end{pmatrix}, \begin{pmatrix} 13 \\ 13 \end{pmatrix}, \begin{pmatrix} 21 \\ 18 \end{pmatrix}, \begin{pmatrix} 26 \\ 26 \end{pmatrix}, \begin{pmatrix} 34 \\ 31 \end{pmatrix}, \cdots . \tag{5.2}$$

We could also change our initial state $x(0)$ to make the trains at each station leave at different times such that $x_1(0) = 2$ and $x_2(0) = 0$ (i.e. The first train at $S1$ leaves at time 2 and the first trains at $S2$ leave at time 0). Then we obtain the sequence,

$$\begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 8 \\ 7 \end{pmatrix}, \begin{pmatrix} 15 \\ 13 \end{pmatrix}, \begin{pmatrix} 21 \\ 20 \end{pmatrix}, \begin{pmatrix} 28 \\ 26 \end{pmatrix}, \begin{pmatrix} 34 \\ 33 \end{pmatrix}, \cdots . \tag{5.3}$$

Comparing the two sequences (5.2) and (5.3), and defining the *interdeparture* time as the time duration between two subsequent departures along the same track, we observe that both sequences yield the same average interdeparture time of 6.5, despite x_1 starting at time 2 in (5.3) instead of at time 0 as in (5.2). [28] asks whether we can select appropriate initial departure times to achieve a smaller average interdeparture time and thereby construct a 'faster' timetable.

The answer is no, as the time duration for a train to go around the inner circuit is equal to 13 and there are two trains on this circuit. Therefore, the average interdeparture time can never be smaller than $13/2 = 6.5$.
This example shows that we can model such systems using max-plus operations, where taking the maximum represents the synchronization of events, and addition accounts for fixed delays. This algebraic framework captures the timing and dependencies in systems such as this railway network, where the next event can only occur after certain events have been completed. Note that, as previously mentioned, this type of modelling can also be applied in manufacturing and communication networks, with examples found in [19] and [20], respectively.

5.2 Fundamentals of max-plus algebra

In max-plus algebra, the ground set is denoted by $\mathbb{R}_{max} = \mathbb{R} \cup \{-\infty\}$. For elements $a, b \in \mathbb{R}_{max}$, we define two binary operations $\oplus$ and $\otimes$, given by,

$$a \oplus b = \max\{a, b\} \qquad \text{and} \qquad a \otimes b = a + b, \tag{5.4}$$

Similar to conventional algebra, the operation $\otimes$ has priority over the operation $\oplus$. For example, consider the expression,

$$-7 \otimes 5 \oplus 12 \otimes -1.$$

Applying the precedence of operations, this is interpreted as,

$$\begin{aligned} (-7 \otimes 5) &\oplus (12 \otimes -1) \\ (-2) &\oplus (11) \\ = 11. & \end{aligned}$$

$\otimes$ also distributes over $\oplus$. For instance, let $a, b, c \in \mathbb{R}_{max}$. Then it holds that,

$$\begin{aligned} a \otimes (b \oplus c) &= a + (\max\{b, c\}) \\ &= \max\{a + b, a + c\} \\ &= (a \otimes b) \oplus (a \otimes c). \end{aligned}$$

Powers are defined in the obvious way for max-plus algebra. For $a \in \mathbb{R}_{max}$ and $n \in \mathbb{N}$,

$$a^{\otimes n} = \underbrace{a + a + a \cdots + a}_{\text{n times}} = n \times a.$$

for example,

$$11^{\otimes 3} = 3 \times 11 = 33.$$

This also holds for negative powers as,

$$8^{\otimes -3} = -3 \times 8 = -24.$$

In max-plus algebra, the additive and multiplicative identities are given by $-\infty$ and 0, respectively. Specifically, for any $a \in \mathbb{R}_{max}$ we have,

$$a \oplus -\infty = -\infty \oplus a = a \tag{5.5}$$

and,

$$a \otimes 0 = 0 \otimes a = a. \tag{5.6}$$

$-\infty$ is also absorbing for $\otimes$ meaning,

$$a \otimes -\infty = -\infty \otimes a = -\infty. \tag{5.7}$$

The algebraic structure

$$\mathcal{R}_{max} = (\mathbb{R}_{max}, \oplus, \otimes, -\infty, 0)$$

is referred to as the max-plus algebra. Note that both operations $\oplus$ and $\otimes$ are associative and commutative, and $\oplus$ is idempotent such that for all $a \in \mathbb{R}_{max}$, $a \oplus a = a$. $\mathcal{R}_{max}$ forms a commutative idempotent semiring [28], [22]. A key observation is that $\oplus$ does not admit an additive inverse. This makes solving simple affine equations significantly more complex over the max-plus algebra.

5.3 Matrices and Max-Plus Algebra

The set of $n \times m$ matrices in max-plus algebra is denoted by $\mathbb{R}_{max}^{n \times m}$, where each entry belongs to $\mathbb{R}_{max}$. Note that max-plus matrix operations are analogous to conventional matrix arithmetic. For matrices $A, B \in \mathbb{R}_{max}^{n \times m}$, the max-plus sum is defined entrywise as,

$$[A \oplus B]_{ij} = a_{ij} \oplus b_{ij} = \max\{a_{ij}, b_{ij}\}$$

for all i, j. That is, each entry of the resulting matrix is the maximum of the corresponding entries of A and B.
Given $A \in \mathbb{R}_{max}^{n\times m}$ and $B \in \mathbb{R}_{max}^{m\times p}$, the max-plus product is defined as,

$$[A \otimes B]_{ik} = \bigoplus_{j=1}^{m} a_{ij} \otimes b_{jk} = \max_{j}\{a_{ij} + b_{jk}\}. \tag{5.8}$$

for all i, k. To illustrate these matrix operations, consider the following example.

Example 5.3.1. *Consider the matrices,*

$$A = \begin{bmatrix} -\infty & 3 \\ 0 & 4 \end{bmatrix} \quad \textit{and} \quad B = \begin{bmatrix} 6 & 5 \\ 1 & -\infty \end{bmatrix}$$

where $A, B \in \mathbb{R}_{max}^{2\times 2}$. $A \oplus B$ is given by,

$$\begin{aligned} A \oplus B &= \begin{bmatrix} -\infty & 3 \\ 0 & 4 \end{bmatrix} \oplus \begin{bmatrix} 6 & 5 \\ 1 & -\infty \end{bmatrix} \\ &= \begin{bmatrix} -\infty \oplus 6 & 3 \oplus 5 \\ 0 \oplus 1 & 4 \oplus -\infty \end{bmatrix} \\ &= \begin{bmatrix} 6 & 5 \\ 1 & 4 \end{bmatrix}. \end{aligned}$$

Max-plus matrix multiplication of A and B is given by,

$$\begin{aligned} A \otimes B &= \begin{bmatrix} -\infty & 3 \\ 0 & 4 \end{bmatrix} \otimes \begin{bmatrix} 6 & 5 \\ 1 & -\infty \end{bmatrix} \\ &= \begin{bmatrix} (-\infty \otimes 6) \oplus (3 \otimes 1) & (-\infty \otimes 5) \oplus (3 \otimes -\infty) \\ (0 \otimes 6) \oplus (4 \otimes 1) & (0 \otimes 5) \oplus (4 \otimes -\infty) \end{bmatrix} \\ &= \begin{bmatrix} (-\infty) \oplus (4) & (-\infty) \oplus (-\infty) \\ (6) \oplus (5) & (5) \oplus (-\infty) \end{bmatrix} \\ &= \begin{bmatrix} 4 & -\infty \\ 6 & 5 \end{bmatrix}. \end{aligned}$$

The zero and identity matrices can also be defined in the max-plus setting. Let $\mathcal{E}(n, m)$ denote the $n \times m$ max-plus zero matrix, whose entries are all equal to

$-\infty$. E_n will denote the $n \times n$ identity matrix, which is defined as,

$$[E_n]_{ij} = \begin{cases} 0 & \text{for } i = j \\ -\infty & \text{otherwise} \end{cases}$$

That is, E_n contains zero in each entry along its diagonal and $-\infty$ everywhere else. The transpose of an element $A \in \mathbb{R}_{max}^{n\times m}$, denoted by A^T, is defined in the usual way by $[A^T]_{ij} = a_{ji}$ for all i, j. In matrix addition and multiplication, $\otimes$ also has priority over $\oplus$.
For $A \in \mathbb{R}_{max}^{n\times n}$, denote the k^{th} power of A by $A^{\otimes k}$:

$$A^{\otimes k} = \underbrace{A \otimes A \otimes \cdots \otimes A}_{k \text{ times}},$$

for $k \in \mathbb{N}$. By convention, $A^{\otimes 0} = E_n$.

5.4 Max-plus Algebra and its Connection to Graph Theory

As shown in [6] and [55], matrices can be used to represent graph-related information in conventional algebra, and the same holds in the max-plus setting. In particular, they can represent the structure of a *communication graph*, defined as follows.

Definition 5.4.1. *(Communication graph [16]) Consider the matrix $A \in \mathbb{R}_{max}^{n\times n}$. The communication graph of A, denoted by $G(A)$, is a weighted directed graph with vertices $1, 2, 3, \ldots, n$ and an arc (j, i) with weight a_{ij} for each $a_{ij} \neq -\infty$.*

As stated in Definition 5.4.1, the matrix entry a_{ij} represents the weight of the directed arc from node j to node i in the communication graph. The entries equal to $-\infty$ indicate the absence of an arc between the corresponding nodes.
In the max-plus setting, there exists a close relation between the communication graph $G(A)$ and the powers of A. The weight of a path in $G(A)$ is defined as the sum of all weights of the arcs in the path. Let $A \in \mathbb{R}_{max}^{n\times n}$ and $k \in \mathbb{N}\backslash\{0\}$, then for all $i, j \in \{1, 2, \ldots, n\}$, we have,

$$(A^{\otimes k})_{ij} = \max_{i_1, i_2, \cdots, i_{k-1} \in \{1,2,\cdots,n\}} (a_{ii_1} + a_{i_1 i_2} + \cdots + a_{i_{k-1} j}).$$

That is, $(A^{\otimes k})_{ij}$ gives the maximal weight of all paths of length k that start at node j and end at node i. If no such path exists, $(A^{\otimes k})_{ij} = -\infty$ [16], [28].
A directed graph $G(A)$ is said to be strongly connected if for any two different nodes i, j of the graph, there exists a directed path from i to j. We now recall the following definition of irreducible matrices in the context of graph theory.

Definition 5.4.2. *(Irreducible matrix [43])*
A matrix $A \in \mathbb{R}_{max}^{n\times n}$ is called irreducible if its associated communication graph $G(A)$ is strongly connected.

In [17], Definition 5.4.2 was reformulated in terms of max-plus matrix powers. $A \in \mathbb{R}_{max}^{n\times n}$ is irreducible if,

$$(A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus \cdots \oplus A^{\otimes n-1})_{ij} \neq -\infty \quad \text{for all } i, j \text{ with } i \neq j. \tag{5.9}$$

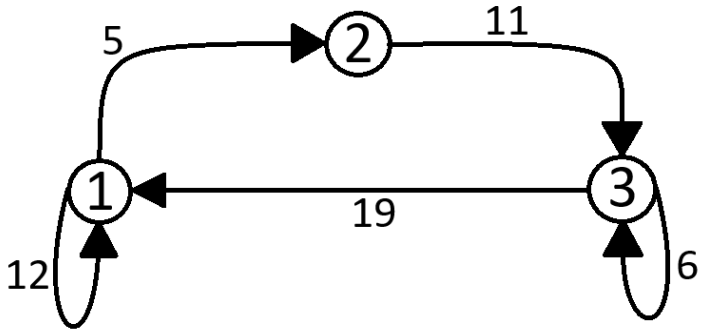
This condition means that for two arbitrary nodes i and j of $G(A)$ with $i \neq j$, there exists at least one path of length less than or equal to $n-1$ from j to i. In the following example we demonstrate these ideas.

Example 5.4.1. *Consider the matrix A defined as,*

$$A = \begin{bmatrix} 12 & -\infty & 19 \\ 5 & -\infty & -\infty \\ -\infty & 11 & 6 \end{bmatrix}.$$

The corresponding communication graph $G(A)$ is shown in Figure 5.2 below. Then we can compute,

Figure 5.2: Communication Graph $G(A)$ for Example 5.4.1



$$A^{\otimes 2} = A \otimes A = \begin{bmatrix} 24 & 30 & 31 \\ 17 & -\infty & 24 \\ 16 & 17 & 12 \end{bmatrix}.$$

Each entry of $A^{\otimes 2}$ *represents the maximum weight of all paths of length* 2 *between the corresponding nodes in* $G(A)$*. For example, the entry* $(A^{\otimes 2})_{13} = 31$ *corresponds to the maximum weight of a path of length* 2 *from node* 3 *to node* 1*, as can be verified using Figure 5.2.*
We can similarly compute the matrix for $k = 3$ *:*

$$A^{\otimes 3} = \begin{bmatrix} 36 & 42 & 43 \\ 29 & 35 & 36 \\ 28 & 23 & 35 \end{bmatrix}.$$

Note that $(A^{\otimes 3})_{11} = 36$*, which is the maximum weight of all paths of length* 3 *from node* 1 *back to itself. There are two such paths:*

- $1 \to 1 \to 1 \to 1$*, which loops at node* 1 *and has a total weight* $12+12+12 = 36$*.*
- $1 \to 2 \to 3 \to 1$ *with a total weight of* $19 + 11 + 19 = 35$*.*

Since 36 *is the larger of the two,* $(A^{\otimes 3})_{11} = 36$ *as expected.*
As shown in the communication graph $G(A)$ *in Figure 5.2, the graph is strongly connected, meaning that there exists a path between every pair of nodes. Therefore, by Definition 5.4.2, the matrix* A *is irreducible. We could also used the max-plus formulation in* (5.9) *such that,*

$$A \oplus A^{\otimes 2} = \begin{bmatrix} 24 & 30 & 31 \\ 17 & -\infty & 24 \\ 16 & 17 & 12 \end{bmatrix}.$$

Hence as,

$$(A \oplus A^{\otimes 2})_{ij} \neq -\infty \quad \text{for all } i \neq j,$$

the matrix A *is irreducible by* (5.9)*.*

5.5 Solving affine equations over max-plus algebra

Let $A \in \mathbb{R}_{max}^{n\times n}$ and $b \in \mathbb{R}_{max}^{n}$. In max-plus algebra, a system of max-plus linear equations of the form $A \otimes x = b$ may not always yield a solution, even if the system overdetermined (i.e., has more columns than rows). To address this issue, the concept of a subsolution was introduced in [13] and [3], and is defined as follows.

Definition 5.5.1. *(Subsolution [16]) Let $A \in \mathbb{R}_{max}^{n\times n}$ and $b \in \mathbb{R}_{max}^{n}$. $\bar{x} \in \mathbb{R}_{max}^{n}$ is a subsolution of the system of max-plus linear equations $A \otimes x = b$ if $A \otimes \bar{x} \leq b$, that is, $[A \otimes \bar{x}]_i \leq b_i$ for all $i \in \{1, \ldots, n\}$.*

When no solution to $A \otimes x = b$ exists, it is of interest to consider subsolutions. Among these, the largest subsolution is often of particular interest. A vector $\bar{x}$ is called the largest solution if it satisfies $A \otimes \bar{x} \leq b$ and for any other subsolution y, it holds that $y \leq \bar{x}$. The largest subsolution $\bar{x}$ of $A \otimes x = b$ is given by,

$$\bar{x}_j = \min_i (b_i - a_{ij}) \qquad \text{for } j = 1, 2, \ldots, n \tag{5.10}$$

or equivalently,

$$\bar{x} = -(A^T \otimes (-b)). \tag{5.11}$$

For (5.10), we assume that all components of b are finite to avoid undefined cases such as $\min(-\infty - (-\infty))$, since $\infty \notin \mathbb{R}_{max}$. The following example shows how to find a subsolution when no exact solution exists.

Example 5.5.1. *Consider the matrices,*

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 0 \\ 3 \end{bmatrix}.$$

Let $x \in \mathbb{R}_{max}^{2}$. In this example, there is no solution that satisfies $A \otimes x = b$. This is because for the equation,

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix},$$

we need to satisfy,

$$\begin{aligned} \max(1+x_1, 2+x_2) &= 0, && (5.12)\\ \max(3+x_1, 4+x_2) &= 3. && (5.13) \end{aligned}$$

For (5.12)*, we must have* $1+x_1 \leq 0$ *and* $2+x_2 \leq 0$ *with at least one equality. Hence,*

$$x_1 \leq -1, \;\; x_2 \leq -2, \quad \textit{with} \;\; x_1 = -1 \; \textit{or} \; x_2 = -2.$$

For (5.13)*, we must have* $3+x_1 \leq 3$ *and* $4+x_2 \leq 3$ *with at least one equality. Hence,*

$$x_1 \leq 0, \;\; x_2 \leq -1, \quad \textit{with} \;\; x_1 = 0 \; \textit{or} \; x_2 = -1.$$

Combining the inequalities gives $x_1 \leq -1$ *and* $x_2 \leq -2$*. Substituting these bounds into the left-hand side of* (5.13)*,*

$$\max(3+x_1, 4+x_2) \leq \max(3-1, 4-2) = \max(2,2) = 2 < 3,$$

which contradicts (5.13)*. Therefore, there is no* $x \in \mathbb{R}^2_{max}$ *that satisfies* $A \oplus x = b$*. However, the largest subsolution* $\bar{x}$ *of* $A \otimes x = b$ *is given by,*

$$\begin{aligned} \bar{x} &= -(A^T \otimes (-b)) \\ &= -\left(\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ -3 \end{bmatrix} \right) \\ &= \begin{bmatrix} -1 \\ -2 \end{bmatrix}. \end{aligned}$$

Hence we have, $A \otimes \bar{x} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \leq b$*, the greatest subsolution to* $A \otimes x = b$*. To see why* $\bar{x}$ *is the greatest subsolution, suppose there exists another subsolution* $y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ *such that,* $A \otimes y \leq b$*. Then from the inequalities above,*

$$y_1 \leq -1, \quad y_2 \leq -2.$$

But $\bar{x} = [-1 \;\; -2]^\top$ *is componentwise the largest vector satisfying these inequalities. Hence, any other subsolution* y *must satisfy* $y \leq \bar{x}$*. Thus* $\bar{x}$ *is the greatest subsolution.*

Since the operation $\oplus$ has no inverse, equations of the form $x = A \otimes x \oplus b$ can not, in general, be rewritten in the form $\bar{A} \otimes x = b$ for some $\bar{A}$. This limitation leads to the use of the Kleene star of a matrix A, defined by

$$A^* = \bigoplus_{k=0}^{\infty} A^{\otimes k} = E_n \oplus A \oplus A^{\otimes 2} \oplus \ldots, \tag{5.14}$$

Recall that a circuit is a closed path that starts and ends at the same node without repeating any arcs. If the communication graph associated with A contains no positive-weight circuits, then the series in (5.14) converges after at most $n-1$ terms. In that case,

$$A^* = E_n \oplus A \oplus A^{\otimes 2} \oplus \ldots, \oplus A^{\otimes n-1}. \tag{5.15}$$

Let $\mathcal{C}(A)$ denote the set of all circuits in $G(A)$. The length and weight of a path are defined as the number of edges it contains and the sum of the weights of all its edges, respectively. For a path p, we denote the length by $|p|_l$ and its weight by $|p|_w$. The *maximal average circuit weight* of a communication graph $G(A)$ is defined as,

$$\sigma_{max} = \max_{p \in \mathcal{C}(A)} \frac{|p|_w}{|p|_l}. \tag{5.16}$$

We note here that the maximal average circuit weight plays a key role in spectral theory over max-plus algebra. In fact, it is the dominant eigenvalue of the matrix A. While not directly relevant to our focus here, this aspect of max-plus theory has been extended to sets of matrices, and versions of the generalized and joint spectral radius have been studied in this context [46]. With (5.15) and (5.16) in mind, these observations lead to the following theorem.

Theorem 5.5.1. *([28]) Let $A \in \mathbb{R}_{max}^{n \times n}$ and $b \in \mathbb{R}_{max}^{n}$. If the communication $G(A)$ has a maximal average circuit weight σ_{max} less than or equal to 0, then the vector $x = A^* \otimes b$ solves the equation $x = A \otimes x \oplus b$. Furthermore, if the circuit weights in $G(A)$ are negative, then the solution to $x = A \otimes x \oplus b$ is unique.*

Theorem 5.5.1 shows us that if $\sigma_{max} \leq 0$ then $x = A \otimes x \oplus b$ admits a solution given by $x = A^* \otimes b$. Moreover, if $\sigma_{max} < 0$ then this solution is also unique. The intuition behind the uniqueness is related to the negative weights in the circuits of A. Specifically, for negative circuits, the powers of A tend to $-\infty$ such that,

$$\lim_{k \to \infty} A^{\otimes k} \otimes x = -\infty.$$

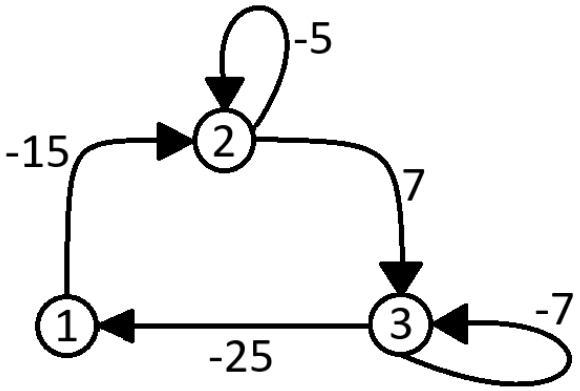
Consequently, the series A^* converges. Any additional 'looping' through A would decrease the value of x (since negative circuit reduce the maximum), which prevents any other solution from satisfying $x = A \otimes x \oplus b$. Therefore, the solution necessarily converges to $x = A \otimes x \oplus b$, ensuring uniqueness [28]. We present the following example of how to compute a solution to $x = A \otimes x \oplus b$.

Example 5.5.2. *Consider the matrices,*

$$A = \begin{bmatrix} -\infty & -\infty & -25 \\ -15 & -5 & -\infty \\ -\infty & 7 & -7 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix}.$$

We aim to find a vector $x \in \mathbb{R}^3_{max}$ that solves the equation $x = A \otimes x \oplus b$. The corresponding communication graph $G(A)$ is depicted in Figure 5.3 below.

Figure 5.3: Communication graph $G(A)$ of Matrix A in Example 5.5.2.



We need to check that the maximal average circuit weight σ_{max} is less than or equal to zero. In $G(A)$ we have three circuits,

- $1 \to 2 \to 3 \to 1$.
- $2 \to 2$.
- $3 \to 3$.

Using (5.16)*, the maximal average circuit weight is given by,*

$$\begin{aligned}\sigma_{max} &= \max(\frac{-15+7-25}{3}, \frac{-7}{1}, \frac{-5}{1}) \\ &= \max(-11, -7, -5) \\ &= -5.\end{aligned}$$

As $\sigma_{max} = -5$*, by Theorem 5.5.1,* $x = A^* \otimes b$ *solves the equation* $x = A \otimes x \oplus b$ *and our solution is unique as* σ_{max} *is negative. The next step is to calculate* $A^* \otimes b$*. Note that,*

$$A^{\otimes 2} = A \otimes A = \begin{bmatrix} -\infty & -18 & -32 \\ -20 & -10 & -40 \\ -8 & 0 & -14 \end{bmatrix}.$$

As $G(A)$ *contains no positive circuits,* A^* *converges after at most* 2 *steps such that,*

$$\begin{aligned}A^* &= E(3) \oplus A \oplus A^2 \\ &= \begin{bmatrix} 0 & -\infty & -\infty \\ -\infty & 0 & -\infty \\ -\infty & -\infty & 0 \end{bmatrix} \oplus \begin{bmatrix} -\infty & -\infty & -25 \\ -15 & -5 & -\infty \\ -\infty & 7 & -7 \end{bmatrix} \oplus \begin{bmatrix} -\infty & -18 & -32 \\ -20 & -10 & -40 \\ -8 & 0 & -14 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -18 & -25 \\ -15 & 0 & -40 \\ -8 & 7 & 0 \end{bmatrix}.\end{aligned}$$

Hence,

$$x = A^* \otimes b = \begin{bmatrix} 0 & -18 & -25 \\ -15 & 0 & -40 \\ -8 & 7 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix}.$$

We can verify that x does in fact satisfy $x = A \otimes x \oplus b$,

$$\begin{aligned}
\begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix} &= \begin{bmatrix} -\infty & -\infty & -25 \\ -15 & -5 & -\infty \\ -\infty & 7 & -7 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix} \\
&= \begin{bmatrix} -20 \\ -7 \\ 5 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix} \\
&= \begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix}.
\end{aligned}$$

Therefore, by Theorem 5.5.1, $x = \begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix}$ is the unique solution to the equation $x = A \otimes x \oplus b$.

5.6 Concluding Remarks

In this chapter, we provided a very brief introduction to some fundamental aspects of max-plus algebra. We first discussed its effectiveness in modeling and analyzing systems where synchronization and timing are critical, using a simple railway system inspired by [28]. We then formally introduced max-plus algebra in a general setting and extended its definition to matrices. We highlight the connection between matrices and graphs, showing that the powers of a general matrix A can provide information about the maximal weighted path between two nodes. Lastly, we examined solving equations in the max-plus semiring. Equations of the form $A \otimes x = b$ may not always have a solution, which leads to the notion of subsolutions such that $A \otimes \bar{x} \leq b$. We also studied equations of the form $x = A \otimes x \oplus b$, where a solution can be obtained using the Kleene star. This chapter serves as a foundation for discussing max-plus linear systems and opacity concepts in the next chapter.

Chapter 6

Opacity Results for Max-plus Systems

6.1 Introduction

In this chapter, we consider linear systems within the max-plus algebra framework. Certain types of discrete event systems (DESs) that are usually described with nonlinear models can be formulated as linear systems over the max-plus algebra. In particular, this is possible for DESs with synchronisation but no concurrency [17].

The literature on max-plus linear systems spans both control-oriented and algebraic approaches. For instance, the authors of [26] explore geometric control for max-plus linear systems. On the other hand, works such as [37] investigate algebraic aspects of max-plus structures, including interval tensors, which provide a means to extend interval linear systems to interval multi-linear systems in the max-plus setting.

System properties such as reachability, controllability, and observability have also been studied in the max-plus setting [21], [61], and [25]. When compared to the formulation for conventional linear systems in Chapter 3, the max-plus versions are often more restrictive. This is due to key properties of max-plus algebra; in particular, the absence of additive inverses and the idempotency of addition complicate the analysis of these properties [21].

In this chapter, we provide a brief overview of some control-theoretic properties within the max-plus algebra framework and offer some initial thoughts on

formulating opacity in the context of max-plus linear systems, with particular emphasis on its connection to reachability.

6.2 Fundamental Properties of Max-plus Linear Systems

Consider the following linear time-invariant system expressed in max-plus algebra,

$$\begin{aligned} x(t+1) &= A \otimes x(t) \oplus B \otimes u(t) \\ y(t) &= C \otimes x(t). \end{aligned} \tag{6.1}$$

Here, $x \in \mathbb{R}_{max}^n$, $u \in \mathbb{R}_{max}^m, y \in \mathbb{R}_{max}^p$ denote the state, input and output, respectively. The matrices A, B, C, have entries in $\mathbb{R}_{max}$ and are of dimensions, $A \in \mathbb{R}_{max}^{n\times n}, B \in \mathbb{R}_{max}^{n\times m}$, and $C \in \mathbb{R}_{max}^{p\times n}$. The system (6.1) has the same form as the standard LTI system described in Chapter 3 (seen in (3.1)). The only distinction here is that we are using max-plus operations. As before, $x(0) = x_0 \in X_0$, where $X_0 \subseteq \mathbb{R}_{max}^n$ denotes the set of possible initial states for the system. The time t takes discrete values, where $t \in \mathbb{Z}_+$, the set of non-negative integers. As before, we use the notation $U(T) = [u(0)^\top \;\; u(1)^\top \;\; ... \;\; u(T)^\top]^\top$ for the vector of inputs up to and including time T.
The state equation solutions for max-plus linear systems take an analogous form to the classical case. Given an initial state x_0 and a sequence of inputs $U(T-1)$, the states in the system (6.1) are given by,

$$\begin{aligned} x(1) &= A \otimes x_0 \oplus B \otimes u(0) \\ x(2) &= A \otimes x(1) \oplus B \otimes u(1) \\ &= A^{\otimes 2} \otimes x_0 \oplus A \otimes B \otimes u(0) \oplus B \otimes u(1) \\ &\vdots \end{aligned}$$

The general formula is

$$x(t, x_0, U(t-1)) = A^{\otimes t} \otimes x_0 \oplus \bigoplus_{i=0}^{t-1} A^{\otimes t-i-1} \otimes B \otimes u(i). \tag{6.2}$$

The corresponding formula for the output of the system is

$$y(t, x_0, U(t-1)) = CA^{\otimes t} \otimes x_0 \oplus \bigoplus_{i=0}^{t-1} CA^{t-i-1} \otimes B \otimes u(i). \tag{6.3}$$

The output vector $Y(T, x_0, U(T-1)) = [y(0)^\top \;\; y(1)^\top ... \;\; y(T)^\top]^\top$, generated by applying the input sequence $U(T-1)$ to the initial state x_0, can be expressed in a similar manner to that presented in Chapter 4, with slight modifications, as there is no direct feed-through from the input to the output in the model here. That is,

$$Y(T, x_0, U(T-1)) = \mathcal{O}_T \otimes x_0 \oplus \mathcal{M}_T \otimes U(T-1) \tag{6.4}$$

where

$$\mathcal{O}_T = \begin{bmatrix} C \\ C \otimes A \\ \vdots \\ C \otimes A^{\otimes T} \end{bmatrix}, \tag{6.5}$$

$$\mathcal{M}_T = \begin{bmatrix} -\infty & -\infty & -\infty & ... \\ C \otimes B & -\infty & -\infty & ... \\ C \otimes A \otimes B & C \otimes B & -\infty & ... \\ \vdots & \vdots & \vdots & \ddots \\ C \otimes A^{\otimes T-1} \otimes B & C \otimes A^{\otimes T-2} \otimes B & ... & C \otimes B \end{bmatrix} \tag{6.6}$$

are the max-plus observability and input-output matrices, respectively.

6.3 System Properties For Max-plus Systems

In [50] and [21], the notions of reachability, controllability, and observability have been extended to max-plus algebra. In comparison to the conventional LTI setting, these concepts are more restrictive when defined for the system (6.1). In what follows, we focus on the weaker forms of reachability and observability defined in [21] for the system (6.1).

6.3.1 Reachability

As we saw for the LTI case in Chapter 3, the concept of reachability addresses the question of whether a system can be steered from the origin to a specified final state with an appropriate sequence of inputs. For max-plus linear systems,

the transfer to an arbitrary state is not always possible except for very specific cases. Consequently, the set of reachable states is rarely all of $\mathbb{R}^n_{max}$ [21]. Using (6.2), we can write the state of the system (6.1) at time q as

$$x(q, x_0, U(q-1)) = A^{\otimes q} \otimes x_0 \oplus \begin{bmatrix} B & A \otimes B & \dots & A^{q-1} \otimes B \end{bmatrix} \otimes \begin{bmatrix} u(q-1) \\ u(q-2) \\ \vdots \\ u(0) \end{bmatrix}$$

where $M_{R,q}$ is the reachability matrix

$$M_{R,q} = [B \quad A \otimes B \quad A^{\otimes 2} \otimes B ... \quad A^{\otimes q-1} \otimes B].$$

Here, we consider the following definition of a reachable state.

Definition 6.3.1. *(Reachable State [50]) Given the system* (6.1)*, a state* $x_f \in \mathbb{R}^n_{max}$ *is* reachable *in* q *steps if there exists an initial state* x_0 *and a sequence of inputs* $U(q-1)$ *such that the solution of the system satisfies* $x(q, x_0, U(q-1)) = x_f$*.*

Definition 6.3.1 says that a state x_f is reachable in the max-plus setting if there exists an initial state x_0 and an input sequence $U(q-1)$ such that we can drive the system (6.1) from this initial state x_0 to x_f. The collection of reachable states of a given system (6.1) leads to the following definition.

Definition 6.3.2. *(Reachable Set [21]) Given a set of initial states* $X_0 \in \mathbb{R}^n_{max}$*, and a positive integer* q*, let* $R_f(q, X_0)$ *be the set of all states* $x \in \mathbb{R}^n_{max}$ *that can be reached in* q *steps from some initial state* $x_0 \in X_0$ *with an appropriate sequence of inputs* $U(q-1)$*.*

To describe a necessary and sufficient condition for a state to be reachable, we first need to introduce two operations. For a matrix $A \in \mathbb{R}^{n \times m}$, the operation †, referred to as *conjugation* in [15], involves negation and the matrix transpose operation. Formally, for $A = \{a_{ij}\}$, $A^\dagger = \{-a_{ji}\}$. The second operation $\otimes'$ is defined analogously to the max-plus matrix product only with the min operation replacing the max operation. Now consider the following theorem.

Theorem 6.3.1. *([21]) Given an initial state* $x_0 \in \mathbb{R}^n_{max}$*, and a state* $x_f \in \mathbb{R}^n_{max}$*, then* $x_f \in R_f(q, x_0)$ *if and only if,*

$$M_{R,q} \otimes (M^\dagger_{R,q} \otimes' x_f) \oplus A^{\otimes q} x_0 = x_f.$$

Further, when this holds, the input sequence $U(q-1) = M_{R,q}^{\dagger} \otimes' x_f$ *drives the system state from* x_0 *to* $x(q, x_0, U(q-1)) = x_f$.

Theorem 6.3.1 provides a necessary and sufficient condition for a state to be reachable for the system (6.1). Moreover, it explicitly describes an input sequence that is independent of the initial state that drives the system from x_0 to x_f in q steps. In Example 6.3.2, we illustrate how this result can be used to check if a state is reachable in q-steps.
For conventional LTI systems, reachability guarantees that any state in $\mathbb{R}^n$ can be reached from the origin, meaning that the set of reachable states is $\mathbb{R}^n$ for $X_0 = \{0\}$. In the max-plus setting, the system law,

$$x(t+1) = A \otimes x(t) \oplus B \otimes u(t)$$

can only lead to states at time q that are entrywise greater than or equal to the *unforced terminal state* $A^{\otimes q} \otimes x_0$. With this in mind, the authors of [21] consider a more restricted form of reachability, where it is possible to reach a state whose components are strictly greater than the unforced terminal state and call such systems *weakly reachable*. We next recall the definition. Note that we use the notation $x_i(q, x_0, U(q-1))$ for the i^{th} component of $x(q, x_0, U(q-1))$.

Definition 6.3.3. *(q-step weakly reachable [21])*
The system (6.1) *is q-step weakly reachable if given any* $x_0 \in X_0$*, an input sequence* $U(q-1)$ *exists such that each component of the final state* $x(q, x_0, U(q-1))$ *is strictly greater than the unforced terminal state* $A^{\otimes q} \otimes x_0$*; formally,* $x_i(q, x_0, U(q-1)) > (A^{\otimes q} \otimes x_0)_i$ *for* $i = 1, 2, ..., n$.

In [21], a condition for weak reachability is given in terms of a matrix property known as *row-asticity*. We now recall this.

Definition 6.3.4. *(Row-astic [1])* $A \in \mathbb{R}_{max}^{n \times m}$ *is row-astic if for each row* $i = 1, 2, ..., n$*, we have,* $\bigoplus_{j=1}^{m} a_{ij} \in \mathbb{R}$.

Definition 6.3.4 states that a matrix $A \in \mathbb{R}_{max}^{n \times m}$ is row-astic if every row contains at least one finite entry. Similarly, a matrix is *column-astic* if every column contains at least one finite entry. This will be useful when we discuss the idea of observability in the max-plus setting.
The next theorem establishes the connection between row-asticity of the reachability matrix $M_{R,q}$ and weakly reachable systems.

Theorem 6.3.2. *([21]) A system* (6.1) *is q-step weakly reachable if and only if* $M_{R,q}$ *is row-astic.*

To clarify the notion of weak reachability, consider the following example.

Example 6.3.1. *Consider the system*

$$x(t+1) = \begin{bmatrix} 0 & -\infty \\ 1 & 0 \end{bmatrix} \otimes x(t) \oplus \begin{bmatrix} 0 \\ -1 \end{bmatrix} \otimes u(t),$$

where, $A = \begin{bmatrix} 0 & -\infty \\ 1 & 0 \end{bmatrix}$ *and* $B = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$. *We check that this system is* 2*-step weakly reachable using Theorem 6.3.2. First we need to calculate,* $M_{R,2} = [B \quad A \otimes B]$. *Note that,*

$$A \otimes B = \begin{bmatrix} 0 & -\infty \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Hence,

$$M_{R,2} = \begin{bmatrix} 0 & 0 \\ -1 & 1 \end{bmatrix}.$$

As $M_{R,2}$ *contains all finite entries,* $M_{R,2}$ *is row-astic and Theorem 6.3.2 implies that the system is* 2*-step weakly reachable. To demonstrate Definition 6.3.3, consider the initial state* $x_0 = \begin{bmatrix} 5 & 2 \end{bmatrix}^\top$. *We first need to calculate the unforced terminal state* $A^{\otimes 2} \otimes x_0$. *Note that,*

$$A^{\otimes 2} = \begin{bmatrix} 0 & -\infty \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & -\infty \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\infty \\ 1 & 0 \end{bmatrix}.$$

Hence the unforced terminal state is given by,

$$A^{\otimes 2} \otimes x_0 = \begin{bmatrix} 0 & -\infty \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 5 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \end{bmatrix}.$$

Now consider the input vector $U(1) = [2, 8]$. *At step* 1*:*

$$\begin{aligned} x(1, x_0, U(0)) &= A \otimes x_0 \oplus B \otimes u(0) \\ &= \begin{bmatrix} 0 & -\infty \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 5 \\ 2 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ -1 \end{bmatrix} \otimes 2 \\ &= \begin{bmatrix} 5 \\ 6 \end{bmatrix} \oplus \begin{bmatrix} 2 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 5 \\ 6 \end{bmatrix}. \end{aligned}$$

At step 2 *and using* (6.2)*:*

$$\begin{aligned} x(2, x_0, U(1)) &= A^{\otimes 2} \otimes x_0 \oplus A \otimes B \otimes u(0) \oplus B \otimes u(1) \\ &= \begin{bmatrix} 0 & -\infty \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 5 \\ 2 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes 2 \oplus \begin{bmatrix} 0 \\ -1 \end{bmatrix} \otimes 8 \\ &= \begin{bmatrix} 8 \\ 7 \end{bmatrix} \end{aligned}$$

Clearly,

$$\begin{bmatrix} 8 \\ 7 \end{bmatrix} > \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

componentwise. Therefore, for this system, we have shown that for $x_0 = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$*, the input vector* $U(1)$ *is such that* $x_i(2, x_0, U(1)) > (A^{\otimes 2} \otimes x_0)_i$ *for* $i = 1, 2$.

6.3.2 Controllability

Versions of the concept of controllability have also been studied in the max-plus setting. As with reachability, the formulation of controllability for max-plus systems differs slightly from that for LTI systems, as given in Chapter 3. We will work with a definition given in [50]. Loosely speaking, a state $x_f \in \mathbb{R}^n_{max}$ is controllable if there exists a sequence of inputs that can drive the system (6.1) from the zero state $x_0 = [-\infty \ \ldots \ -\infty]^\top$ to x_f in some finite number of steps. This is formalised in the following definition.

Definition 6.3.5. *(Controllable state [50])*
Given the system (6.1)*, a state* $x_f \in \mathbb{R}^n_{max}$ *is* controllable *if there exist a*

nonnegative integer q and an input sequence $U(q-1)$ such that with $x_0 = [-\infty \ \ldots \ -\infty]^\top$, $x(q, x_0, U(q-1)) = x_f$.

From Definitions 6.3.5 and 6.3.1, it is clear that every controllable state is also a reachable state. However, note that the converse does not always hold. In the following example, we show a case where a state is reachable but not controllable.

Example 6.3.2. *Consider the system*

$$x(t+1) = \begin{bmatrix} 0 & -1 \\ -2 & 0 \end{bmatrix} \otimes x(t) \oplus \begin{bmatrix} 0 \\ 0 \end{bmatrix} \otimes u(t), \tag{6.7}$$

where

$$A = \begin{bmatrix} 0 & -1 \\ -2 & 0 \end{bmatrix} \text{ and } B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

We will show that the state $x_f = \begin{bmatrix} 0 & -2 \end{bmatrix}^\top$ is reachable in some finite number of steps q, but not controllable for any q. Consider the set of initial states

$$X_0 = \{x_0 = \begin{bmatrix} 0 \\ \alpha \end{bmatrix} : \alpha \leq -2\}$$

and select an input $u(0) \leq -2$, we can show that x_f is reachable in 1-step. Consider the initial state $x_0 = \begin{bmatrix} 0 & -7 \end{bmatrix}^\top$. We will use Theorem 6.3.1 to show that x_f is reachable in 1-step. To do this, we need to verify the equation

$$M_{R,1} \otimes (M_{R,1}^\dagger \otimes' x_f) \oplus A^{\otimes 1} x_0 = x_f. \tag{6.8}$$

For this example,

$$M_{R,1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, M_{R,1}^\dagger = \begin{bmatrix} 0 & 0 \end{bmatrix}, \text{ and } A^{\otimes 1} x_0 = \begin{bmatrix} 0 & -1 \\ -2 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ -7 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \end{bmatrix}.$$

Substituting these matrices into (6.8) we see that

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \otimes (\begin{bmatrix} 0 & 0 \end{bmatrix} \otimes' \begin{bmatrix} 0 \\ -2 \end{bmatrix}) \oplus \begin{bmatrix} 0 \\ -2 \end{bmatrix} \quad = \quad \begin{bmatrix} 0 \\ 0 \end{bmatrix} \otimes (-2) \oplus \begin{bmatrix} 0 \\ -2 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$$

Therefore, by Theorem 6.3.1, x_f is reachable in 1-step using the input $u(0) =$

$M_{R,1}^{\dagger} \otimes' x_f$, *that is,*

$$u(0) = \begin{bmatrix} 0 & 0 \end{bmatrix} \otimes' \begin{bmatrix} 0 \\ -2 \end{bmatrix} = -2.$$

This can also be verified directly using (6.7).

$$\begin{aligned} x(1, x_0, -2) &= \begin{bmatrix} 0 & -1 \\ -2 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ -7 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 0 \end{bmatrix} \oplus -2 \\ &= \begin{bmatrix} 0 \\ -2 \end{bmatrix} \oplus \begin{bmatrix} -2 \\ -2 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ -2 \end{bmatrix}. \end{aligned}$$

We next show that x_f is not controllable in q-steps for any q. First note that $A \otimes B = B$. This implies that $M_{R,q}$ takes the simple form

$$\begin{bmatrix} B & B \cdots & B \end{bmatrix}.$$

Using the system law (6.7) *with the initial state $x_0 = \begin{bmatrix} -\infty & -\infty \end{bmatrix}^{\top}$ and any given input sequence $U(q-1)$ it follows by a simple calculation that*

$$x(q, x_0, U(q-1)) = \begin{bmatrix} max(u(0), \ldots, u(q-1)) \\ max(u(0), \ldots, u(q-1)) \end{bmatrix},$$

so that all states have the same element in both of their entries, i.e., $x_1(q) = x_2(q)$. So we can't make $x_1(q) = 0$ and $x_2(q) = -2$. Therefore, x_f is not controllable.

6.3.3 Observability

Recall that the concept of *observability* in the LTI system setting is concerned with the ability to uniquely determine the initial state corresponding to a given sequence of outputs $Y(T, x_0, U(T-1))$ and inputs $U(T-1)$. In the max-plus setting, the condition for state observability is again more restrictive than the LTI case. Because addition in max-plus algebra is idempotent, it is often impossible to uniquely determine the actual system state. Hence, the authors of [21] considered whether it is possible to identify the latest initial state consistent with a given output sequence. The latest state represents the latest comple-

tion times consistent with the observed outputs. Now consider the following definition of the *latest event-time state*

Definition 6.3.6. *(Latest Event-time State [21])*
Given a sequence of outputs, $Y(q, x_0, U(q-1))$, with a known sequence of inputs $U(q-1)$, the latest event-time state γ_0 which results in $Y(q, x_0, U(q-1))$ is

$$\gamma_0 = \max\{x \in \mathbb{R}^n_{max} : Y(q, x, U(q-1)) = \mathcal{O}_q \otimes x_0 \oplus \mathcal{M}_q \otimes U(q-1)\},$$

where the max is over each component.

Note that the latest event-time state may not be finite. When this happens, it provides no information about the initial state of the system. Hence, the definition of weak observability below excludes this case by requiring the latest event-time state to be finite [21]. The example below demonstrates the idea of γ_0.

Example 6.3.3. *Consider the system,*

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 0 & -\infty \\ 0 & 2 \end{bmatrix} \otimes x(t) \oplus \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \otimes x(t). \end{aligned}$$

Let the input $u(0) = 0$ and the given 2-length output sequence be $Y(1, x_0, U(0)) = [y(0) \;\; y(1)]^\top = [2 \;\; 3]^\top$. Note that for

$$Y(1, x_0, U(0)) = \begin{bmatrix} C \\ C \otimes A \end{bmatrix} x_0 \oplus \begin{bmatrix} -\infty & -\infty \\ C \otimes B & -\infty \end{bmatrix} \begin{bmatrix} 0 \\ u(1) \end{bmatrix},$$

the input $u(1)$ does not influence the outputs due to no direct input–output mapping in (6.1).
We will now calculate the latest event-time state

$$\gamma_0 = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}.$$

From the system above we have,

$$y(0, \gamma_0, U(\cdot)) = 2 = \max(\gamma_1 + 1, \gamma_2),$$

such that $U(\cdot) = \emptyset$. *Then we must have,*

$$\gamma_1 + 1 \leq 2 \Rightarrow \gamma_1 \leq 1 \quad \text{and} \quad \gamma_2 \leq 2, \tag{6.9}$$

with at least $\gamma_1 + 1$ *or* γ_2 *equal to* 2.
Now we need to calculate $x(1, x_0, U(0)) = \begin{bmatrix} x_1(1) \\ x_2(1) \end{bmatrix}$ *in regards to the latest event-time state. We have,*

$$\begin{aligned} x(1, \gamma_0, U(0)) &= \begin{bmatrix} 0 & -\infty \\ 0 & 2 \end{bmatrix} \otimes \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes 0 \\ &= \begin{bmatrix} \gamma_1 \\ \max(\gamma_1, \gamma_2 + 2) \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \max(\gamma_1, 1) \\ \max(\gamma_1, \gamma_2 + 2, 1) \end{bmatrix} = \begin{bmatrix} x_1(1) \\ x_2(1) \end{bmatrix}. \end{aligned}$$

Hence, we calculate,

$$\begin{aligned} y(1, \gamma_0, U(0)) &= C \otimes x(1, \gamma_0, U(0)) \\ &= \max(1 + x_1(1), 0 + x_2(1)) \end{aligned}$$

Note that, $1 + x_1(1) = \max(\gamma_1 + 1, 2)$ *and* $0 + x_2(1) = \max(\gamma_1, \gamma_2 + 2, 1)$. *Then we have,*

$$\begin{aligned} y(1, \gamma_0, U(0)) &= \max(\gamma_1 + 1, 2, \gamma_1, \gamma_2 + 2, 1) \\ &= \max(\gamma_1 + 1, 2, \gamma_2 + 2). \end{aligned}$$

Then this implies that,

$$\gamma_1 + 1 \leq 3 \Rightarrow \gamma_1 \leq 2 \quad \text{and} \quad \gamma_2 + 2 \leq 3 \Rightarrow \gamma_2 \leq 1, \tag{6.10}$$

with at least $\gamma_1 + 1$ *or* $\gamma_2 + 2$ *equal to* 3.
Combining the inequalities in (6.9) (6.10) *we must have* $\gamma_1 \leq 1$ *and* $\gamma_2 \leq 1$. *Hence, the latest event-time state* γ_0 *that produces the given output sequence* $Y(1, x_0, U(0))$ *is*

$$\gamma_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

which is finite. We can compute $Y(1, \gamma_0, U(0))$ *to check if we get the given outputs* $Y(1, x_0, U(0)) = [2 \;\; 3]^\top$. *At* $t = 0$*:*

$$y(0, \gamma_0, U(\cdot)) = C \otimes \gamma_0 = \begin{bmatrix} 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 2.$$

At $t = 1$*:*

$$\begin{aligned} x(1, \gamma_0, U(0)) &= \begin{bmatrix} 0 & -\infty \\ 0 & 2 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes 0 \\ &= \begin{bmatrix} 1 \\ 3 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 3 \end{bmatrix} \end{aligned}$$

and,

$$y(1, \gamma_0, U(0)) = \begin{bmatrix} 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 3 \end{bmatrix} = 3.$$

Therefore we have, $Y(1, \gamma_0, U(0)) = Y(1, x_0, U(0))$.

Definition 6.3.7. *(q-step Weakly Observable [21])*
The system (6.1) *is q-step weakly observable if for any given output sequence* $Y(q, x_0, U(q-1))$*, the latest event-time state* γ_0 *is finite and can be computed from* $Y(q, x_0, U(q-1))$ *and* $U(q-1)$.

A necessary and sufficient condition for a system (6.1) to be q-step weakly observable is given next.

Theorem 6.3.3. *([21]) A system* (6.1) *is q-step weakly observable if and only if* $\mathcal{O}_{q-1}$ *is column-astic.*

The example below illustrates how we can check if a system (6.1) is q-step weakly observable using Theorem 6.3.3.

Example 6.3.4. *Consider the max-plus linear system,*

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 0 & -\infty \\ 0 & 2 \end{bmatrix} \otimes x(t) \oplus \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \otimes x(t). \end{aligned}$$

We can use Theorem 6.3.3 to check if this system is 2*-step weakly observable. To do this, we first calculate the observability matrix,*

$$\mathcal{O}_1 = \begin{bmatrix} C \\ C \otimes A \end{bmatrix}.$$

Note that,

$$C \otimes A = \begin{bmatrix} 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & -\infty \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \end{bmatrix}.$$

Hence the observability matrix is,

$$\mathcal{O}_1 = \begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix}.$$

As $\mathcal{O}_1$ contains all finite entries, it is column-astic. Therefore, by Theorem 6.3.3, this system is 2*-step weakly observable.*

6.4 Opacity for Max-plus Linear Systems

In this section, we present some initial thoughts on discussing how opacity may be formulated in a max-plus setting and its relation to reachability. We shall focus on the notion of k-initial state opacity (k-ISO), as described in [52]. Note that the concept of weak k-ISO won't be relevant in the discussion, so we will refer to strong k-initial state opacity simply as *k-initial state opacity*. As in the LTI case, we will let $K \subseteq \mathbb{Z}_+$ denote the instants of time at which the intruder makes an observation of the system (6.1). Based on the outputs of the system (6.1) at times $k \in K$, the intruder attempts to discover whether or not the system started from a state that belongs to the set of secret initial states X_s. The property of k-ISO means that it is not possible for the intruder to determine this with certainty. As before, we assume that the intruder has knowledge of the initial sets of secret and non-secret states, X_s and X_{ns} respectively, where $X_s \subset X_0$ and $X_{ns} = X_0 \backslash X_s$. In addition, the intruder is assumed to know the system matrices A, B and C.
For the sake of completeness, we now recall the definition of k-ISO formulated for the system (6.1).

Definition 6.4.1. *(k-Initial State Opacity ([53]))*
For the system (6.1)*, given $X_s, X_{ns} \subseteq X_0$, and $k \in K$, X_s is strongly k-initial*

state opaque (k-ISO) with respect to X_{ns} if for every $x_s \in X_s$ and for every sequence of inputs $U_s(k-1)$, there exists an $x_{ns} \in X_{ns}$ and a sequence of inputs $U_{ns}(k-1)$ such that we have,

$$y(k, x_s, U_s(k-1)) = y(k, x_{ns}, U_{ns}(k-1)).$$

X_s is strongly K-ISO with respect to X_{ns} if X_s is strongly k-ISO with respect to X_{ns}, $\forall\ k \in K$ [53]. Definition 6.4.1 means the following. Suppose we start from any secret state $x_s \in X_s$ and apply any sequence of inputs $U_s(k-1)$. When an intruder observes the outputs of the system (6.1) at any time $k \in K$, the observation will not be unique. There will be an identical observation that comes from a state reached by applying some sequence of inputs $U_{ns}(k-1)$, starting from $x_{ns} \in X_{ns}$. To illustrate how the definition of k-ISO works in the max-plus framework, consider the following example.

Example 6.4.1. *Consider the following max-plus linear system,*

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 1 & 2 \\ -\infty & 1 \end{bmatrix} \otimes x(t) \oplus \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes u(t) \\ y(t) &= \begin{bmatrix} 1 & 1 \end{bmatrix} \otimes x(t) \end{aligned}$$

Note that we are going to use slightly different notation to keep things simpler for this example. We denote $x(k)$ and $y(k)$ as the state and output at time k (i.e. the time when an intruder takes an observation). We are going to show that the system satisfies k-ISO for all $k \in K$, where $K = \{1, 2\}$. Consider the set of secret states that only contains one element such that $X_s = \{\begin{bmatrix} 1 \\ 1 \end{bmatrix}\}$ and a non-secret state $x_{ns} \in X_{ns}$, where $x_{ns} = \begin{bmatrix} -2 & 0 \end{bmatrix}^\top$. Then, the system satisfies K-initial state opacity as, for k=1, we have the secret state,

$$\begin{aligned} x_s(1) &= \begin{bmatrix} 1 & 2 \\ -\infty & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes u_s(0) \\ &= \begin{bmatrix} 3 \\ 2 \end{bmatrix} \oplus \begin{bmatrix} 1 \otimes u_s(0) \\ 1 \otimes u_s(0) \end{bmatrix} \\ &= \begin{bmatrix} 3 \oplus (1 + u_s(0)) \\ 2 \oplus (1 + u_s(0)) \end{bmatrix}. \end{aligned}$$

The output is given by,

$$\begin{aligned} y_s(1) &= \begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 3 \oplus (1 + u_s(0)) \\ 2 \oplus (1 + u_s(0)) \end{bmatrix} \\ &= \max(4, 2 + u_s(0), 3, 2 + u_s(0)) \\ &= \max(4, 2 + u_s(0)). \end{aligned}$$

Starting from the non-secret state,

$$\begin{aligned} x_{ns}(1) &= \begin{bmatrix} 1 & 2 \\ -\infty & 1 \end{bmatrix} \otimes \begin{bmatrix} -2 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_{ns}(0) \\ &= \begin{bmatrix} 2 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \otimes u_{ns}(0) \\ 1 \otimes u_{ns}(0) \end{bmatrix} \\ &= \begin{bmatrix} 2 \oplus (1 + u_{ns}(0)) \\ 1 \oplus (1 + u_{ns}(0)) \end{bmatrix}, \end{aligned}$$

with the corresponding non-secret state,

$$\begin{aligned} y_{ns}(1) &= \begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 2 \oplus (1 + u_{ns}(0)) \\ 1 \oplus (1 + u_{ns}(0)) \end{bmatrix} \\ &= 3 \oplus 2 + u_{ns}(0) \oplus 2 \oplus 2 + u_{ns}(0) \\ &= \max(3, 2 + u_{ns}(0)). \end{aligned}$$

Using this, we can show that k-ISO is satisfied for $k = 1$. To see this, note that ensuring $y_s(1) = y_{ns}(1)$ can be achieved in both cases: if $y_s(1) = 4$, we can choose $u_{ns}(0) = 2$, which yields $y_{ns}(1) = 4$ as well. Alternatively, if $y_s(1) = 2 + u_s(0)$, then setting $u_{ns}(0) = u_s(0)$ guarantees equality. Therefore, k-ISO holds.

For $k = 2$ we have,

$$\begin{aligned} x_s(2) &= \begin{bmatrix} 1 & 2 \\ -\infty & 1 \end{bmatrix} \otimes \begin{bmatrix} 3 \oplus 1 + u_s(0) \\ 2 \oplus 1 + u_s(0) \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes u_s(1) \\ &= \begin{bmatrix} 4 \oplus 3 + u(0) \\ 3 \oplus 2 + u(0) \end{bmatrix} \oplus \begin{bmatrix} 1 + u_s(1) \\ 1 + u_s(1) \end{bmatrix} \\ &= \begin{bmatrix} 4 \oplus 3 + u_s(0) \oplus 1 + u_s(1) \\ 3 \oplus 2 + u_s(0) \oplus 1 + u_s(1) \end{bmatrix}, \end{aligned}$$

with the output,

$$\begin{aligned} y_s(2) &= \begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 4 \oplus 3 + u_s(0) \oplus 1 + u_s(1) \\ 3 \oplus 2 + u_s(0) \oplus 1 + u_s(1) \end{bmatrix} \\ &= 5 \oplus 4 + u_s(0) \oplus 2 + u_s(1) \oplus 4 \oplus 3 + u_s(0), 2 + u_s(1) \\ &= \max(5, 4 + u_s(0), 2 + u_s(1)). \end{aligned}$$

The non-secret state is given by

$$\begin{aligned} x_{ns}(2) &= \begin{bmatrix} 1 & 2 \\ -\infty & 1 \end{bmatrix} \otimes \begin{bmatrix} 2 \oplus 1 + u_{ns}(0) \\ 1 \oplus 1 + u_{ns}(0) \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes u_{ns}(1) \\ &= \begin{bmatrix} 3 \oplus 3 + u_{ns}(0) \\ 2 \oplus 2 + u_{ns}(0) \end{bmatrix} \oplus \begin{bmatrix} 1 + u_{ns}(1) \\ 1 + u_{ns}(1) \end{bmatrix} \\ &= \begin{bmatrix} 3 \oplus 3 + u_{ns}(0) \oplus 1 + u_{ns}(1) \\ 2 \oplus 2 + u_{ns}(0) \oplus 1 + u_{ns}(1) \end{bmatrix}, \end{aligned}$$

with the corresponding output,

$$\begin{aligned} y_{ns}(2) &= \begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 3 \oplus 3 + u_{ns}(0) \oplus 1 + u_{ns}(1) \\ 2 \oplus 2 + u_{ns}(0) \oplus 1 + u_{ns}(1) \end{bmatrix} \\ &= 4 \oplus 4 + u_{ns}(0) \oplus 2 + u_{ns}(1) \oplus 3 \oplus 3 + u_{ns}(0), 2 + u_{ns}(1) \\ &= \max(4, 4 + u_{ns}(0), 2 + u_{ns}(1)). \end{aligned}$$

k-ISO is also satisfied for $k = 2$. $y_s(2) = y_{ns}(2)$ can be achieved all three cases: if $y_s(1) = 5$, we can choose $u_{ns}(0) = 1$ and $u_{ns}(1) \leq 3$, which yields $y_{ns}(1) = 5$. Alternatively,for the other two cases, if $y_s(1) = 4 + u_s(0)$ or $2 + u_s(1)$, then setting $u_{ns}(0) = u_s(0)$ and $u_{ns}(1) = u_s(1)$ guarantees equality. Therefore, k-ISO holds. Therefore, our example satisfies K-ISO for $K = \{1, 2\}$.

6.4.1 Opacity and Reachability.

Recall that in Chapter 3, opacity was characterized in terms of reachability, with the authors of [53] characterizing k-ISO using the outputs associated with reachable states from the sets of secret and non-secret initial states, X_s and X_{ns}, respectively. It is natural to question if something similar can be done for max-plus systems. We present some initial thoughts in this direction here.
Let $x_s \in X_s$ and consider the state $x(k, x_s, U_s(k-1))$. One simple way to ensure

opacity is to show that $x(k, x_s, U_s(k-1))$ also lies in the reachable set of some non-secret initial state $x_{ns} \in X_{ns}$: formally, $x(k, x_s, U_s(k-1)) \in R_f(k, x_{ns})$. Using Theorem 6.3.1, $x(k, x_s, U_s(k-1)) \in R_f(k, x_{ns})$ is equivalent to

$$M_{R,k} \otimes (M_{R,k}^{\dagger} \otimes' x(k, x_s, U_s(k-1))) \oplus A^{\otimes k} x_{ns} = x(k, x_s, U_s(k-1)).$$

Interestingly, Theorem 6.3.1 implies that the same input sequence

$$U_s(k-1) = M_{R,k}^{\dagger} \otimes' x(k, x_s, U_s(k-1)) = U_{ns}(k-1)$$

can be applied to drive the system from both x_s and x_{ns} to $x(k, x_s, U_s(k-1))$. This follows because the input sequence given in Theorem 6.3.1 is independent of the initial state. It is important to emphasize that this condition is rather restrictive, as it requires exact matching of states rather than outputs at time k. Thus, it is sufficient but not necessary for opacity.
When the input sequence $U_s(k-1)$ is applied to the secret initial state x_s, the corresponding output is given by $C \otimes x(k, x_s, U_s(k-1))$. From Theorem 6.3.1,

$$x(k, x_s, U_s(k-1)) = M_{R,k} \otimes (M_{R,k}^{\dagger} \otimes' x(k, x_s, U_s(k-1))) \oplus A^{\otimes k} x_s$$

which implies that

$$C \otimes x(k, x_s, U_s(k-1)) = C \otimes M_{R,k} \otimes (M_{R,k}^{\dagger} \otimes' x(k, x_s, U_s(k-1))) \oplus C \otimes A^{\otimes k} x_s.$$

For k-ISO to hold, there must exist some $x_{ns} \in X_{ns}$, and some input sequence $U_{ns}(k-1)$ such that

$$C \otimes x(k, x_s, U_s(k-1)) = C \otimes x(k, x_{ns}, U_{ns}(k-1)).$$

This can be formulated as

$$\begin{aligned} &C \otimes M_{R,k} \otimes (M_{R,k}^{\dagger} \otimes' x(k, x_s, U_s(k-1))) \oplus C \otimes A^{\otimes k} x_s \\ &= C \otimes M_{R,k} \otimes (M_{R,k}^{\dagger} \otimes' x(k, x_{ns}, U_{ns}(k-1))) \oplus C \otimes A^{\otimes k} x_{ns}. \end{aligned}$$

In Chapter 3, within the LTI system setting, we saw in the works of [53] that k-ISO was characterised in terms of outputs of the reachable states from X_s and X_{ns}. We can also do this in the max-plus setting. Let's denote the output

reachable set at time k from a given x_0 by

$$\mathcal{Y}(k, x_0) = \{C \otimes x_f : x_f \in R_f(k, x_0)\}.$$

For the set of secret initial states $X_s \subset X_0$, we write,

$$R_f(k, X_s) = \bigcup_{x_s \in X_s} R_f(k, x_s)$$

and,

$$\mathcal{Y}(k, X_s) = \bigcup_{x_s \in X_s} \mathcal{Y}(k, x_s) = C \otimes R_f(k, X_s).$$

For k-ISO to hold, it is required for all $y_s \in \mathcal{Y}(k, X_s)$ there must exist $x_{ns} \in X_{ns}$ such that $y_s \in \mathcal{Y}(k, x_{ns})$. In general, k-ISO is equivalent to

$$\mathcal{Y}(k, X_s) \subseteq \mathcal{Y}(k, X_{ns}).$$

which implies that,

$$R_f(k, X_s) \subseteq C^{-1}(C \otimes R_f(k, X_{ns})). \tag{6.11}$$

Then for $x_s \in R_f(k, X_s)$ there exists $x_s \in X_s$ such that

$$x(k, x_s, U_s(k-1) = M_{R,k} \otimes (M_{R,k}^{\dagger} \otimes' x(k, x_s, U_s(k-1))) \oplus A^{\otimes k} x_s. \tag{6.12}$$

Hence, the solution set of (6.12) must be contained in (6.11) for any $x_s \in X_s$.

6.5 Concluding Remarks

In this chapter, we first described linear systems in max-plus algebra, and the form of the solutions of the state equations (6.1). In particular, we recalled the observability and input–output matrices within the max-plus framework. Building on this, we discussed system properties such as reachability, controllability, and observability in the context of max-plus linear systems. In comparison to the LTI setting, these concepts become more restrictive, primarily due to the absence of an additive inverse and the idempotent nature of addition in the max-plus algebra. Instead, we described weaker versions of reachability and observability as defined in [21]. Lastly, we presented some initial thoughts on formulating opacity for max-plus linear systems, and outlined some aspects of

the relation between reachable states and k-ISO.

Chapter 7

Conclusions and Future Work

In this final chapter, we review and summarise the work presented in the thesis. We also highlight some open questions for future work based on it.

7.1 Summary

In Chapter 2, we reviewed some of the main notions of opacity for discrete event systems. We saw that opacity provided a formal framework to specify and verify whether certain secret states or behaviours can be inferred by an external observer based on their observations of the system's events. Our discussion focused on state-based approaches, highlighting how each captures different aspects of information that must remain hidden from an intruder within a partially observed DES. We also briefly discussed verification and enforcement problems for opacity in DES. Lastly, we recalled the definitions of controllability and observability in this setting, as these emerge as recurring themes throughout the thesis.

Chapter 3 provided an overview of how opacity was recently formulated for LTI systems. We looked at the idea of k-ISO and explored its connection to several key system properties such as controllability, reachability, observability, and output controllability. In particular, we saw that controllability and output controllability over the time interval $[0, k]$ implied the existence of a set of secret initial states X_s that satisfies k-ISO. We noted an issue with the first definition of K-ISO, by showing that it is possible to have a system that satisfies K-ISO that is also observable. In this chapter, we also examined k-ISO using backward

reachable sets. We strengthened a result from [53] characterising k-ISO in terms of backward reachable sets, by showing that only one of the two conditions from that paper is required.

Chapter 4 focused on security and attack detection concepts for LTI systems. In it, we discussed undetectable attacks and examined them using side initial state information. We also described a design procedure for a consistent detector, where this detector can detect all attacks that don't belong to the set of undetectable attacks. In the final sections of the chapter, we considered the trade-off between undetectable attacks and opacity. We highlighted results from [34], showing that it is not possible for a system to satisfy opacity without being vulnerable to undetectable attacks, and we provided an alternative proof to this fact.

In Chapter 5 we recalled several fundamental aspects of max-plus algebra. To illustrate the use of max-plus linear systems in modelling and analysing systems where synchronization and timing are critical, we described a simple railway system inspired by [28]. We recalled the connection between matrices and graphs, showing that the max-plus powers of a general matrix A can provide information about the maximal weighted path between two nodes. We also reviewed methods of solving equations in the max-plus semiring.

In Chapter 6, we moved on to study some fundamental control-theoretic concepts for max-plus linear systems. This included generalising the solutions of the state equations and examining the observability and input–output matrices within the max-plus framework. Building on this, we discussed system properties such as reachability, controllability, and observability in the context of max-plus linear systems. These notions are more restrictive compared to their definitions in Chapter 3. We also presented some initial thoughts on opacity for max-plus linear systems. In this setting, we worked with the definition of k-ISO described in [52]. We then highlighted the relation between reachable states and k-ISO, providing some simple sufficient conditions for opacity.

7.2 Future Work

There are several questions that arise from the work described in this thesis.

- In Chapter 6, we gave some preliminary ideas on how opacity could be considered in a max-plus linear system setting. These initial observations open several directions for further investigation. In particular, it would

be worthwhile to explore the connections between opacity, controllability, and q-step weak observability.

- Since max-plus linear systems form a subclass of discrete event systems, one may consider if the other state-based approaches discussed in Chapter 2 can also be formulated in this framework.

- In Chapter 6, we reviewed the concepts of reachability, controllability, and observability for max-plus linear systems. These notions are restrictive due to the absence of additive inverses and the idempotent nature of addition in max-plus algebra. This naturally raises the question of whether it is possible to reformulate these concepts in a less restrictive manner, aligning them more closely with their counterparts in Chapter 3.

- In Chapter 4, we discussed security and attack detection concepts for LTI systems. A natural direction for future work is to extend these concepts to max-plus linear systems. Such an extension could reveal new challenges and insights, particularly in classifying undetectable attacks and understanding their relation to opacity.

Bibliography

[1] Hazem Abdul Abbas Al Bermanei. "Applications of max-plus algebra to scheduling". In: (2021).

[2] Majid Alizadeh, Nazila Azizi, Samireh Mahdavi, and Fouad Baghlani. "Unveiling the shadows: obstacles, consequences, and challenges of information opacity in healthcare systems". In: *Philosophy, Ethics, and Humanities in Medicine* 20.1 (2025), p. 6.

[3] François Baccelli, Guy Cohen, Geert Jan Olsder, and Jean-Pierre Quadrat. *Synchronization and linearity*. Vol. 1. Wiley New York, 1992.

[4] Eric Badouel, Marek Bednarczyk, Andrzej Borzyszkowski, Benoit Caillaud, and Philippe Darondeau. "Concurrent secrets". In: *Discrete Event Dynamic Systems* 17 (2007), pp. 425–446.

[5] Jiří Balun and Tomáš Masopust. "On opacity verification for discrete-event systems". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 2075–2080.

[6] Ravindra B Bapat. *Graphs and matrices*. Springer, 2010.

[7] Béatrice Bérard, Krishnendu Chatterjee, and Nathalie Sznajder. "Probabilistic opacity for Markov decision processes". In: *Information Processing Letters* 115.1 (2015), pp. 52–59.

[8] Jeremy W Bryans, Maciej Koutny, and Peter YA Ryan. "Modelling opacity using Petri nets". In: *Electronic Notes in Theoretical Computer Science* 121 (2005), pp. 101–115.

[9] Christos G Cassandras and Stéphane Lafortune. *Introduction to discrete event systems*. Springer, 2008.

[10] Franck Cassez, Jérémy Dubreil, and Hervé Marchand. "Synthesis of opaque systems with static and dynamic masks". In: *Formal Methods in System Design* 40 (2012), pp. 88–115.

[11] Yuan Chen, Soummya Kar, and José MF Moura. "Dynamic attack detection in cyber-physical systems with side initial state information". In: *IEEE Transactions on Automatic Control* 62.9 (2016), pp. 4618–4624.

[12] B Ciffler. "Scheduling general production systems using schedule algebra". In: *Naval Research Logistics Quarterly* 10.1 (1963), pp. 237–255.

[13] Raymond A Cuninghame–Green. *Minimax Algebra (Lecture Notes in Economics and Mathematical Systems 166)*. 1979.

[14] Raymond A Cuninghame-Green. "Describing industrial processes with interference and approximating their steady-state behaviour". In: *Journal of the Operational Research Society* 13.1 (1962), pp. 95–100.

[15] Raymond A Cuninghame-Green. "Projections in minimax algebra". In: *Mathematical Programming* 10.1 (1976), pp. 111–123.

[16] Bart De Schutter and Ton van den Boom. "Max-plus algebra and max-plus linear discrete event systems: An introduction". In: *2008 9th International Workshop on Discrete Event Systems*. IEEE. 2008, pp. 36–42.

[17] Bart De Schutter, Ton van den Boom, Jia Xu, and Samira S Farahani. "Analysis and control of max-plus linear discrete-event systems: An introduction". In: *Discrete Event Dynamic Systems* 30.1 (2020), pp. 25–54.

[18] Jérémy Dubreil, Philippe Darondeau, and Hervé Marchand. "Opacity enforcing control synthesis". In: *2008 9th international workshop on discrete event systems*. IEEE. 2008, pp. 28–35.

[19] Hoda A ElMaraghy and A Seleim. "Generating max-plus equations for efficient analysis of manufacturing flow lines". In: *Journal of Manufacturing Systems* 37 (2015), pp. 426–436.

[20] Vincenzo Eramo, Tiziana Fiori, Francesco G Lavacca, et al. "A max plus algebra based scheduling algorithm for supporting time triggered services in ethernet networks". In: *Computer Communications* 198 (2023), pp. 85–97.

[21] Michael J Gazarik and Edward W Kamen. "Reachability and observability of linear systems over max-plus". In: *Kybernetika* 35.1 (1999), pp. 2–12.

[22] Michel Gondran and Michel Minoux. *Graphs, dioids and semirings: new models and algorithms*. Springer, 2008.

[23] Rob MP Goverde. "The max-plus algebra approach to railway timetable design". In: *WIT Transactions on The Built Environment* 37 (1998).

[24] Ye Guo, Xiaoning Jiang, Chen Guo, Shouguang Wang, and Oussama Karoui. "Overview of opacity in discrete event systems". In: *IEEE Access* 8 (2020), pp. 48731–48741.

[25] Laurent Hardouin, Bertrand Cottenceau, Ying Shang, Jörg Raisch, et al. "Control and state estimation for max-plus linear systems". In: *Foundations and Trends in Systems and Control* 6.1 (2018), pp. 1–116.

[26] Laurent Hardouin, Mehdi Lhommeau, and Ying Shang. "Towards geometric control of max-plus linear systems with applications to manufacturing systems". In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE. 2011, pp. 1149–1154.

[27] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. "Differential privacy techniques for cyber physical systems: A survey". In: *IEEE Communications Surveys & Tutorials* 22.1 (2019), pp. 746–789.

[28] Bernd Heidergott, Geert Jan Olsder, and Jacob Van Der Woude. *Max Plus at work: modeling and analysis of synchronized systems: a course on Max-Plus algebra and its applications*. Vol. 48. Princeton University Press, 2014.

[29] Aleksey Imaev and Robert P Judd. "Hierarchial modeling of manufacturing systems using max-plus algebra". In: *2008 American Control Conference*. IEEE. 2008, pp. 471–476.

[30] Romain Jacob, Jean-Jacques Lesage, and Jean-Marc Faure. "Overview of discrete event systems opacity: Models, validation, and quantification". In: *Annual reviews in control* 41 (2016), pp. 135–146.

[31] Yiding Ji, Yi-Chin Wu, and Stéphane Lafortune. "Enforcement of opacity by public and private insertion functions". In: *Automatica* 93 (2018), pp. 369–378.

[32] Varkey M John and Vaibhav Katewa. "On Connections between Opacity and Security in Linear Systems". In: *arXiv preprint arXiv:2206.06074* (2022).

[33] Varkey M John and Vaibhav Katewa. "Opacity and its Trade-offs with Security in Linear Systems". In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE. 2022, pp. 5443–5449.

[34] Varkey M John and Vaibhav Katewa. "Opacity Versus Security in Linear Dynamical Systems". In: *IEEE Transactions on Automatic Control* 70.1 (2025), pp. 323–338.

[35] Steven T Karris. *Introduction to Simulink with engineering applications*. Orchard Publications, 2006.

[36] Christoforos Keroglou and Christoforos N Hadjicostis. "Initial state opacity in stochastic DES". In: *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE. 2013, pp. 1–8.

[37] Sedighe Khaleghzade, Mostafa Zangiabadi, Aljoša Peperko, and Masoud Hajarian. "Interval multi-linear systems for tensors in the max-plus algebra and their application in solving the job shop problem". In: *Kybernetika* 58.5 (2022), pp. 708–732.

[38] Jan Komenda, Sébastien Lahaye, J-L Boimond, and Ton van den Boom. "Max-plus algebra in the history of discrete event systems". In: *Annual Reviews in Control* 45 (2018), pp. 240–249.

[39] Charalambos Konstantinou, Michail Maniatakos, Fareena Saqib, et al. "Cyber-physical systems: A security perspective". In: *2015 20th IEEE European Test Symposium (ETS)*. IEEE. 2015, pp. 1–8.

[40] SC Lauzon, AKL Ma, JK Mills, and B Benhabib. "Application of discrete-event-system theory to flexible manufacturing". In: *IEEE Control Systems Magazine* 16.1 (2002), pp. 41–48.

[41] Feng Lin. "Opacity of discrete event systems and its applications". In: *Automatica* 47.3 (2011), pp. 496–503.

[42] Mohammed Nasser Al-Mhiqani, Rabiah Ahmad, Warusia Yassin, et al. "Cyber-security incidents: a review cases in cyber-physical systems". In: *International Journal of Advanced Computer Science and Applications* 9.1 (2018).

[43] Henryk Minc. "Irreducible matrices". In: *Linear and Multilinear Algebra* 1.4 (1974), pp. 337–342.

[44] Sándor Molnár and Ferenc Szigeti. "Controllability and reachability of dynamic discrete-time linear systems". In: *2003 4th International Conference on Control and Automation Proceedings*. IEEE. 2003, pp. 350–354.

[45] Sándor Molnár, Ferenc Szigeti, and Márk Molnár. "A Rank Condition for Controllability and Reachability of Time-Varying Discrete-Time Linear Systems". In: *Mechanical Engineering Letters, Szent István University* (), p. 26.

[46] Vladimir Müller and Aljoša Peperko. "Generalized spectral radius and its max algebra version". In: *Linear Algebra and its Applications* 439.4 (2013), pp. 1006–1016.

[47] Geert Jan Olsder and Subiono. "On large scale max-plus algebra models in railway systems". In: *IFAC Proceedings Volumes* 31.18 (1998), pp. 649–653.

[48] Fabio Pasqualetti, Florian Dörfler, and Francesco Bullo. "Attack detection and identification in cyber-physical systems". In: *IEEE transactions on automatic control* 58.11 (2013), pp. 2715–2729.

[49] Miles Pollard. "A Case Study of Russian Cyber-Attacks on the Ukrainian Power Grid: Implications and Best Practices for the United States". In: *Pepperdine Policy Review* 16.1 (2024), p. 1.

[50] Jean-Michel Prou and Edouard Wagneur. "Controllability in the max-algebra". In: *Kybernetika* 35.1 (1999), pp. 13–24.

[51] Peter JG Ramadge and Walter Murray Wonham. "The control of discrete event systems". In: *Proceedings of the IEEE* 77.1 (1989), pp. 81–98.

[52] Bhaskar Ramasubramanian, Rance Cleaveland, and Steven I Marcus. "A framework for opacity in linear systems". In: *2016 American Control Conference (ACC)*. IEEE. 2016, pp. 6337–6344.

[53] Bhaskar Ramasubramanian, Rance Cleaveland, and Steven I Marcus. "Notions of centralized and decentralized opacity in linear systems". In: *IEEE Transactions on Automatic Control* 65.4 (2019), pp. 1442–1455.

[54] Bhaskar Ramasubramanian, Rance Cleaveland, and Steven I Marcus. "Opacity for switched linear systems: Notions and characterization". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 5310–5315.

[55] Gordon F Royle and Chris Godsil. *Algebraic graph theory*. Vol. 207. New York: Springer, 2001.

[56] Wilson J Rugh. *Linear system theory*. Prentice-Hall, Inc., 1996.

[57] Anooshiravan Saboori. *Verification and enforcement of state-based notions of opacity in discrete event systems*. University of Illinois at Urbana-Champaign, 2010.

[58] Anooshiravan Saboori and Christoforos N Hadjicostis. "Notions of security and opacity in discrete event systems". In: *2007 46th IEEE Conference on Decision and Control*. IEEE. 2007, pp. 5056–5061.

[59] Anooshiravan Saboori and Christoforos N Hadjicostis. "Verification of initial-state opacity in security applications of discrete event systems". In: *Information Sciences* 246 (2013), pp. 115–132.

[60] Manuel Silva. "On the history of discrete event systems". In: *Annual Reviews in Control* 45 (2018), pp. 213–222.

[61] Pavel Spacek, A El Moudni, S Zerhouni, and M Ferney. "Max-plus algebra for discrete event systems-some links to structural controllability and structural observability". In: *Proceedings of Tenth International Symposium on Intelligent Control*. IEEE. 1995, pp. 579–584.

[62] Marcelo Teixeira, Robi Malik, José ER Cury, and Max H de Queiroz. "Supervisory control of DES with extended finite-state machines and variable abstraction". In: *IEEE Transactions on Automatic Control* 60.1 (2014), pp. 118–129.

[63] Yin Tong, Zhiwu Li, Carla Seatzu, and Alessandro Giua. "Current-state opacity enforcement in discrete event systems under incomparable observations". In: *Discrete Event Dynamic Systems* 28 (2018), pp. 161–182.

[64] Harry L Trentelman, Anton A Stoorvogel, Malo Hautus, and L Dewell. "Control theory for linear systems". In: *Appl. Mech. Rev.* 55.5 (2002), B87–B87.

[65] Valessa V Viana, Jérémie Kreiss, and Marc Jungers. "On the computation of controlled invariant and output invisible subspaces for parameter-dependent systems". In: *IEEE Transactions on Automatic Control* 69.7 (2024), pp. 4695–4701.

[66] Yi-Chin Wu and Stéphane Lafortune. "Comparative analysis of related notions of opacity in centralized and coordinated architectures". In: *Discrete Event Dynamic Systems* 23.3 (2013), pp. 307–339.

[67] Yi-Chin Wu and Stéphane Lafortune. "Synthesis of insertion functions for enforcement of opacity security properties". In: *Automatica* 50.5 (2014), pp. 1336–1348.

[68] Yifan Xie, Xiang Yin, and Shaoyuan Li. "Opacity enforcing supervisory control using nondeterministic supervisors". In: *IEEE Transactions on Automatic Control* 67.12 (2021), pp. 6567–6582.

[69] Jingkai Yang and Weilin Deng. "Opacity Verification for a Class of Modular Discrete Event Systems". In: *IEEE Access* (2025).

[70] Xiang Yin, Majid Zamani, and Siyuan Liu. "On approximate opacity of cyber-physical systems". In: *IEEE Transactions on Automatic Control* 66.4 (2020), pp. 1630–1645.

[71] Kuize Zhang, Xiang Yin, and Majid Zamani. "Opacity of nondeterministic transition systems: A (bi) simulation relation approach". In: *IEEE Transactions on Automatic Control* 64.12 (2019), pp. 5116–5123.

[72] Lin Zhang, Xin Chen, Fanxin Kong, and Alvaro A Cardenas. "Real-time attack-recovery for cyber-physical systems using linear approximations". In: *2020 IEEE Real-Time Systems Symposium (RTSS)*. IEEE. 2020, pp. 205–217.

[73] Xiange Zhang. "Application of discrete event simulation in health care: a systematic review". In: *BMC health services research* 18.1 (2018), p. 687.

[74] Dmitry Zinoviev. "Discrete Event Simulation. It's Easy with SimPy!" In: (Feb. 2018).