POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

State Estimation with Switching Measurements: Sensor Scheduling and Existence of Observable Schedules

TESI DI LAUREA MAGISTRALE IN
AUTOMATION AND CONTROL ENGINEERING
INGEGNERIA DELL'AUTOMAZIONE

Author: **Amin Biglary Makvand**

Student ID: 10699953
Advisor: Prof. Alessandro Colombo
Co-advisor: Prof. Marcello Farina
Academic Year: 2021-22

Abstract

With the emergence of large sensor networks, there is a growing need of algorithms that can decide, for specific objectives, which sensors in a large network of sensors should be used at each time step. Regarding linear time-invariant systems that are under the measurement of large sensor networks, such sensors should grant observability and, at the same time, should minimize the state estimation error covariance. In the literature, most works are concerned with the minimization of the error covariance. This work aims to establish the foundation necessary to investigate the observability of such systems. When different sensors are selected at different time steps, the system measurement vector is time-variant. Thus, we are dealing with systems under time-variant measurement schemes. Therefore, we first attempt to extend the concept of observability to the case of linear systems with time-variant measurement. Then, we investigate conditions that allow the existence of observable sensor schedules first presented in [11]. We mainly focus on a theorem on the existence of observable schedules, and we try to restructure and simplify it by introducing new definitions and lemmas. As a result, an algorithm is introduced. This algorithm can find a subset of sensors in a set of available sensors that can be used to construct an observable sensor schedule. The algorithm is then numerically implemented. The results of the numerical implementation of the algorithm support the claim of the theorem and provide further insights on the structure of the observable sensor schedule.

Key-words: Sensor Scheduling, Observability, Sensor Selection, Time-Variant Measurement, Observable Schedule, N-horizon Observability, Sensor Networks.

Abstract in italiano

Con la diffusione di vaste reti di sensori, è nato un crescente bisogno di algoritmi che possano decidere, per specifici obiettivi, quali sensori di una grande rete debbano essere utilizzati ad ogni istante. Riguardo a sistemi lineari tempo-invarianti, la letteratura esistente si è concentrata su due problemi fondamentali: garantire l'osservabilità e minimizzare la variazione dell'errore di stima. Nella letteratura, molte ricerche hanno portato a sviluppare algoritmi in grado di minimizzare l'errore di stima. Questo studio punta a stabilire le basi necessarie per investigare l'osservabilità di suddetti sistemi. Quando diversi sensori sono selezionati in differenti istanti di tempo, il vettore di misura del sistema è tempo-variante. Perciò il problema considera sistemi sottoposti a schemi di misura tempo-varianti. Quindi prima di tutto si prova ad estendere il concetto di osservabilità al caso di sistemi lineari con misure tempo-varianti. Dopo di che, si studiano le condizioni che permettono l'esistenza di sequenze di sensori che garantiscano l'osservabilità. Concentrandosi principalmente su un teorema riguardante l'esistenza di sequenza osservabile, introdotto nel [11], si prova a ristrutturare il teorema in questione introducendo nuove definizioni. Come risultato si ottiene un algoritmo. Questo algoritmo può trovare un sottogruppo in un insieme di sensori disponibili che possono essere usati per costruire una sequenza di sensori osservabile. L'algoritmo è stato poi successivamente implementato. Il risultato dell'implementazione numerica supporta la tesi del teorema e mostra ulteriori approfondimenti riguardanti la struttura delle sequenze di sensori osservabili.

Parole chiave: Programmazione di sensori, osservabilità, selezione di sensori, misura tempo variante, sequenza osservabile, osservabilità su orizzonti, rete di sensori.

Contents

1 Introduction

In complex systems, the measurement scheme can involve a large network of sensors measuring the state of the system at each time step. Different constraints such as network bandwidth or power consumption create an incentive to use at each time step only a subset of the sensors in the sensor network, scheduling specific sensors to be used at specific time steps [1, 2, 3].

Sensor scheduling has applications in localization, energy management, wireless sensor networks (WSN), robotics, networked control systems (NCS), etc. [1, 2, 3, 4, 5]. For example, an active robotic mapping problem can be reformulated into a sensor scheduling problem and solved by the available algorithms in the literature [4]. In case there are energy constraints on WSNs, sensor scheduling solutions can be used to reduce the energy consumption of the sensor network with minimal compromise on estimation accuracy [1, 3]. For remote state estimation in NCSs, bandwidth limitations can be addressed using sensor scheduling-based approaches [5].

Most works in the literature [2, 3, 4, 6, 7] address the search of a sensor's scheduling from an optimization-based perspective.

Usually, in sensor scheduling problems, there are two main constraints: the number of sensors selected at each time step and the number of time steps or, in other words, the time horizon of the sensor schedule. Concerning the objectives of the problem, in literature, two main objectives can be found, minimization of the estimation error and observability. The focus in the literature is mainly on the minimization of estimation error under certain constraints using different methods and algorithms [1, 4]. On the other hand, observability has not been a focal point of the literature, and in many works, it is an assumption of the problem [2, 3, 6, 8, 9].

When we talk about the minimization of the estimation error, usually it concerns the error covariance matrix of a Kalman filter [4]. Different algorithms are used to find the schedule that provides the minimum estimation error. Tree search algorithms are one of the methods for finding an optimal sensor schedule. Theoretically, they can be used to find sensor schedules of any finite length with an arbitrary number of sensors, but computational limitations do not allow this in practice. Thus, certain pruning methods may be used to reduce the computational cost of these algorithms [4].

Observability as an objective of the sensor scheduling problem is not a focus of the literature and, as said before, in many works it is an assumption of the problem. One reason for this can be the computational cost of algorithms that consider observability as well. Overall, the sensor scheduling problem is computationally expensive and can be shown to be generally an NP-hard problem [10]. As the length of the schedule and the number of sensors grows, so does the computational cost.

Usually, observability is considered a binary condition that determines whether the initial condition can be recovered using a finite number of measurements. However, efforts have been made to expand this binary definition and employ a metric that can determine how observable a system is under different measurement schemes and what conditions determine whether an observable measurement scheme or, in other words, an observable sensor schedule exits [11].

This thesis aims to provide an understanding and foundation to explore the conditions that allow the existence of observable schedules in sensor scheduling problems. To achieve this goal, we focus on the work presented in [11] and we reformulate the algorithm and main results reported therein.

The original contributions of this thesis are the following:

- Formulating the observability problem for linear discrete-time systems with time-variant measurements.
- Reformulating a theorem on the existence of observable sensor schedules [11] by introducing new definitions, lemmas and proofs.
- Deriving an algorithm that can be used to construct observable sensor schedules.

This document is organized as follows: Chapter 2 first discusses the classical definition of observability, then extends this definition to the case of linear discrete-time systems with time-variant measurement. In Chapter 3, we focus on restructuring a theorem on the existence of observable schedules [11] by providing new definitions and lemmas. The first part of the chapter is dedicated to the formulation of the problem and providing important definitions. The second part of the chapter presents a new algorithm and the theorem on the existence of an observable schedule. The third part of the chapter deals with the proof. In Chapter 4, the results of the numerical implementation of the proposed algorithm are presented and discussed. Finally, In Chapter 5, the concluding remarks of this work are presented, where we attempt to highlight the most important results of the work and present possible topics for its future development.

2 Observability for linear discrete-time systems

Observability is a condition that determines if it is possible to infer the state of a system from the knowledge of system outputs over a finite period. This concept was introduced by [12, 13]. In this chapter, we present the concept of observability for linear time-invariant systems, which provides the necessary context to extend the definition and the theorem to the case of systems with time-variant measurements.

2.1. Linear time-invariant systems

This section is partially based on what is provided regarding the topic of observability in [14].

Consider the following state equation, where $C \in \mathbb{R}^{n_c \times n}$,

$$x(k+1) = Ax(k), \tag{2.1}$$

$$y(k) = Cx(k).$$

From now on, we denote by k_0 the initial time instant.

Definition 2.1 (*Observability*). Given the state equation (2.1), a state $x_0 \in \mathbb{R}^n$ is unobservable if, setting $x(k_0) = x_0$ as the initial state, $y(k) \equiv 0$ for all $k \geq k_0$. The state equation is observable if the zero vector $0 \in \mathbb{R}^n$ is the only unobservable state. ■

Before presenting Theorem 2.2, we need to introduce the following theorem:

Theorem 2.1 (*Cayley-Hamilton*). Let A be an $n \times n$ matrix and let $\{\lambda_0 \dots \lambda_{n-1}\}$ be the set of eigenvalues of A, then

$$(A - \lambda_0 I)(A - \lambda_1 I) \dots (A - \lambda_{n-1} I) = 0.$$

■

One consequence of the Cayley-Hamilton theorem is that any matrix $A^{\acute{n}}$, $\acute{n} \geq n$ can be expressed as the linear combination of $I, A, \ldots A^{n-1}$, i.e.,

$$A^{\acute{n}} = \sum_{j=0}^{n-1} m_j \, A^j,$$

where m_j is a scalar coefficient.

Theorem 2.2. The proposed linear system (2.1) is observable if and only if $\operatorname{rank}(\phi) = n$, where

$$\phi = \begin{bmatrix} C \\ CA \\ CA^2 \\ \cdots \\ CA^{n-1} \end{bmatrix}.$$

Proof of Theorem 2.2.

There are two implications involved in this theorem:

- If the state equation (2.1) is observable, then $\operatorname{rank}(\phi) = n$.
- If $\operatorname{rank}(\phi) = n$, then the state equation (2.1) is observable.

We first prove the contrapositive of the second implication.

- If $\operatorname{rank}(\phi) \neq n$ then the state equation (2.1) is not observable.

If $\operatorname{rank}(\phi) \neq n$ then $\operatorname{rank}(\phi) < n$ and the columns of ϕ are not linearly independent. Thus,

$$\phi a_0 = \begin{bmatrix} C \\ CA \\ CA^2 \\ \cdots \\ CA^{n-1} \end{bmatrix} a_0 = 0, a_0 \in \mathbb{R}^n.$$

Consequently

$$\begin{bmatrix} Ca_0 \\ CAa_0 \\ CA^2 a_0 \\ \cdots \\ CA^{n-1} a_0 \end{bmatrix} = 0.$$

Therefore,

$$Ca_0 = 0,$$

$$CAa_0 = 0,$$

$$CA^2 a_0 = 0,$$

$$\vdots$$

$$CA^{n-1} a_0 = 0.$$

Considering the time-domain solution of the state equation (2.1)

$$y(k) = CA^{k-k_0} x(k_0), k \geq k_0,$$

assuming $x(k_0) = a_0$, for any $k_0 \leq k < k_0 + n$ we have $0 \leq k - k_0 < n$. Thus, $y(k) \equiv 0$ for every $k_0 \leq k < k_0 + n$. Any power of A larger than n can be written as a linear combination of all A^i, $0 < i < n - 1$ (see Theorem 2.1). Thus, for any $k \geq k_0 + n$ we have $k - k_0 \geq n$, and

$$y(k) = CA^{k-k_0} a_0 = C \sum_{j=0}^{n-1} m_{j,k} A^j \ a_0,$$

which means

$$y(k) = \ m_{1,k} \, Ca_0 + \cdots + m_{n-1,k} \, CA^{n-1} a_0,$$

where $CA^j a_0 = 0$ for every $0 < j < \ n - 1$. Thus, $y(k) \equiv 0$ for every $k \geq k_0 + n$.

Now we prove the contrapositive of the first implication.

- If the state equation (2.1) is not observable, then $\text{rank}(\phi) \neq n$.

Under the assumption that the state equation is unobservable, according to Definition 2.1, there exists a non-zero $x(k_0)$ such that $y(k) \equiv 0$ for all $k \geq k_0$. Considering the time-domain solution of the state equation

$$y(k) = CA^{k-k_0} x(k_0), k \geq k_0,$$

for $k_0 \leq k < k_0 + n$,

$$y(k_0) = Cx(k_0) = 0,$$

$$y(k_0 + 1) = CAx(k_0) = 0,$$

$$\vdots$$

$$y(k_0 + n - 1) = CA^{n-1}x(k_0) = 0.$$

Therefore,

$$\begin{bmatrix} Cx(k_0) \\ CAx(k_0) \\ CA^2x(k_0) \\ \cdots \\ CA^{n-1}x(k_0) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \cdots \\ CA^{n-1} \end{bmatrix} x(k_0) = \phi x(k_0) = 0.$$

This means that the columns of ϕ are linearly dependent. Thus, rank$(\phi) < n$. ■

Definition 2.2 (*Observability Gramian*)**.** We denote

$$M(k_0, k) = \sum_{l=0}^{k-k_0} A^{T^{k-k_0-l}} C^T C A^{k-k_0-l}, k \geq k_0,$$

as the observability Gramian. ■

Corollary 2.1. rank$(\phi) = n$ if and only if Observability Gramian $M(k_0, k)$ is nonsingular for any $k \geq k_0 + n - 1$.

Proof of Corollary 2.1.

The corollary involves two implications:

- If rank$(\phi) = n$, then $M(k_0, k)$ for any $k \geq k_0 + n - 1$ is nonsingular.
- If $M(k_0, k)$ for any $k \geq k_0 + n - 1$ is nonsingular, then rank$(\phi) = n$.

We proceed with proving the contrapositive of the second implication.

- If $M(k_0, k)$ for some $k \geq k_0 + n - 1$ is not nonsingular then rank$(\phi) \neq n$.

A singular $M(k_0, k)$ for some $k \geq k_0 + n - 1$ means that there exists a non-zero x_0 such that

$$M(k_0, k)x_0 = \left[\sum_{l=0}^{k-k_0} A^{T^{k-k_0-l}} C^T C A^{k-k_0-l} \right] x_0 = 0.$$

Therefore

$$x_0{}^T\, M(k_0,k)x_0 = x_0{}^T \left[\sum_{l=0}^{k-k_0} A^{T^{k-k_0-l}} C^T C A^{k-k_0-l}\right] x_0 =$$

$$\textstyle\sum_{l=0}^{k-k_0} \|CA^{k-k_0-l}x_0\|^2 = 0.$$

The sum of squared Euclidean norms is equal to zero; thus, each element of the series must be zero, and we have

$$Cx_0 = 0,$$

$$CAx_0 = 0,$$

$$\vdots$$

$$CA^{k-k_0}x_0 = 0.$$

Because $k \geq k_0 + n - 1$ we have

$$\begin{bmatrix} Cx_0 \\ CAx_0 \\ CA^2x_0 \\ \cdots \\ CA^{n-1}x_0 \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \cdots \\ CA^{n-1} \end{bmatrix} x_0 = \phi x_0 = 0.$$

Thus, rank$(\phi) < n$. Now we prove the contrapositive of the first implication.

- If rank$(\phi) \neq n$ then $M(k_0,k)$ for some $k \geq k_0 + n - 1$ is not nonsingular.

A rank$(\phi) \neq n$ then rank$(\phi) < n$. This means that there exists a non-zero x_0 such that

$$\phi x_0 = \begin{bmatrix} C \\ CA \\ CA^2 \\ \cdots \\ CA^{n-1} \end{bmatrix} x_0 = 0.$$

Therefore,

$$\begin{bmatrix} Cx_0 \\ CAx_0 \\ CA^2x_0 \\ \cdots \\ CA^{n-1}x_0 \end{bmatrix} = 0,$$

consequently

$$Cx_0 = 0,$$

$$CAx_0 = 0,$$

$$\vdots$$

$$CA^{n-1}x_0 = 0.$$

As a consequence of Theorem 2.1, $CA^ix_0 = 0$ for any $i \geq n$. Thus, we have

$$M(k_0, k)x_0 = \left[\sum_{l=0}^{k-k_0} A^{T^{k-k_0-l}} C^T (CA^{k-k_0-l})\right] x_0 = 0.$$

CA^ix_0 appears in all elements of the $M(k_0, k)x_0$ series; thus, for any $k \geq k_0$, and consequently any $k \geq k_0 + n - 1$, $M(k_0, k)x_0 = 0$. This means that $M(k_0, k)$ is singular. ■

The Gramian matrix $M(k_0, k)$, $k \geq k_0 + n - 1$ can be used to determine $x(k_0)$. Using the measurements, we can see the relationship $M(k_0, k)$ has with $x(k_0)$ for any $k > k_0$ below.

$$Y_{k,k_0} = \sum_{l=0}^{k-k_0} A^{T^{k-k_0-l}} C^T y(k-l) = \sum_{l=0}^{k-k_0} A^{T^{k-k_0-l}} C^T CA^{k-k_0-l} x(k_0)$$

$$= M(k_0, k)x(k_0).$$

Corollary 2.1 states that for an observable system, $M(k_0, k)$ is nonsingular for any $k \geq k_0 + n - 1$; therefore, it is guaranteed that $x(k_0)$ can be determined uniquely if there is sufficient measurement data Y_{k,k_0}. Thus, for any $k \geq k_0 + n - 1$,

$$M(k_0, k)^{-1}Y_{k,k_0} = M(k_0, k)^{-1}M(k_0, k)x(k_0) = x(k_0).$$

2.2. Linear systems with time-variant measurements

In this section, the definition of observability is extended to the case of a linear discrete-time system with variant measurements.

Consider state equation (2.1) with time-variant measurements $C_k \in \mathbb{R}^{n_c \times n}$. Thus, we have,

$$x(k+1) = Ax(k), \tag{2.2}$$

$$y(k) = C_k x(k).$$

Definition 2.3. (*Measurement Time Horizon*). The Measurement time horizon T_N is defined as a sequence of time steps $\{k_0, k_0 + 1, ..., k_0 + N - 1\}$ in which the system is under measurement. ■

Definition 2.4. (*N-horizon Observability*). Given the state equation (2.2), a state $x_0 \in \mathbb{R}^n$ is unobservable over a measurement time horizon $T_N = \{k_0, k_0 + 1, \ldots, k_0 + N - 1\}$ if setting $x(k_0) = x_0$ as the initial state, $y(k) \equiv 0$ for all $k \in T_N$. The state equation is N-horizon observable if the zero vector $0 \in \mathbb{R}^n$ is the only unobservable state. ■

Theorem 2.3. The proposed linear system (2.2) is N-horizon observable over the measurement time horizon $T_N = \{k_0, k_0 + 1, \ldots, k_0 + N - 1\}$ if and only if $\text{rank}(\phi_N) = n$, where

$$\phi_N = \begin{bmatrix} C_{k_0} \\ C_{k_0+1}A \\ C_{k_0+2}A^2 \\ \cdots \\ C_{k_0+N-1}A^{N-1} \end{bmatrix}.$$

Proof of Theorem 2.3.

There are two implications involved in this theorem:

- If the state equation (2.2) is N-horizon observable over a measurement time horizon T_N, then $\text{rank}(\phi_N) = n$.
- If $\text{rank}(\phi_N) = n$, then the state equation (2.2) is N-horizon observable over a measurement time horizon T_N.

We first prove the contrapositive of the second implication.

- If $\text{rank}(\phi_N) \neq n$, then the state equation (2.2) is not observable.

If $\operatorname{rank}(\phi_N) \neq n$ then $\operatorname{rank}(\phi_N) < n$. Thus, the columns of ϕ_N are linearly dependent, and we have

$$\phi_N a_0 = \begin{bmatrix} C_{k_0} \\ C_{k_0+1}A \\ C_{k_0+2}A^2 \\ \cdots \\ C_{k_0+N-1}A^{N-1} \end{bmatrix} a_0 = 0, a_0 \in \mathbb{R}^n.$$

Consequently

$$\begin{bmatrix} C_{k_0}a_0 \\ C_{k_0+1}Aa_0 \\ C_{k_0+2}A^2a_0 \\ \cdots \\ C_{k_0+N-1}A^{N-1}a_0 \end{bmatrix} = 0.$$

Therefore,

$$C_{k_0}a_0 = 0,$$

$$C_{k_0+1}Aa_0 = 0,$$

$$C_{k_0+2}A^2a_0 = 0,$$

$$\vdots$$

$$C_{k_0+N-1}A^{N-1}a_0 = 0.$$

Considering the time-domain solution of the state equation (2.2)

$$y(k) = C_kA^{k-k_0}x(k_0), k \geq k_0,$$

assuming $x(k_0) = a_0$, for any $k_0 \leq k \leq k_0 + N - 1$ we have $0 \leq k - k_0 \leq N - 1$; thus, $y(k) \equiv 0$ for every $k_0 \leq k \leq k_0 + N - 1$. Now we prove the contrapositive of the first implication.

- If the state equation (2.2) is not observable over a measurement time horizon T_N, then $\operatorname{rank}(\phi_N) \neq n$.

Under the assumption that the state equation is not observable, according to Definition 2.4, there exists a non-zero $x(k_0)$ such that $y(k) \equiv 0$ for all $k_0 \leq k \leq k_0 + N - 1$. Considering the time-domain solution of the state equation

$$y(k) = C_k A^{k-k_0} x(k_0), k \geq k_0,$$

for $k_0 \leq k \leq k_0 + N - 1$, we have

$$y(k_0) = C_{k_0} x(k_0) = 0,$$

$$y(k_0 + 1) = C_{k_0+1} A x(k_0) = 0,$$

$$\vdots$$

$$y(k_0 + N - 1) = C_{k_0+N-1} A^{N-1} x(k_0) = 0.$$

Therefore,

$$\begin{bmatrix} C_{k_0} x(k_0) \\ C_{k_0+1} A x(k_0) \\ C_{k_0+2} A^2 x(k_0) \\ \cdots \\ C_{k_0+N-1} A^{N-1} x(k_0) \end{bmatrix} = \begin{bmatrix} C_{k_0} \\ C_{k_0+1} A \\ C_{k_0+2} A^2 \\ \cdots \\ C_{k_0+N-1} A^{N-1} \end{bmatrix} x(k_0) = \phi x(k_0) = 0.$$

This means column vectors of ϕ are not linearly independent. Thus, rank$(\phi) < n$. ■

Definition 2.5 (*Time Variant Observability Gramian*). Consider the matrix below and call it the time-variant observability Gramian.

$$M_T(k_0, k) = \sum_{l=0}^{k-k_0} A^{T^{k-k_0-l}} {C_{k-l}}^T C_{k-l} A^{k-k_0-l}, k \geq k_0.$$

■

Corollary 2.2. rank$(\phi_N) = n$ if and only if Observability Gramian $M_T(k_0, k_0 + N - 1)$ is nonsingular.

Proof of Corollary 2.2.

The corollary involves two implications:

- If rank$(\phi_N) = n$, then $M_T(k_0, k_0 + N - 1)$ is nonsingular.
- If $M_T(k_0, k_0 + N - 1)$ is nonsingular, then rank$(\phi_N) = n$.

We proceed with proving the contrapositive of the second implication.

- If $M_T(k_0, k_0 + N - 1)$ is not nonsingular, then rank$(\phi_N) \neq n$.

Given a singular $M_T(k_0, k_0 + N - 1)$, there exists a non-zero $x_0 \in \mathbb{R}^n$ such that

$$M_T(k_0, k_0 + N - 1)x_0 =$$

$$\left[\sum_{l=0}^{N-1} A^{T^{N-l-1}} C_{k_0+N-l-1}{}^T C_{k_0+N-l-1} A^{N-l-1}\right] x_0 = 0.$$

Therefore,

$$x_0{}^T\, M_T(k_0, k_0 + N - 1)x_0 =$$

$$x_0{}^T \left[\sum_{l=0}^{N-1} A^{T^{N-l-1}} C_{k_0+N-l-1}{}^T C_{k_0+N-l-1} A^{N-l-1}\right] x_0 =$$

$$\textstyle\sum_{l=0}^{N} \left\| C_{k_0+N-l-1} A^{N-l-1} x_0 \right\|^2 = 0.$$

The sum of squared Euclidean norms is equal to zero; thus, each element of the series must be zero, and we have

$$C_{k_0} x_0 = 0,$$

$$C_{k_0+1} A x_0 = 0,$$

$$\vdots$$

$$C_{k_0+N-1} A^{N-1} x_0 = 0.$$

Thus, we have

$$\begin{bmatrix} C_{k_0} x_0 \\ C_{k_0+1} A x_0 \\ C_{k_0+2} A^2 x_0 \\ \cdots \\ C_{k_0+N-1} A^{N-1} x_0 \end{bmatrix} = \begin{bmatrix} C_{k_0} \\ C_{k_0+1} \\ C_{k_0+2} A^2 \\ \cdots \\ C_{k_0+N-1} A^{N-1} \end{bmatrix} x_0 = \phi_N x_0 = 0.$$

Which means $\text{rank}(\phi_N) < n$. Now we prove the contrapositive of the first implication.

- If $\text{rank}(\phi_N) \neq n$, then $M_T(k_0, k_0 + N - 1)$ is not nonsingular.

If $\text{rank}(\phi_N) \neq n$, then $\text{rank}(\phi_N) < n$. This means that there exists a non-zero x_0 such that

$$\phi_N x_0 = \begin{bmatrix} C_{k_0} \\ C_{k_0+1} \\ C_{k_0+2}A^2 \\ \cdots \\ C_{k_0+N-1}A^{N-1} \end{bmatrix} x_0 = 0.$$

Therefore,

$$\begin{bmatrix} C_{k_0}x_0 \\ C_{k_0+1}Ax_0 \\ C_{k_0+2}A^2x_0 \\ \cdots \\ C_{k_0+N-1}A^{N-1}x_0 \end{bmatrix} = 0,$$

consequently

$$C_{k_0}x_0 = 0$$

$$C_{k_0+1}Ax_0 = 0$$

$$\vdots$$

$$C_{k_0+N-1}A^{N-1}x_0 = 0.$$

Thus, we have

$$M_T(k_0, k_0+N-1)x_0 =$$

$$\left[\sum_{l=0}^{N-1} A^{T^{N-l-1}} C_{k_0+N-l-1}{}^{T} C_{k_0+N-l-1}A^{N-l-1}\right] x_0 =$$

$$\sum_{l=0}^{N-1} A^{T^{N-l-1}} C_{k_0+N-l-1}{}^{T} (C_{k_0+N-l-1}A^{N-l-1})x_0 = 0.$$

Therefore, $M(k_0, k_0+N-1)$ is singular. ■

The Gramian matrix $M_T(k_0, k_0+N-1)$ can be used to determine $x(k_0)$. Using the measurements, we can see the relationship $M_T(k_0, k)$ has with $x(k_0)$ for any $k > k_0$ below.

$$Y_{k_0+N-1,k_0} = \sum_{l=0}^{N-1} A^{T^{N-l-1}} C_{k_0+N-l-1}{}^{T} y(k_0+N-l-1) =$$

$$\sum_{l=0}^{N-1} A^{T^{N-l-1}} C_{k_0+N-l-1}{}^{T} C_{k_0+N-l-1} A^{N-l-1} x(k_0) =$$

$$M_T(k_0, k_0 + N - 1)\, x(k_0).$$

Corollary 2.2 states that for an observable system, $M_T(k_0, k_0 + N - 1)$ is nonsingular. Therefore, it is guaranteed that $x(k_0)$ can be determined uniquely using all measurement data of T_N. Thus, we have

$$M_T(k_0, k_0 + N - 1)^{-1} Y_{k_0+N-1,k_0} =$$

$$M_T(k_0, k_0 + N - 1)^{-1} M_T(k_0, k_0 + N - 1)\, x(k_0)$$

$$= x(k_0).$$

3 Existence of an observable schedule

In this chapter, we first define the concept of sensor schedule and its observability properties. Then we explore the conditions that allow for the existence of an observable sensor schedule for a discrete-time linear system with time-variant measurements. An algorithm drawn from [11] is proposed. If specific conditions discussed below are met, this algorithm can be used to construct an observable sensor schedule. A theorem introduced in [11] on the existence of observable schedules is restructured here and used as the basis of the proposed algorithm.

The structure of this chapter is as follows. First, the problem is formulated borrowing terms and notations from [4]. Then new definitions are introduced. Following the definitions, an algorithm is proposed, and a theorem on the existence of observable sensor schedules is presented. To prove the main theorem, two lemmas are introduced, and proofs are provided. At the end of the chapter, the results of the numerical implementation of the algorithm are presented and discussed.

3.1. Problem formulation

Consider the following linear system:

$$x(k+1) = Ax(k), \tag{3.1}$$

where $x(k) \in \mathbb{R}^n$ is the state of the system. Consider the ordered set $S = \{S[0], S[1], \ldots, S[p-1]\}$, where $S[i] \in \mathbb{R}^n$ is a sensor vector that can be used to measure the system state. Each element of the set S represents one sensor.

Definition 3.1 (*Sensor Schedule*). A sensor schedule Σ over an ordered sensor set S is defined as a set of elements $\sigma_k \in \{0, \ldots, p-1\}$ where $\sigma_k = i$ denotes the usage of sensor $S[\sigma_k] \in S$ at time k. ■

Definition 3.2 (*N-horizon Sensor Schedule*)**.** Let T_N be the measurement time horizon (see Definition 2.3). Denote by $\Sigma_N = \{\sigma_{k_0}, \sigma_{k_0+1}, \ldots, \sigma_{k_0+N-1}\}$, $\sigma_k \in \{0, \ldots, p-1\}$ a N-horizon sensor schedule. ■

To measure the system state, only one sensor is allowed to operate at each time step. Under a given schedule Σ_N, the scheduled measurement at each time step is

$$y(k) = S[\sigma_k]^T x(k), \quad \forall\, k \in \{k_0, k_0 + 1, \dots, k_0 + N - 1\}.$$

3.2. Observable sensor schedule

Definition 3.3 (*Observable N-horizon Sensor Schedule*). A N-horizon sensor schedule with an observability matrix ϕ_N of rank n is called an observable N-horizon sensor schedule (see Theorem 2.3). ■

The following definitions are required to define conditions for N-horizon observability and the definition of observable N-horizon sensor schedules.

Definition 3.4 (*Cover Set*). Let $\alpha = [c_0 \dots c_{n-1}]^T, c_i \in \mathbb{R}$ be the coordinate vector of vector $v \in \mathbb{R}^n$, with respect to a basis $G = \{g_0, \dots, g_{n-1}\}$ spanning vector space $\mathbb{R}^n$. The cover set of vector v with respect to the basis G, denoted by β, is the set of all basis elements $g_i \in G$ with an associated $c_i \neq 0$. ■

Definition 3.5 (*Basis Coverage*). A vector set $V = \{v_0, v_1, \dots, v_{n_v-1}\}, v_i \in \mathbb{R}^{n_v}$ covers the basis G when

$$g_i \in \cup_{l=0}^{n_v-1} \beta_l, \forall\, g_i \in G,$$

where each β_l is the cover set of its corresponding vector v_l, with respect to the basis G. ■

Definition 3.6 (*Non-common Element*). A set β_a has a non-common element with respect to a set β_b, if there exists a $g_i \in G$ such that $g_i \in \beta_a$ and $g_i \notin \beta_b$. g_i is called a non-common element of β_a with respect to β_b. ■

3.3. Algorithm and theorem

What follows next is an algorithm that can be used to find a specific subset of S, denoted by S_o. Later in Theorem 3.1, it is indicated that elements of S_o ,under certain conditions, can be used to construct an observable sensor schedule. To proceed, consider the following assumption.

Assumption 3.1. Matrix A is nonsingular with distinct eigenvalues. ■

Consistently with Assumption 3.1, matrix A has n linearly independent eigenvectors. These eigenvectors are used as the elements of basis $G = \{g_0, \dots, g_{n-1}\}$. More specifically vectors $g_i \in \mathbb{R}^n$ are the left eigenvectors of A.

As mentioned, the proposed algorithm accepts ordered set S and returns the subset S_o. Here we repeat a short explanation of its main rational. The algorithm initializes a set S_r with S (line 1), then at each iteration, it selects a sensor vector from the set S_r, removes it from S_r (line 28) and adds it to S_o (line 24). To do so, the algorithm looks at the cover sets of every sensor vector that has in S_r (15 to 22), and selects the sensor vector (denoted B_o^1) with the largest set (denoted B_o^0) of non-common elements (see Definition 3.6) with respect to the union of previously chosen sets of non-common elements (lines 15 to 23). The algorithm then adds the largest set of the non-common elements B_o^0 to the set $B_{\mathcal{M}}$ (line 26), removes it from γ_r (line 27) and adds its corresponding sensor vector B_o^1 to S_o. The algorithm continues this process until one of two conditions is met: either there are no sensors left to check, corresponding to $S_r = \emptyset$, or there are no non-common elements left, corresponding to $\gamma_r = \emptyset$. Note that the elements in the cover sets are the vectors of basis G.

Before proceeding with the algorithm, consider Table 3.1, explaining the role of each parameter and function in the algorithm.

Table 3.1:Algorithm parameters and functions description.

Element	Role
Cover(·,·)	Function that accepts a vector and a basis and returns the cover set of the input vector with respect to the input basis as an ordered set.
n(·)	Function that accepts an ordered set and returns the cardinality of the set.
Max(·)	Function that accepts an ordered set of pairs (E_0, E_1), and returns the pair with the largest $n(E_0)$. If there are multiple pairs with the largest $n(E_0)$, it returns the first one it finds.
Append(·,·)	Function that accepts an ordered set with N elements as the first argument and a set element as the second and adds the element to the set as the $N+1$-th element.
/	Operator that removes a subset from a set
S_r	Ordered set initialized with all the sensor vectors of set S. At each iteration, one element is removed from this set. If this set is empty, the algorithm terminates.

γ_r	Ordered set initialized with all the basis elements of G. At each iteration, one element is removed from this set. If this set is empty, the algorithm terminates.
S_o	Ordered set initialized with $\emptyset$. At every iteration, a sensor vector is added, chosen from S_r. This ordered set is an output when the algorithm terminates.
β_{sub}	Ordered set with an initial value of $\emptyset$. At every iteration of the second while loop (line 15), it is calculated for one of the remaining elements of S_r denoted by $S_r[l]$. It contains the non-common elements of Cover$(S_r[l], G)$ with respect to sets of non-common elements selected in previous iterations.
B_p	Ordered set with an initial value of $\emptyset$. At every iteration of the second while loop (line 15), a pair $(\beta_{sub}, S_r[l])$ is appended to it. At end of the iteration of the first while loop (line 13), it is reset to $\emptyset$. Thus, after the end of the second while loop, B_p contains the set of non-common elements of every remaining sensor vector in S_r as well as the corresponding sensor vectors. Later it is used in line 23 to find the largest set of non-common elements.
B_o^0	Ordered set. At each iteration contains the largest set of non-common elements with respect to the previously selected ones.
B_o^1	Vector containing the corresponding sensor vector of the largest set of non-common elements at each iteration.
$B_{\mathcal{M}}$	Ordered set with an initial value of $\emptyset$. At each iteration, the set B_o^0 is added to it. All elements of $B_{\mathcal{M}}$ are disjoint with each other and with γ_r at each iteration.
N_o	Ordered set with an initial value of $\emptyset$. At every iteration, the cardinality of the largest set of non-common is added to it.
z	Ordered set with integer elements where each element $z[m] = \sum_{i=0}^{m} n(B_{\mathcal{M}}[i])$. Later in Theorem 3.1, it is shown that a sensor $S_o[i]$ in the observable sensor schedule Σ_n is used from time step $z[i-1]$ to time step $z[i]-1$. Consider $z[-1] = 0$.

Note that all indexes, subscripts, and superscripts start from zero in this work.

Algorithm 3.1 (*Sensor Set Observability Filter*).

ordered set $S_r \leftarrow S$
ordered set $\gamma_r \leftarrow G$
ordered set $S_o \leftarrow \emptyset$
ordered set $N_o \leftarrow \emptyset$
ordered set $z \leftarrow \emptyset$
ordered set $B_{\mathcal{M}} \leftarrow \emptyset$
ordered set $\beta_{sub} \leftarrow \emptyset$
ordered set $B_p \leftarrow \emptyset$
ordered set $B_o^0 \leftarrow \emptyset$
ordered set $B_o^1 \leftarrow \emptyset$
integer $l \leftarrow 0$
integer $\zeta \leftarrow 0$
while $\gamma_r \neq \emptyset$ **or** $S_r \neq \emptyset$
    $l \leftarrow 0$
    while $l < \mathbf{n}(S_r)$
        if $B_{\mathcal{M}} = \emptyset$
            $\beta_{sub} \leftarrow \mathbf{Cover}(S_r[l], G)$
            $B_p \leftarrow \mathbf{Append}(B_p, (\beta_{sub}, S_r[l]))$
        else
            $\beta_{sub} \leftarrow \mathbf{Cover}(S_r[l], G) / \left(\left(\cup_{k=0}^{\mathbf{n}(B_{\mathcal{M}})-1} B_{\mathcal{M}}[k]\right) \cap \mathbf{Cover}(S_r[l], G)\right)$
            $B_p \leftarrow \mathbf{Append}(B_p, (\beta_{sub}, S_r[l]))$
        $l \leftarrow l + 1$
    $(B_o^0, B_o^1) \leftarrow \mathbf{Max}(B_p)$
    $S_o \leftarrow \mathbf{Append}(S_o, B_o^1)$
    $N_o \leftarrow \mathbf{Append}(N_o, \mathbf{n}(B_o^0))$
    $B_{\mathcal{M}} \leftarrow \mathbf{Append}(B_{\mathcal{M}}, B_o^0)$
    $\gamma_r \leftarrow \gamma_r / B_o^0$
    $S_r \leftarrow S_r / B_o^1$
    $\zeta \leftarrow \zeta + \mathbf{n}(B_o^0)$
    $z \leftarrow \mathbf{Append}(z, \zeta)$
    $B_p \leftarrow \emptyset$
return $S_o, N_o, B_{\mathcal{M}}, \gamma_r, z$

Assumption 3.2. Integer θ is equal to 1 + the number of elements in returned sets S_o, N_o, $B_{\mathcal{M}}$ and z when the algorithm terminates under the condition $\gamma_r = \emptyset$. ■

The following theorem can be proved, providing a fundamental result for defining observable sensor schedules.

Theorem 3.1. If S covers basis G (see Definition 3.5), then there exists an observable n-horizon sensor schedule $\Sigma_n = \{\sigma_{k_0}, \sigma_{k_0+1}, \dots, \sigma_{k_0+n-1}\}$ over sensor set S_o (see Definition 3.2), with the following structure:

$$\sigma_{k_0} = 0,$$

$$\vdots$$

$$\sigma_{k_0+z[0]-1} = 0,$$

$$\sigma_{k_0+z[0]} = 1,$$

$$\vdots$$

$$\sigma_{k_0+z[1]-1} = 1,$$

$$\vdots$$

$$\sigma_{k_0+z[\theta-1]} = \theta,$$

$$\vdots$$

$$\sigma_{k_0+z[\theta]-1} = \theta,$$

such that $z[\theta] = n$. ■

In this schedule, every sensor vector in the returned ordered set S_o is used. Each sensor is used $z[i] - z[i-1]$ number of times.

3.4. Proofs and discussion

To approach the proof of Theorem 3.1, we should take specific steps, stated as follows. First, we define and prove Lemma 3.1 stated below. Then, we construct the so-called basis observability matrix. As the next step, we define and prove Lemma 3.2 and finally, we will use the result of Lemma 3.2 to prove Theorem 3.1 schematically:

- Theorem 3.1: $H1 \rightarrow H4$
- Lemma 3.1: $H2 \leftrightarrow H1$
- Lemma 3.2: $H2 \rightarrow H3$

Following the rule of inference, to prove Theorem 3.1 we will have

Premise 1: $H1$
Premise 2: $H1 \rightarrow H2$ (Lemma 3.1)
Premise 3: $H2 \rightarrow H3$ (Lemma 3.2)
Premise 4: $H3 \rightarrow H4$ (Theorem 3.1)

Conclusion: $H4$

We can now proceed with the step-by-step process of proving Theorem3.1.

3.4.1. Statement and proof of Lemma 3.1

Lemma 3.1. Algorithm 3.1 terminates under the condition $\gamma_r = \emptyset$ if and only if S covers the basis G.

Proof of Lemma 3.1. There are two implications involved:

- If S covers the basis G, Algorithm 3.1 terminates under the condition $\gamma_r = \emptyset$.
- If Algorithm 3.1 terminates under the condition $\gamma_r = \emptyset$, then S cover the basis G.

Now we begin with the proof of the contrapositive of the first implication.

- If S does not cover the basis G then Algorithm 3.1 does not terminate under the condition $\gamma_r = \emptyset$.

Assume Algorithm 3.1 terminates after $\mu + 1$ elements are added to S_o and removed from S_r . It is not clear yet whether the algorithm has terminated under the condition $\gamma_r = \emptyset$ or $S_r = \emptyset$ or both, but we know that S_o is a subset of S. Considering that S does not cover the basis G, we have the following:

$$\exists\, g_i \left(g_i \in G \wedge g_i \notin \cup_{l=0}^{\mu} \text{Cover}(S_o[l], G) \right), \qquad (I)$$

$$\forall\, l \left(0 \leq l \leq \mu \Rightarrow B_{\mathcal{M}}[l] \subseteq \text{Cover}(S_o[l], G)\right), \qquad (II)$$

$$\gamma_r \cup \left(\cup_{l=0}^{\mu} B_{\mathcal{M}}[l]\right) = G, \qquad (III)$$

$$(II) \Rightarrow \cup_{l=0}^{\mu} B_{\mathcal{M}}[l] \subseteq \cup_{l=0}^{\mu} \text{Cover}(S_o[l], G),$$

$$(I) \wedge \left(\cup_{l=0}^{\mu} B_{\mathcal{M}}[l] \subseteq \cup_{l=0}^{\mu} \text{Cover}(S_o[l], G)\right) \Rightarrow \exists\, g_i \left(g_i \in G \wedge g_i \notin \cup_{l=0}^{\mu} B_{\mathcal{M}}[l]\right),$$

$$(III) \wedge \exists\, g_i \left(g_i \in G \wedge g_i \notin \cup_{l=0}^{\mu} B_{\mathcal{M}}[l]\right) \Rightarrow \gamma_r \neq \emptyset.$$

Now we prove the contrapositive of the second implication.

- If Algorithm 3.1 does not terminate under the condition $\gamma_r = \emptyset$, then S does not cover the basis G.

If $\gamma_r \neq \emptyset$ when the algorithm terminates, then $S_r = \emptyset$. This means that S_o now contains all the elements of S. To prove this contrapositive, we need to show that

$$\cup_{l=0}^{p-1} B_{\mathcal{M}}[l] = \cup_{l=0}^{p-1} \text{Cover}(S_o[l], G).$$

From Algorithm 3.1, for $l = 0$ we have

$$B_{\mathcal{M}}[0] = \text{Cover}(S_o[0], G),$$

and using set algebra, for $1 \leq l \leq p-1$,

$$B_{\mathcal{M}}[l] = \text{Cover}(S_o[l], G) / \left(\left(\cup_{k=0}^{l-1} B_{\mathcal{M}}[k]\right) \cap \text{Cover}(S_o[l], G)\right) =$$

$$\text{Cover}(S_o[l], G) \cap \left(\left(\cup_{k=0}^{l-1} B_{\mathcal{M}}[k]\right) \cap \text{Cover}(S_o[l], G)\right)' =$$

$$\text{Cover}(S_o[l], G) \cap \left(\left(\cup_{k=0}^{l-1} B_{\mathcal{M}}[k]\right)' \cup \text{Cover}(S_o[l], G)'\right) =$$

$$\text{Cover}(S_o[l], G) \cap \left(\cup_{k=0}^{l-1} B_{\mathcal{M}}[k]\right)' =$$

$$\text{Cover}(S_o[l], G) \cap \left(\cap_{k=0}^{l-1} B_{\mathcal{M}}[k]'\right).$$

Therefore, for $1 \leq l \leq p-1$,

$$B_{\mathcal{M}}[l] = \text{Cover}(S_o[l], G) \cap \left(\cap_{k=0}^{l-1} B_{\mathcal{M}}[k]'\right).$$

Expanding for every $0 \leq l \leq p-1$,

$$B_{\mathcal{M}}[0] = \text{Cover}(S_o[0], G)$$

$$B_{\mathcal{M}}[1] = \mathrm{Cover}(S_o[1], G) \cap \mathrm{Cover}(S_o[0], G)'$$

$$B_{\mathcal{M}}[2] = \mathrm{Cover}(S_o[2], G) \cap \mathrm{Cover}(S_o[0], G)' \cap \mathrm{Cover}(S_o[1], G)'$$

$$\vdots$$

$$B_{\mathcal{M}}[p-1] = \mathrm{Cover}(S_o[p-1], G) \cap \mathrm{Cover}(S_o[0], G)' \cap \dots$$

$$\cap\, \mathrm{Cover}(S_o[p-2], G)'.$$

Taking the union

$$\cup_{l=0}^{p-1} B_{\mathcal{M}}[l] = B_{\mathcal{M}}[0] \cup B_{\mathcal{M}}[1] \cup \dots \cup B_{\mathcal{M}}[p-1] =$$

$$\mathrm{Cover}(S_o[0], G) \cup \dots$$

$$\dots \cup (\mathrm{Cover}(S_o[p-1], G) \cap \mathrm{Cover}(S_o[0], G)' \cap \dots \cap \mathrm{Cover}(S_o[p-2], G)') =$$

$$\big((\mathrm{Cover}(S_o[0], G) \cup \mathrm{Cover}(S_o[1], G)) \cap U\big) \cup \dots$$

$$\dots \cup (\mathrm{Cover}(S_o[p-1], G) \cap \mathrm{Cover}(S_o[0], G)' \cap \dots \cap \mathrm{Cover}(S_o[p-2], G)') =$$

$$\dots = \big(\mathrm{Cover}(S_o[0], G) \cup \mathrm{Cover}(S_o[1], G) \dots \cup \mathrm{Cover}(S_o[p-1], G)\big) \cap U =$$

$$\mathrm{Cover}(S_o[0], G) \cup \mathrm{Cover}(S_o[1], G) \dots \cup \mathrm{Cover}(S_o[p-1], G) = \cup_{l=0}^{p-1} \mathrm{Cover}(S_o[l], G),$$

where U is the universal set. Thus,

$$\cup_{l=0}^{p-1} B_{\mathcal{M}}[l] = \cup_{l=0}^{p-1} \mathrm{Cover}(S_o[l], G). \qquad (IV)$$

From Algorithm 3.1 and the contrapositive of the second implication, we have

$$\gamma_r \cup \big(\cup_{l=0}^{p-1} B_{\mathcal{M}}[l]\big) = G, \qquad (V)$$

$$\gamma_r \neq \emptyset. \qquad (VI)$$

Thus,

$$(VI) \wedge (V) \Rightarrow \exists\, g_i \big(g_i \in G \wedge g_i \notin \cup_{l=0}^{p-1} B_{\mathcal{M}}[l]\big),$$

$$(IV) \wedge \exists\, g_i \left(g_i \in G \wedge g_i \notin \cup_{l=0}^{p-1} B_{\mathcal{M}}[l]\right) \Rightarrow$$

$$\exists\, g_i \left(g_i \in G \wedge\ g_i \notin \cup_{l=0}^{p-1} \mathrm{Cover}(S_o[l], G)\right).$$

Since S_o contains all elements of S, the set S does not cover the basis G (see Definition 3.2). ■

3.4.2. Statement and proof of Lemma 3.2

The next step requires to introduce the concept of basis observability matrix. This matrix is denoted by ${V_o}^{(\mu)}$ where $\mu + 1$ is the number of elements in S_o and $B_{\mathcal{M}}$ after the algorithm terminates under one of the conditions $\gamma_r = \emptyset$ or $S_r = \emptyset$, or both. This matrix is later used in the proof of Theorem 3.1, and it is constructed using the returned ordered sets of Algorithm 3.1.

To build this matrix, we take the following steps:

Step 1:	Construct matrix $\Gamma = [g_0 \dots\ g_{n-1}]$ from basis G.
Step 2:	Permutate Γ such that $\tilde{\Gamma}_\mu =\ \Gamma P^T = [B_{\mathcal{M}}[0]\ B_{\mathcal{M}}[1] \dots B_{\mathcal{M}}[\mu]\ \gamma_r]$.
Step 3:	Define coordinate vectors $\tilde{\alpha}_i$, $0 \le i \le \mu$ such that $\tilde{\Gamma}_\mu \tilde{\alpha}_i = \Gamma P^T P \alpha_i = \Gamma\, \alpha_i = S_o[i]$.
Step 4:	Construct matrices Δ_i such that $\Delta_i = diag(\tilde{\alpha}_i), 0 \le i \le \mu$.
Step 5:	Construct matrix Λ using the corresponding eigenvalues of the basis elements of G.
Step 6:	Use matrices Λ and Δ_i to build ${V_o}^{(\mu)}$.

What follows describes each of the above steps in detail.

Step 1:

Under the assumptions that the algorithm has terminated and the output sets are obtained, define as $\Gamma = [g_0 \dots\ g_{n-1}]$ the matrix with as columns, the elements of basis $G = \{g_0, \dots, g_{n-1}\}$, $g_i \in \mathbb{R}^n$. Using this matrix, the coordinate vector α_i of each sensor vector $S_o[i] \in S_o$ with respect to the basis G can be obtained as follows:

$$\alpha_i = \Gamma^{-1} S_o[i], 0 \le i \le \mu\,, \alpha_i \in \mathbb{R}^n.$$

Step 2:

To construct the basis observability matrix, a reorganization of the columns of Γ is necessary. This is done consistently with sets γ_r and $B_{\mathcal{M}}$.

As mentioned in Table 3.1, γ_r is the ordered set that initially contains all the elements of basis G and every $B_{\mathcal{M}}[i]$, $0 \leq i \leq \mu$, is a subset of the cover set of the sensor vector $S_o[i]$. At any algorithm iteration, a subset $B_{\mathcal{M}}[i]$ (for some i) is removed from γ_r and appended to $B_{\mathcal{M}}$, for this reason, the sets $B_{\mathcal{M}}[i]$ are disjoint with γ_r and with each other. Indeed, after the termination of the algorithm, the union of all obtained sets $B_{\mathcal{M}}[i]$s and what is left of γ_r is G.

Therefore, there exists a permutation matrix P that allows to permute the columns of Γ such that they are grouped as follows:

$$\tilde{\Gamma}_\mu = \Gamma P^T = [B_{\mathcal{M}}[0]\ B_{\mathcal{M}}[1] \ldots B_{\mathcal{M}}[\mu]\ \gamma_r].$$

Step 3:

Define vectors $\tilde{\alpha}_i$ such that

$$\tilde{\alpha}_i = P\alpha_i\ , 0 \leq i \leq \mu.$$

Note that

$$\tilde{\Gamma}_\mu \tilde{\alpha}_i = \Gamma P^T P \alpha_i = \Gamma\ \alpha_i = S_o[i].$$

Given the structure of $\tilde{\Gamma}_\mu$ and by construction of set $B_{\mathcal{M}}$, each vector $\tilde{\alpha}_i$ takes the following form:

$$\tilde{\alpha}_i = \begin{bmatrix} {\alpha_r}^{(i)} \\ {\alpha_\beta}^{(i)} \\ {\alpha_o}^{(i)} \end{bmatrix},$$

where

$${\alpha_r}^{(i)} \in \mathbb{R}^{z[i-1]}\ , {\alpha_\beta}^{(i)} \in \mathbb{R}^{N_o[i]}\ , {\alpha_o}^{(i)} = 0 \in \mathbb{R}^{n-z[i]}\ .$$

Note that, based on Algorithm 3.1, $z[l] = \sum_{i=0}^{l} n(B_{\mathcal{M}}[i])$ and $N_o[i] = n(B_{\mathcal{M}}[i])$ with $z[-1] = 0$.

For each $\tilde{\alpha}_i$, $0 \le i \le \mu$, sub-vector ${\alpha_\beta}^{(i)}$ corresponds to $B_{\mathcal{M}}[i]$, ${\alpha_r}^{(i)}$ corresponds to all $B_{\mathcal{M}}[l]$, $0 \le l < i$ and ${\alpha_o}^{(i)} = 0$. Thus,

$$S_o[i] = \tilde{\Gamma}_\mu \tilde{\alpha}_i = \ [B_{\mathcal{M}}[0]\ B_{\mathcal{M}}[1] \dots B_{\mathcal{M}}[\mu]\ \gamma_r] \begin{bmatrix} {\alpha_r}^{(i)} \\ {\alpha_\beta}^{(i)} \\ {\alpha_o}^{(i)} \end{bmatrix} =$$

$$[B_{\mathcal{M}}[0]\ B_{\mathcal{M}}[1] \dots B_{\mathcal{M}}[i-1]][{\alpha_r}^{(i)}] + \ [B_{\mathcal{M}}[i]][{\alpha_\beta}^{(i)}] +$$

$$[B_{\mathcal{M}}[i+1] \dots B_{\mathcal{M}}[\mu]\ \gamma_r][{\alpha_o}^{(i)}].$$

The structure of $\tilde{\alpha}_i$ is essential for understanding the exact structure of ${V_o}^{(\mu)}$ and it is an integral part of later proofs.

Step 4:

Consider the following diagonal matrix with the coordinate vector $\tilde{\alpha}_i$ as the diagonal.

$$\Delta_i = diag(\tilde{\alpha}_i), 0 \le i \le \mu.$$

Step 5:

Knowing that $\tilde{\Gamma}_\mu$ is a column permutation of Γ, for each column in $\tilde{\Gamma}_\mu$ there exists a corresponding eigenvalue, denoted by $\tilde{\lambda}_i, 0 \le i < n$. Construct matrix Λ containing the eigenvalues corresponding with the columns of $\tilde{\Gamma}_\mu$ as follows:

$$\eta = [\tilde{\lambda}_0 \dots \tilde{\lambda}_{n-1}],$$

$$\Lambda = \left[\eta^{T^{\circ 0}}\ \eta^{T^{\circ 1}} \dots \eta^{T^{\circ n-1}}\right],$$

where the operator "∘" denotes elementwise or Hadamard power. [1]

Step 6:

Now using both matrices Λ and Δ_i we construct the basis observability matrix ${V_o}^{(\mu)}$. Denote by $\Lambda(i,j)$ the column submatrix of Λ from column i to column j. Construct the basis observability matrix ${V_o}^{(\mu)}$ as follows:

$${V_o}^{(\mu)} = [\Delta_0 \Lambda(0, z[0]-1\ )\ \Delta_1 \Lambda(z[0], z[1]-1\ ) \dots \Delta_\mu \Lambda(z[\mu-1]\ , z[\mu]-1)\ ].$$

---

[1] For example, $[\tilde{\lambda}_0 \dots \tilde{\lambda}_{n-1}]^{T^{\circ i}} = \left[{\tilde{\lambda}_0}^{\ i} \dots {\tilde{\lambda}_{n-1}}^{\ i}\right]^T$

${V_o}^{(\mu)}$ has $\mu + 1$ submatrices.

Because each column submatrix $\Delta_i \Lambda(z[i-1], z[i]-1\ )$ has $z[i]-z[i-1] = n(B_{\mathcal{M}}[i])$ columns, the sum of the columns of all submatrices of ${V_o}^{(\mu)}$ is

$$n(B_{\mathcal{M}}[0]) + \cdots + n(B_{\mathcal{M}}[\mu]) = \sum_{i=0}^{\mu} n(B_{\mathcal{M}}[i]).$$

Remember that

$$z[l] = \sum_{i=0}^{l} n(B_{\mathcal{M}}[i]).$$

Thus,

$$z[\mu] = \sum_{i=0}^{\mu} n(B_{\mathcal{M}}[i]),$$

and consequently ${V_o}^{(\mu)} \in \mathbb{R}^{n \times z[\mu]}$.

Now we are in the position to state and prove Lemma 3.2.

Lemma 3.2. If $\gamma_r = \emptyset$, then the basis observability matrix ${V_o}^{(\theta)}$ is nonsingular and ${V_o}^{(\theta)} \in \mathbb{R}^{n \times n}$.

Proof of Lemma 3.2. Before proceeding with the proof, recall that $z[l] = \sum_{i=0}^{l} n(B_{\mathcal{M}}[i])$ and $N_o[i] = n(B_{\mathcal{M}}[i])$.

To prove Lemma 3.2, we take the following steps:

Step 1:	Show that ${V_o}^{(\theta)} \in \mathbb{R}^{n \times n}$.
Step 2:	Show that ${V_o}^{(\theta)}$ has a specific block matrix structure corresponding to the structure of coordinate vector $\tilde{\alpha}_i$.
Step 3:	Show that ${V_o}^{(\theta)}$ is an upper triangular block matrix.
Step 4:	Prove that the block diagonal elements of ${V_o}^{(\theta)}$ are nonsingular.
Step 5:	Prove that ${V_o}^{(\theta)}$ is nonsingular due to its upper triangular block matrix structure and the nonsingularity of its block diagonal elements.

Step 1:

If $\gamma_r = \emptyset$ then every single element of γ_r is appended to $B_{\mathcal{M}}$ in the form of sets $B_{\mathcal{M}}[i]$, $0 \leq i \leq \theta$. Thus, we have the following structures for $\tilde{\Gamma}_\theta$ and the basis observability matrix ${V_o}^{(\theta)}$:

$$\tilde{\Gamma}_\theta = [B_{\mathcal{M}}[0]\ B_{\mathcal{M}}[1] \ldots B_{\mathcal{M}}[\theta]\ ].$$

$$V_o^{(\theta)} = [\Delta_0 \Lambda(0, z[0]-1\ )\ \Delta_1 \Lambda(z[0], z[1]-1\ ) \ldots \Delta_\theta \Lambda(z[\theta-1]\ , z[\theta]-1)\ ].$$

Following the definition of basis observability matrix, we know that $V_o^{(\theta)} \in \mathbb{R}^{n \times z[\theta]}$. On the other hand, $z[\theta] = \sum_{i=0}^{\theta} n(B_{\mathcal{M}}[i])$ which together with $\gamma_r = \emptyset$ yields $z[\theta] = n$; thus, $V_o^{(\theta)} \in \mathbb{R}^{n \times n}$.

Step 2:

We know that

$$\Delta_i = diag(\tilde{\alpha}_i) = diag\left(\begin{bmatrix} \alpha_r^{(i)} \\ \alpha_\beta^{(i)} \\ \alpha_o^{(i)} \end{bmatrix}\right), 0 \le i \le \theta$$

$$\alpha_r^{(i)} \in \mathbb{R}^{z[i-1]}, \alpha_\beta^{(i)} \in \mathbb{R}^{N_o[i]}, \alpha_o^{(i)} = 0 \in \mathbb{R}^{n-z[i]}.$$

This indicates that each submatrix $\Delta_i \Lambda(z[i-1],\ z[i]-1)$ in $V_o^{(\theta)}$, has its corresponding coordinate vector $\tilde{\alpha}_i$. Due to the structure of each $\tilde{\alpha}_i$ the sub matrices themselves consist of three sub-matrices stacked on top of each other, each corresponding to one of the sub-vectors of $\tilde{\alpha}_i$ meaning $\alpha_r^{(i)}$, $\alpha_\beta^{(i)}$, and $\alpha_o^{(i)}$.

Denote by $\Delta\alpha_r^{(i)} \in \mathbb{R}^{z[i-1] \times N_o[i]}$, $\Delta\alpha_\beta^{(i)} \in \mathbb{R}^{N_o[i] \times N_o[i]}$ and $\Delta\alpha_o^{(i)} = 0 \in \mathbb{R}^{(n-z[i]) \times N_o[i]}$ the submatrices corresponding to $\alpha_r^{(i)}, \alpha_\beta^{(i)}$ and $\alpha_o^{(i)}$ respectively. We have

$$V_o^{(\theta)} = \begin{bmatrix} \Delta\alpha_r^{(0)} & \Delta\alpha_r^{(1)} & \ldots & \Delta\alpha_r^{(\theta)} \\ \Delta\alpha_\beta^{(0)} & \Delta\alpha_\beta^{(1)} & \ldots & \Delta\alpha_\beta^{(\theta)} \\ \Delta\alpha_o^{(0)} & \Delta\alpha_o^{(1)} & \ldots & \Delta\alpha_o^{(\theta)} \end{bmatrix}.$$

Thus, $V_o^{(\theta)}$ has a block matrix structure.

Step 3:

The following matrix shows that each $\Delta\alpha_\beta^{(i)} \in \mathbb{R}^{N_o[i] \times N_o[i]}$ is a diagonal element of block matrix $V_o^{(\theta)}$. This means that the main diagonal of every $\Delta\alpha_\beta^{(i)}$ is part of the main diagonal of $V_o^{(\theta)}$.

$$V_o^{(\theta)} = \begin{bmatrix} \Delta\alpha_\beta^{(0)} & \Delta\alpha_r^{(0,1)} & \Delta\alpha_r^{(0,2)} & \Delta\alpha_r^{(0,3)} & \dots & \dots & \Delta\alpha_r^{(0,\theta)} \\ 0 & \Delta\alpha_\beta^{(1)} & \Delta\alpha_r^{(1,3)} & \Delta\alpha_r^{(1,3)} & & & \\ & 0 & \Delta\alpha_\beta^{(2)} & \Delta\alpha_r^{(2,3)} & & & \\ & & 0 & \Delta\alpha_\beta^{(3)} & & & \vdots \\ \vdots & & & 0 & & & \\ & \vdots & & & \ddots & & \\ & & \vdots & & & & \Delta\alpha_r^{(\theta-2,\theta)} \\ & & & \vdots & & & \Delta\alpha_r^{(\theta-1,\theta)} \\ 0 & & \dots & & \dots & 0 & \Delta\alpha_\beta^{(\theta)} \end{bmatrix}$$

Where each $\Delta\alpha_r^{(i,j)} \in N_o[i] \times N_o[j]$ is a submatrix of $\Delta\alpha_r^{(j)}$.

Under each block, diagonal element of $V_o^{(\theta)}$, there is a zero block $\Delta_i(\alpha_o^{(i)}) = 0 \in \mathbb{R}^{(n-z[i])\times N_o[i]}$. Thus, $V_o^{(\theta)}$ is an upper triangular block matrix.

Step 4:

Denoting kth element of sub-vector $\alpha_\beta^{(i)}$ by $\alpha_\beta^{(i)}(k-1)$, in more details $V_o^{(\theta)}$ can be presented as follows:

$$V_o^{(\theta)} =$$

$$\begin{bmatrix} \alpha_\beta^{(0)}(0) & \dots & \alpha_\beta^{(0)}(0)\tilde{\lambda}_0^{\ z[0]-1} & & \vdots & & \vdots \\ \vdots & \ddots & \vdots & & & & \\ \alpha_\beta^{(0)}(N_o[0]-1) & \dots & \alpha_\beta^{(0)}(N_o[0]-1)\tilde{\lambda}_{z[0]-1}^{\ z[0]-1} & & \vdots & & \vdots \\ 0 & 0 & 0 & \ddots & & & \\ \vdots & \vdots & \vdots & 0 & \alpha_\beta^{(\theta)}(0)\tilde{\lambda}_{z[\theta-1]}^{\ z[\theta-1]} & \dots & \alpha_\beta^{(\theta)}(0)\tilde{\lambda}_{z[\theta-1]}^{\ z[\theta]-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \alpha_\beta^{(\theta)}(N_o[\theta]-1)\tilde{\lambda}_{z[\theta]-1}^{\ z[\theta-1]} & \dots & \alpha_\beta^{(\theta)}(N_o[\theta]-1)\tilde{\lambda}_{z[\theta]-1}^{\ z[\theta]-1} \end{bmatrix}.$$

Given the structure of $V_o^{(\theta)}$ to prove its nonsingularity, first the nonsingularity of the block diagonal elements of $V_o^{(\theta)}$ is proved, then the nonsingularity of $V_o^{(\theta)}$ is concluded.

With regards to the nonsingularity of any $\Delta\alpha_\beta^{(i)}$, consider

$$\Delta\alpha_\beta^{(i)} = \begin{bmatrix} \alpha_\beta^{(i)}(0)\tilde{\lambda}_{z[i-1]}^{\ z[i-1]} & \dots & \alpha_\beta^{(i)}(0)\tilde{\lambda}_{z[i-1]}^{\ z[i]-1} \\ \vdots & \ddots & \vdots \\ \alpha_\beta^{(i)}(N_o[i]-1)\tilde{\lambda}_{z[i]-1}^{\ z[i-1]} & \dots & \alpha_\beta^{(i)}(N_o[i]-1)\tilde{\lambda}_{z[i]-1}^{\ z[i]-1} \end{bmatrix} =$$

$$\begin{bmatrix} \alpha_\beta^{(i)}(0) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \alpha_\beta^{(i)}(N_o[i]-1) \end{bmatrix} \begin{bmatrix} \tilde{\lambda}_{z[i-1]}^{\ z[i-1]} & \dots & \tilde{\lambda}_{z[i-1]}^{\ z[i]-1} \\ \vdots & \ddots & \vdots \\ \tilde{\lambda}_{z[i]-1}^{\ z[i-1]} & \dots & \tilde{\lambda}_{z[i]-1}^{\ z[i]-1} \end{bmatrix} =$$

$$\begin{bmatrix} \alpha_\beta^{(i)}(1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \alpha_\beta^{(i)}(N_o[i]-1) \end{bmatrix} \begin{bmatrix} \tilde{\lambda}_{z[i-1]}^{\ z[i-1]} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \tilde{\lambda}_{z[i]-1}^{\ z[i-1]} \end{bmatrix} \begin{bmatrix} 1 & \tilde{\lambda}_{z[i-1]} & \dots & \tilde{\lambda}_{z[i-1]}^{\ N_o[i]-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \tilde{\lambda}_{z[i]-1} & \dots & \tilde{\lambda}_{z[i]-1}^{\ N_o[i]-1} \end{bmatrix}.$$

Thus, $\Delta\alpha_{\beta}{}^{(i)}$ results from multiplying three square matrices, two square diagonal matrices, and one square Vandermonde matrix. Considering Definition 3.4 and Assumption 3.1, the diagonal matrices have non-zero elements on their main diagonal. Therefore, they have non-zero determinants and are nonsingular. The third matrix is a square Vandermonde matrix. We know that A is such that $\tilde{\lambda}_i$s are distinct (see Assumption 3.1). Thus, the Vandermonde matrix is nonsingular [15]. The determinant of the product of square matrices is equal to the product of their determinants. Thus, the determinant of any $\Delta\alpha_{\beta}{}^{(i)}$ is non-zero, and all block diagonal elements of $V_o{}^{(\theta)}$ are nonsingular.

Step 5:

Now that we know the block diagonal elements of $V_o{}^{(\theta)}$ are nonsingular, we proceed with the proof of nonsingularity of $V_o{}^{(\theta)}$.

Given the upper triangular block structure of $V_o{}^{(\theta)}$, there exists a proper column permutation P_o that allows the decomposition of each diagonal element to a lower and upper triangular matrix. Thus, we have

$$V_o{}^{(\theta)} P_o = \begin{bmatrix} \Delta\alpha_{\beta}{}^{(0)} & \cdots & \\ & \ddots & \vdots \\ 0 & & \Delta\alpha_{\beta}{}^{(\theta)} \end{bmatrix} P_o = \begin{bmatrix} \Delta\acute{\alpha}_{\beta}{}^{(0)} & \cdots & \\ & \ddots & \vdots \\ 0 & & \Delta\acute{\alpha}_{\beta}{}^{(\theta)} \end{bmatrix} =$$

$$\begin{bmatrix} \Delta\acute{\alpha}_{\beta}{}^{(0)} & \Delta\acute{\alpha}_r{}^{(0,1)} & \Delta\acute{\alpha}_r{}^{(0,2)} & \Delta\acute{\alpha}_r{}^{(0,3)} & \cdots & \cdots & & \Delta\acute{\alpha}_r{}^{(0,\theta)} \\ 0 & \Delta\acute{\alpha}_{\beta}{}^{(1)} & \Delta\acute{\alpha}_r{}^{(1,3)} & \Delta\acute{\alpha}_r{}^{(1,3)} & & & & \\ & 0 & \Delta\acute{\alpha}_{\beta}{}^{(2)} & \Delta\acute{\alpha}_r{}^{(2,3)} & & & & \\ & & 0 & \Delta\acute{\alpha}_{\beta}{}^{(3)} & & & & \vdots \\ \vdots & & & 0 & \ddots & & & \\ & & & & & & \ddots & \Delta\acute{\alpha}_r{}^{(\theta-2,\theta)} \\ & & & & & & & \Delta\acute{\alpha}_r{}^{(\theta-1,\theta)} \\ 0 & & \cdots & & \cdots & & 0 & \Delta\acute{\alpha}_{\beta}{}^{(\theta)} \end{bmatrix} =$$

$$\begin{bmatrix} L_0U_0 & \Delta\acute{\alpha}_r{}^{(0,1)} & \Delta\acute{\alpha}_r{}^{(0,2)} & \Delta\acute{\alpha}_r{}^{(0,3)} & \cdots & \cdots & & \Delta\acute{\alpha}_r{}^{(0,\theta)} \\ 0 & L_1U_1 & \Delta\acute{\alpha}_r{}^{(1,3)} & \Delta\acute{\alpha}_r{}^{(1,3)} & & & & \\ & 0 & L_2U_2 & \Delta\acute{\alpha}_r{}^{(2,3)} & & & & \\ & & 0 & L_3U_3 & & & & \vdots \\ \vdots & & & 0 & \ddots & & & \\ & & & & & & \ddots & \Delta\acute{\alpha}_r{}^{(\theta-2,\theta)} \\ & & & & & & & \Delta\acute{\alpha}_r{}^{(\theta-1,\theta)} \\ 0 & & \cdots & & \cdots & & 0 & L_{\theta}U_{\theta} \end{bmatrix} =$$

$$\begin{bmatrix} L_0 & & \dots & 0 \\ 0 & L_1 & & \\ \vdots & 0 & \ddots & \\ 0 & \dots & 0 & L_\theta \end{bmatrix} \begin{bmatrix} U_0 & {L_0}^{-1}\Delta\acute{\alpha}_r{}^{(0,1)} & {L_0}^{-1}\Delta\acute{\alpha}_r{}^{(0,2)} & {L_0}^{-1}\Delta\acute{\alpha}_r{}^{(0,3)} & \dots & \dots & {L_0}^{-1}\Delta\acute{\alpha}_r{}^{(0,\theta)} \\ 0 & U_1 & {L_1}^{-1}\Delta\acute{\alpha}_r{}^{(1,2)} & {L_1}^{-1}\Delta\acute{\alpha}_r{}^{(1,3)} & & & \\ & 0 & U_2 & {L_2}^{-1}\Delta\acute{\alpha}_r{}^{(2,3)} & & & \\ & & 0 & U_3 & & & \vdots \\ \vdots & & & 0 & \ddots & & \\ & & & & & \ddots & {L_{\theta-2}}^{-1}\Delta\acute{\alpha}_r{}^{(\theta-2,\theta)} \\ & & & & & & {L_{\theta-1}}^{-1}\Delta\acute{\alpha}_r{}^{(\theta-1,\theta)} \\ 0 & & \dots & & \dots & 0 & U_\theta \end{bmatrix}$$

$$= L_o U_o.$$

We know that the main diagonal of every $\Delta\acute{\alpha}_\beta{}^{(i)}$ is on the main diagonal of $V_o{}^{(\theta)}P_o$; thus, matrix $V_o{}^{(\theta)}P_o$ is decomposed into a lower and upper triangular matrix. The nonsingularity of every matrix $\Delta\alpha_\beta{}^{(i)}$ and consequently, every matrix $\Delta\acute{\alpha}_\beta{}^{(i)}$, guarantees that all diagonal elements of L_o and U_o are non-zero. Thus, $V_o{}^{(\theta)}P_o$ and $V_o{}^{(\theta)}$ have non-zero determinants and are nonsingular. ■

3.4.3. Proof of Theorem 3.1

The n-horizon observable schedule $\Sigma_n = \{\sigma_{k_0}, \sigma_{k_0+1}, \dots, \sigma_{k_0+n-1}\}$ has the following form

$$\sigma_{k_0} = 0,$$

$$\vdots$$

$$\sigma_{k_0+z[0]-1} = 0,$$

$$\sigma_{k_0+z[0]} = 1,$$

$$\vdots$$

$$\sigma_{k_0+z[1]-1} = 1,$$

$$\vdots$$

$$\sigma_{k_0+z[\theta-1]} = \theta,$$

$$\vdots$$

$$\sigma_{k_0+z[\theta]-1} = \theta.$$

Therefore,

$$y(k) = S_o[\sigma_k]^T x(k)\,,\, \sigma_k \in \Sigma_n\,,\, k_0 \leq k < k_0 + n.$$

We want to show that this schedule has an observability matrix ϕ_n of rank n. We know that

$$\tilde{\Gamma}_\theta \tilde{\alpha}_i = \Gamma P^T P \alpha_i = \Gamma \alpha_i = S_o[i], \forall\, 0 \leq i \leq\ \theta,$$

where

$$\tilde{\Gamma}_\theta = [\tilde{g}_0 \cdots\ \tilde{g}_{n-1}].$$

Because S covers the basis G, Lemma 3.1 tells us that $\gamma_r = \emptyset$ when the algorithm terminates. On the other hand, Lemma 3.2 indicates that if $\gamma_r = \emptyset$ then there exists a basis observabillity matrix ${V_o}^{(\theta)}$ such that ${V_o}^{(\theta)} \in \mathbb{R}^{n\times n}$ and ${V_o}^{(\theta)}$ is nonsingular. Thus, we have

$${V_o}^{(\theta)} = \begin{bmatrix} \tilde{\alpha}_0(0) & \cdots & \tilde{\alpha}_1(0){\tilde{\lambda}_0}^{z[0]} & \cdots & \tilde{\alpha}_\theta(0){\tilde{\lambda}_0}^{z[\theta]-1} \\ \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots \\ \tilde{\alpha}_0(n-1) & \cdots & \tilde{\alpha}_1(n-1){\tilde{\lambda}_{n-1}}^{z[0]} & \cdots & \tilde{\alpha}_\theta(n-1){\tilde{\lambda}_{n-1}}^{z[\theta]-1} \end{bmatrix},$$

where $z[\theta] = n$ (see Proof 3.2) and $\tilde{\alpha}_i(k-1)$ denotes the kth element of the coordinate vector $\tilde{\alpha}_i$. Now we show that $\tilde{\Gamma}_\theta {V_o}^{(\theta)} = {\phi_n}^T$, where ϕ_n is the observability matrix of Σ_n.

For the first column of the first column submatrix of ${V_o}^{(\theta)}$ we have

$$[\tilde{g}_0 \cdots\ \tilde{g}_{n-1}] \begin{bmatrix} \tilde{\alpha}_0(0) \\ \vdots \\ \tilde{\alpha}_0(n-1) \end{bmatrix} = \tilde{\alpha}_0(0)\tilde{g}_0 + \cdots + \tilde{\alpha}_0(n-1)\tilde{g}_{n-1} = \tilde{\Gamma}_\theta \tilde{\alpha}_0 = S_o[0];$$

The same goes for the rest of the columns of the submatrix, such that we have $A^T S_o[0], \ldots, {A^T}^{z[0]-1} S_o[0]$.

For the first column of the second column submatrix of ${V_o}^{(\theta)}$ we have

$$\begin{aligned}[\tilde{g}_0 \cdots\ \tilde{g}_{n-1}] \begin{bmatrix} \tilde{\alpha}_1(0){\tilde{\lambda}_0}^{z[0]} \\ \vdots \\ \tilde{\alpha}_1(n-1){\tilde{\lambda}_{n-1}}^{z[0]} \end{bmatrix} &= \tilde{\alpha}_1(0){\tilde{\lambda}_0}^{z[0]}\tilde{g}_0 + \cdots + \tilde{\alpha}_1(n-1){\tilde{\lambda}_{n-1}}^{z[0]}\tilde{g}_{n-1} \\ &= \tilde{\alpha}_1(0){A^T}^{z[0]}\tilde{g}_0 + \cdots + \tilde{\alpha}_1(n-1){A^T}^{z[0]}\tilde{g}_{n-1} = \end{aligned}$$

$$A^{T^{z[0]}}(\tilde{\alpha}_1(0)\tilde{g}_0 + \cdots + \tilde{\alpha}_1(n-1)\tilde{g}_{n-1}) = A^{T^{z[0]}}\tilde{\Gamma}_\theta\tilde{\alpha}_1 = A^{T^{z[0]}}S_o[1]\,;$$

The same goes for the rest of the columns of the submatrix, such that we have $A^{T^{z[0]+1}}S_o[1], \dots, A^{T^{z[1]-1}}S_o[1]$.

We continue this for the rest of the submatrices and their columns until finally, for the last column of the $\theta + 1$-th column submatrix we have

$$[\tilde{g}_0 \dots \tilde{g}_{n-1}]\begin{bmatrix} \tilde{\alpha}_\theta(0)\tilde{\lambda}_0{}^{n-1} \\ \vdots \\ \tilde{\alpha}_\theta(n-1)\tilde{\lambda}_{n-1}{}^{n-1} \end{bmatrix} = \tilde{\alpha}_\theta(0)\tilde{\lambda}_0{}^{n-1}\tilde{g}_0 + \cdots + \tilde{\alpha}_\theta(n-1)\tilde{\lambda}_{n-1}{}^{n-1}\,\tilde{g}_{n-1} =$$

$$\tilde{\alpha}_\theta(0)A^{T^{n-1}}\tilde{g}_0 + \cdots + \tilde{\alpha}_\theta(n-1)A^{T^{n-1}}\tilde{g}_{n-1} =$$

$$A^{T^{n-1}}(\tilde{\alpha}_\theta(0)\tilde{g}_0 + \cdots + \tilde{\alpha}_\theta(n-1)\tilde{g}_{n-1}) = A^{T^{n-1}}\tilde{\Gamma}_\theta\tilde{\alpha}_\theta = A^{T^{n-1}}S_o[\theta];$$

therefore,

$$\tilde{\Gamma}_\theta V_o{}^{(\theta)} = \left[S_o[0] \;\dots\; A^{T^{z[0]}}S_o[1] \;\dots\; A^{T^{n-1}}S_o[\theta]\right] = \phi_n{}^T.$$

We know that $\tilde{\Gamma}_\theta$ consists of linearly independent columns; thus, it is nonsingular. Given that both matrices $\tilde{\Gamma}_\theta$ and $V_o{}^{(\theta)}$ are square and nonsingular, $\phi_n{}^T$ is nonsingular as well. Thus, ϕ_n has rank n, and the system under the schedule Σ_n is n-horizon observable. ■

Remarks

An important feature of the schedule Σ_n is that every sensor vector $S_o[i]$ is repeated exactly $N_o[i]$ times. It is of interest to investigate whether the rotation of schedule elements of Σ_n could change the observability results. The uniqueness of this scheduling structure is another topic of interest that deserves further investigation. Another question is whether the condition "*S covers basis G*" is a necessary and sufficient condition. These cases are numerically examined in the next chapter, where the results of the numerical implementation are provided and briefly discussed.

4 Algorithm numerical implementation

In this chapter, the proposed algorithm is implemented, and numerical examples are provided to verify the theoretical results of Chapter 3. The algorithm is implemented in Python language[2].

The code accepts the system matrix A, left eigenvectors of A, and the desired set of sensor vectors and returns several values. A Boolean value indicating whether the sensor set covers the basis; a list of sensor vectors corresponding to S_o; a list of integers N_o indicating the number of times the sensors of S_o should be repeated in the proposed schedule of Theorem 3.1 and, finally, the rank of the observability matrix of the schedule of Theorem 3.1.

4.1. Examples

Let A be an arbitrary matrix that follows Assumption 3.1. Thus, it has n linearly independent eigenvectors.

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 11 & 0 & 0 & 0 \\ 4 & 4 & -8 & 0 & 0 \\ 5 & 8 & 0 & 7 & 0 \\ 7 & 7 & 3 & 0 & 6 \end{bmatrix}.$$

[2] For the development of the code and its version control, a GitHub repository is used. The repository is public and available to anyone interested in reviewing or testing the algorithm. Comments and explanations are provided throughout the code. The link to the repository is:

https://github.com/abiglary1372/Sensor-Observability-Filter.

Consider basis G with left eigenvectors of A as its elements.

$$G = \{[1\ 0\ 0\ 0\ 0]^T, [\frac{-68}{171}\ \frac{-4}{19}\ \ 1\ \ 0\ \ 0]^T, [\frac{33}{35}\ \frac{-11}{7}\ \frac{3}{14}\ \ 0\ \ 1]^T,$$
$$[\tfrac{1}{6} - 2\ 0\ 1\ 0]^T, [\tfrac{1}{5}\ 1\ 0\ 0\ 6]^T\}.$$

Now consider the following examples. In each example, sensor set S is intentionally designed to cover different scenarios.

Example 4.1. Consider the following sensor vector set S, where S covers the basis G.

$$S = \{S[0] = [3.0332, -5.1353, 2.6428, 0, 3]^T, S[1] = [5.1047, -22.2857, 0.8571, 8, 4]^T,$$
$$S[2] = [0.6, 3, 0, 0, 0]^T, S[3] = [1.1666, -14, 0, 7, 0]^T, S[4] = [1.2, 6, 0, 0, 0]^T$$
$$S[5] = [2.5784, -6.9172, 3.8571, 0, 4]\}$$

In the sensor set S, no sensor alone covers the basis G. Table 4.1 shows that no n-horizon observable sensor schedules can be constructed with only one sensor.

Table 4.1: n-horizon sensor schedules over the set S, involving only one sensor.

	σ_0	σ_1	σ_2	σ_3	σ_4	rank(ϕ_n)
$\Sigma_{n,1}$	0	0	0	0	0	3
$\Sigma_{n,2}$	1	1	1	1	1	2
$\Sigma_{n,3}$	2	2	2	2	2	1
$\Sigma_{n,4}$	3	3	3	3	3	1
$\Sigma_{n,5}$	4	4	4	4	4	1
$\Sigma_{n,6}$	5	5	5	5	5	2

The schedule integer values in Table 4.1 refer to the indices of the ordered set S.

Tables 4.2 and 4.3 present the outputs of numerical implementation of the algorithm where different combinations of sensors are used to construct sensor schedules.

Table 4.2: Sensor vectors of the set S_o and their corresponding coordinate vector α_i.

S_o	α_i	N_o
$S_o[0] = [3.0332, -5.1353, 2.6428, 0, 3]^T$	$\alpha_0 = [1,2,3,0,0]^T$	3
$S_o[1] = [5.1047, -22.2857, 0.8571, 8, 4]^T$	$\alpha_1 = [0,0,4,8,0]^T$	1
$S_o[2] = [0.6, 3, 0, 0, 0]^T$	$\alpha_2 = [0,0,0,0,3]^T$	1

Table 4.3: n-horizon sensor schedules over the set S_o and their corresponding observability matrix rank.

	σ_0	σ_1	σ_2	σ_3	σ_4	$\mathrm{rank}(\phi_n)$
$\Sigma_{n,1}$	0	0	0	1	2	5
$\Sigma_{n,2}$	1	2	0	0	0	5
$\Sigma_{n,3}$	0	1	0	0	2	5
$\Sigma_{n,4}$	2	1	0	0	0	5
$\Sigma_{n,5}$	0	0	1	1	2	5
$\Sigma_{n,6}$	0	1	0	1	2	5
$\Sigma_{n,7}$	0	2	0	1	1	5
$\Sigma_{n,8}$	1	2	1	0	0	5
$\Sigma_{n,9}$	1	2	0	1	2	4
$\Sigma_{n,10}$	2	0	2	1	2	3
$\Sigma_{n,11}$	2	0	1	1	2	4

Note that the schedule integer values in Table 4.3 refer to the indices of the ordered set S_o.

Example 4.2. Consider the following sensor set S, where S does not cover the basis G.

$$S = \{S[0] = [-0.7953, -0.4210, 2, 0, 0]^T, S[1] = [1.3333, -16, 0, 8, 0]^T,$$

$$S[2] = [1.1, -3, 0, 3, 0]^T,$$

$$S[3] = [1.1666, -14, 0, 7, 0]^T, S[4] = [1.2, 6, 0, 0, 0]^T,$$

$$S[5] = [-1.1929, -0.6315, 3, 0, 0]^T\}$$

Table 4.4 shows that no n-horizon observable sensor schedules can be constructed with only one sensor.

Table 4.4: n-horizon sensor schedules over the set S involving only one sensor.

	σ_0	σ_1	σ_2	σ_3	σ_4	$\mathrm{rank}(\phi_n)$
$\Sigma_{n,1}$	0	0	0	0	0	1
$\Sigma_{n,2}$	1	1	1	1	1	1
$\Sigma_{n,3}$	2	2	2	2	2	2
$\Sigma_{n,4}$	3	3	3	3	3	1
$\Sigma_{n,5}$	4	4	4	4	4	1
$\Sigma_{n,6}$	5	5	5	5	5	1

Note that the schedule integer values in Table 4.4 refer to the indices of the ordered set S.

Tables 4.5 and 4.6 present the outputs of numerical implementation of the algorithm where different combinations of sensors are used to construct sensor schedules.

Table 4.5: Sensor vectors of the set S_o and their corresponding coordinate vector α_i

$\boldsymbol{S_o}$	$\boldsymbol{\alpha_i}$	$\boldsymbol{N_o}$
$\boldsymbol{S_o}[\mathbf{0}] = [1.1, -3, 0, 3, 0]^T$	$\alpha_0 = [0,0,0,3,3]^T$	2
$\boldsymbol{S_o}[\mathbf{1}] = [-0.7953, -0.4210, 2, 0, 0]^T$	$\alpha_1 = [0,2,0,0,0]^T$	1
$\boldsymbol{S_o}[\mathbf{2}] = [1.3333, -16, 0, 8, 0]^T$	$\alpha_2 = [0,0,0,8,0]^T$	0
$\boldsymbol{S_o}[\mathbf{3}] = [1.1666, -14, 0, 7, 0]^T$	$\alpha_3 = [0,0,0,7,0]^T$	0
$\boldsymbol{S_o}[\mathbf{4}] = [1.2, 6, 0, 0, 0]^T$	$\alpha_4 = [0,0,0,0,6]^T$	0
$\boldsymbol{S_o}[\mathbf{5}] = [-1.1929, -0.6315, 3, 0, 0]^T$	$\alpha_5 = [0,3,0,0,0]^T$	0

Table 4.6: n-horizon sensor schedules and their corresponding observability matrix rank.

	$\boldsymbol{\sigma_0}$	$\boldsymbol{\sigma_1}$	$\boldsymbol{\sigma_2}$	$\boldsymbol{\sigma_3}$	$\boldsymbol{\sigma_4}$	$\mathbf{rank}(\boldsymbol{\phi}_\mathbf{n})$
$\boldsymbol{\Sigma_{n,1}}$	0	0	1	2	3	3
$\boldsymbol{\Sigma_{n,2}}$	0	0	1	3	4	3
$\boldsymbol{\Sigma_{n,3}}$	0	0	1	4	5	3
$\boldsymbol{\Sigma_{n,4}}$	0	0	1	5	6	3
$\boldsymbol{\Sigma_{n,5}}$	5	0	2	4	3	3
$\boldsymbol{\Sigma_{n,6}}$	3	4	2	0	3	2
$\boldsymbol{\Sigma_{n,7}}$	5	5	1	3	3	2
$\boldsymbol{\Sigma_{n,8}}$	2	1	1	4	2	3
$\boldsymbol{\Sigma_{n,9}}$	2	0	4	0	3	2

Note that each integer value in Table 4.6 refers to an index of the ordered set S_o.

4.2. Discussion and verification

We can see that schedule $\Sigma_{n,1}$ of Table 4.3 supports the main result. Building a schedule using sensors vectors of S_o and following the proposed schedule structure of Theorem 3.1, a full rank observability matrix for schedule $\Sigma_{n,1}$ is obtained.

Looking at the rest of the schedules in Table 4.3, the first thing that we notice is that the proposed schedule structure of Theorem 3.1 is not the only schedule structure that can provide observability. This can be seen in schedules $\Sigma_{n,5}$, $\Sigma_{n,6}$, $\Sigma_{n,7}$ and $\Sigma_{n,8}$ in Table 4.3, which may suggest that there is a more general schedule structure that can determine observable schedules. The second noticeable behavior is the change of time steps. It seems that if a schedule is observable, changing the time steps of the sensors or in other words rotating the schedule elements does not disrupt the observability. This behavior can be seen in schedules $\Sigma_{n,1}$ to $\Sigma_{n,4}$ and $\Sigma_{n,5}$ to $\Sigma_{n,8}$ of Table 4.3.

Schedules $\Sigma_{n,9}$ to $\Sigma_{n,11}$ in Table 4.3 show that even if all sensors of S_o are used in the schedule, there is no guarantee that the schedule will provide observability unless the proposed schedule structure of Theorem 3.1 is used.

Concerning Example 4.2, Table 4.6 suggests that, regardless of what combination of sensors is used, there is no combination of sensors that can provide an observable n-horizon schedule. In this specific example, the condition considered in Theorem 3.1 turns out to be also necessary for the existence of a n-horizon sensor schedule. This remark raises a point that will be the subject of future investigation.

5 Conclusion and future developments

In this thesis, the concept of sensor scheduling was introduced. The observability of systems under scheduled measurements was investigated, resulting in the introduction of the concept of measurement time horizon and extension of the definition of observability to the case of linear discrete-time systems with time-variant measurements.

The conditions guaranteeing the existence of an observable schedule were investigated, and new definitions and results were introduced to provide a simplified proof of the theorem in [11] and a clearer characterization of the proposed theorem for the definition of an observable sensor schedule. In particular, an algorithm was devised capable of defining a set of sensors that constitute an observable sensor schedule of a specific structure.

The algorithm was numerically implemented. The numerical results suggest that a more general structure of an observable schedule could exist. This can be a subject of future development.

In this work, all conditions were investigated for sensors with $1 \times n$ measurement matrices. A future development of this work can also consist in including conditions supporting the existence of observable schedules for sensors with measurement matrices with a higher dimension.

Another topic for future work can be the application of a Kalman filtering approach with scheduled measurements. To do so, we need to develop algorithms that take into consideration both the observability and estimation error in the optimization problem. When developing these algorithms, the computational costs of these algorithms are an important factor to take into consideration. For example, the offline determination of the sensor set, and identification of all observable sensor subsets can be considered as the first step for the development of this approach, then the choice between observable sensor schedules can be taken (possibly online) based on the criterion of minimizing the variance of the estimation error.

Bibliography

[1] Diddigi, R. B., Prabuchandran, K. J., & Bhatnagar, S. (2018). Novel sensor scheduling scheme for intruder tracking in energy efficient sensor networks. IEEE Wireless Communications Letters, 7(5), 712-715.

[2] Han, D., Wu, J., Zhang, H., & Shi, L. (2017). Optimal sensor scheduling for multiple linear dynamical systems. Automatica, 75, 260-270.

[3] Yang, W., Chen, G., Wang, X., & Shi, L. (2014). Stochastic sensor activation for distributed state estimation over a sensor network. Automatica, 50(8), 2070-2076.

[4] Vitus, M. P., Zhang, W., Abate, A., Hu, J., & Tomlin, C. J. (2012). On efficient sensor scheduling for linear dynamical systems. Automatica, 48(10), 2482-2493.

[5] Yang, L., Rao, H., Lin, M., Xu, Y., & Shi, P. (2022). Optimal sensor scheduling for remote state estimation with limited bandwidth: a deep reinforcement learning approach. Information Sciences, 588, 279-292.

[6] Mo, Y., Garone, E., & Sinopoli, B. (2014). On infinite-horizon sensor scheduling. Systems & control letters, 67, 65-70.

[7] Mo, Y., Ambrosino, R., & Sinopoli, B. (2011). Sensor selection strategies for state estimation in energy constrained wireless sensor networks. Automatica, 47(7), 1330-1338.

[8] Han, D., Zhang, H., & Shi, L. (2013, June). An event-based scheduling solution for remote state estimation of two LTI systems under bandwidth constraint. In 2013 American Control Conference (pp. 3314-3319). IEEE.

[9] Ahn, H., & Danielson, C. (2019, July). Moving Horizon Sensor Selection for Reducing Communication Costs with Applications to Internet of Vehicles. In 2019 American Control Conference (ACC) (pp. 1464-1469). IEEE.

[10] Bian, F., Kempe, D., & Govindan, R. (2006, April). Utility based sensor selection. In Proceedings of the 5th international conference on Information processing in sensor networks (pp. 11-18).

[11] Ilkturk, U. (2015). Observability methods in sensor scheduling. Arizona State University.

[12] Kalman, R. E. (1960, August). On the general theory of control systems. In Proceedings First International Conference on Automatic Control, Moscow, USSR (pp. 481-492).

[13] Kalman, R. E. (1963). Mathematical description of linear dynamical systems. Journal of the Society for Industrial and Applied Mathematics, Series A: Control, 1(2), 152-192.

[14] Williams, R. L., & Lawrence, D. A. (2007). Linear state-space control systems. John Wiley & Sons.

[15] Turner, L. R. (1966). Inverse of the Vandermonde matrix with applications (No. NASA-TN-D-3547).

List of Tables

List of symbols

Variable	Description
x	System state
y	System output
k_0	Starting time of measurement
x_0	Initial state
A	System matrix
C	Output matrix
λ_i	Eigenvalue
$M(k_0, k)$	Gramian matrix
ϕ_N	Observability matrix
T_N	Measurement time horizon
C_k	Time-variant output matrix
$M_T(k_0, k_0 + N - 1)$	Time variant Gramian matrix
S	Sensor set
$S[i]$	Sensor vector
Σ	Sensor schedule
Σ_N	N-horizon sensor schedule
σ_{k_0}	Schedule element
G	Basis
β	Cover set
g_i	Basis element
α_i	Coordinate vector
$\tilde{\alpha}_i$	Permutated coordinate
$V_o{}^{(\mu)}$	Basis observability matrix

Acknowledgments

I want to express my deepest thanks to my advisers, professor Alessandro Colombo and Marcello Farina, who helped and guided me through every step of this thesis. Without their help and deep knowledge and understanding of the topic, this work would not have been possible.

I must also thank the polytechnic university of Milan for providing me with an environment to learn and grow.

Finally, I would like to thank my parents for all their love and support throughout this journey, without which none of this would have been possible. I am very grateful to all the friends and people who always offer support and love.