Sensor Management Algorithms for Measurement of Diffusion Processes

by

Anbar Najam

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2016 by the
Graduate Supervisory Committee:

Douglas Cochran, Chair
Pavan Turaga
Chao Wang

ARIZONA STATE UNIVERSITY

May 2016

ABSTRACT

Modern systems that measure dynamical phenomena often have limitations as to how many sensors can operate at any given time step. This thesis considers a sensor scheduling problem in which the source of a diffusive phenomenon is to be localized using single point measurements of its concentration. With a linear diffusion model, and in the absence of noise, classical observability theory describes whether or not the system's initial state can be deduced from a given set of linear measurements. However, it does not describe to what degree the system is observable. Different metrics of observability have been proposed in literature to address this issue. Many of these methods are based on choosing optimal or sub-optimal sensor schedules from a predetermined collection of possibilities. This thesis proposes two greedy algorithms for a one-dimensional and two-dimensional discrete diffusion processes. The first algorithm considers a deterministic linear dynamical system and deterministic linear measurements. The second algorithm considers noise on the measurements and is compared to a Kalman filter scheduling method described in published work.

TABLE OF CONTENTS

Page

LIST OF FIGURES iv

1 INTRODUCTION 1

1.1 Introduction 1

1.2 Sensor Scheduling 2

1.3 Diffusion Applications 2

1.4 Observability Basics 3

1.5 Diffusion Matrix 6

1.5.1 Diffusion of a One-Dimensional System 6

1.5.2 Diffusion of a Two-Dimensional System 8

2 BACKGROUND 11

2.1 Relation between the Observability Gramian and Estimation Covariance 11

2.2 Observability Measure Metrics 13

3 MATHEMATICAL FORMULATION 15

3.1 Sensor Scheduling Algorithm for Deterministic Measurements 15

3.2 Sensor Scheduling Algorithm for Noisy Measurements 18

3.2.1 Method based on the Pseudoinverse 18

3.2.2 Method based on the Kalman Filter 21

4 RESULTS 24

4.1 Simulation Parameters 24

4.2 Deterministic Measurements 24

4.2.1 One-Dimensional Diffusion System 26

4.2.2 Two-Dimensional Diffusion System 28

4.3 Noisy Measurements 31

CHAPTER Page

4.3.1 One-Dimensional Diffusion System 33
4.3.2 Two-Dimensional Diffusion System 35
4.4 Comparison of Algorithm 2 and Algorithm 3 38
4.4.1 One-Dimensional Diffusion System 39
4.4.2 Two-Dimensional Diffusion System 40
5 CONCLUSIONS 42
REFERENCES 44

LIST OF FIGURES

Figure Page

1.1 The Figure Illustrates the Placement of $U_{i,j}$ on a Two-Dimensional Grid Where $N = 7$. The Label on Each Point Represents the Position in the Vector b. 9

4.1 Results of Algorithm 1 Illustrate the Condition Number $\kappa(\Phi)$ as a Function of the $N \times N$ Dimension of the One-Dimensional Diffusive System A for a $\gamma = 0.04$. 25

4.2 Results of Algorithm 1 Illustrate the Condition Number $\kappa(\Phi)$ as a Function of the $N \times N$ Dimension of the One-Dimensional Diffusive System A for a $\gamma = 0.004$. 25

4.3 Sensor Schedule Determined by Algorithm 1 for the One-Dimensional Diffusive System with $\gamma = 0.04$ and $N = 25$. The Graph Illustrates the Sensor Chosen as the System Propagates for Time Steps $k = 0, \ldots, 24$.. 26

4.4 Algorithm 1 for the One-Dimensional Case with $\gamma = 0.04$ and $N = 25$. The Graph Illustrates the State $\hat{x}_{24}$. 27

4.5 Algorithm 1 for the One-Dimensional Case with $\gamma = 0.04$ and $N = 25$. The Graph Illustrates the State $\hat{x}_{12}$. 27

4.6 Algorithm 1 for the One-Dimensional Case with $\gamma = 0.04$ and $N = 25$. The graph illustrates the reconstructed initial state $\hat{x}_0$. 28

4.7 Sensor Schedule Determined by Algorithm 1 for the Two-Dimensional Diffusive system with $\gamma = 0.04$ and $N = 49$. The Graph Illustrates the Time Steps at Which a Sensor was Chosen at Each Grid Point Corresponding to $U_{i,j}$. 29

4.8 Algorithm 1 for the Two-Dimensional Case with $\gamma = 0.04$ and $N = 49$. The Graph Illustrates the State $\hat{x}_{48}$. 30

Figure Page

4.9 Algorithm 1 for the Two-Dimensional Case with $\gamma = 0.04$ and $N = 49$. The Graph Illustrates the State $\hat{x}_{24}$. 30

4.10 Algorithm 1 for the Two-Dimensional Case with $\gamma = 0.04$ and $N = 49$. The Graph Illustrates the Reconstructed Initial State $\hat{x}_0$. 31

4.11 Results of Algorithm 2 Illustrate the Trace of the Inverse of the Observability Gramian trace$((\Phi^T\Phi)^{-1})$ as a Function of the $N \times N$ Dimension of the One-Dimensional Diffusive System A for a $\gamma = 0.04$. 32

4.12 Results of Algorithm 2 Illustrate the Trace of the Inverse of the Observability Gramian trace$((\Phi^T\Phi)^{-1})$ as a Function of the $N \times N$ Dimension of the One-Dimensional Diffusive System A for a $\gamma = 0.004$. 32

4.13 Sensor Schedule Determined by Algorithm 2 for the One-Dimensional Diffusive System with $\gamma = 0.04$, $N = 25$, and Measurement Noise Variance of 0.1^2. The Graph Illustrates the Sensor Chosen as the System Propagates for Time Steps $k = 0, \ldots, 24$. 33

4.14 Algorithm 2 for the One-Dimensional Case with $\gamma = 0.04$, $N = 25$, and Measurement Noise Variance of 0.1^2. The Graph Illustrates the State $\hat{x}_{24}$. 34

4.15 Algorithm 2 for the One-Dimensional Case with $\gamma = 0.04$, $N = 25$, and Measurement Noise Variance of 0.1^2. The Graph Illustrates the State $\hat{x}_{12}$. 34

4.16 Algorithm 2 for the One-Dimensional Case with $\gamma = 0.04$, $N = 25$, and Measurement Noise Variance of 0.1^2. The Graph Illustrates the Reconstructed Initial State $\hat{x}_0$. 35

Figure Page

4.17 Sensor Schedule Determined by Algorithm 2 for the Two-Dimensional Diffusive System with $\gamma = 0.04$, $N = 49$, and Measurement Noise Variance of 0.01^2. The Graph Illustrates the Time Steps at Which a Sensor was Chosen at Each Grid Point Corresponding to $U_{i,j}$. 36

4.18 Algorithm 2 for the Two-Dimensional Case with $\gamma = 0.04$, $N = 49$, and Measurement Noise Variance of 0.1^2. The Graph Illustrates the State $\hat{x}_{48}$ 37

4.19 Algorithm 2 for the Two-Dimensional Case with $\gamma = 0.04$, $N = 49$, and Measurement Noise Variance of 0.1^2. The Graph Illustrates the State $\hat{x}_{24}$ 37

4.20 Algorithm 2 for the Two-Dimensional Case with $\gamma = 0.04$, $N = 49$, and Measurement Noise Variance of 0.1^2. The Graph Illustrates the Reconstructed Initial State $\hat{x}_0$. 38

4.21 Average Mean-Square Error for Algorithm 2 and Algorithm 3 with $\breve{P}_0 = 100^2 I$ for $N = 25$, $\gamma = 0.04$ for a One-Dimensional Diffusion System. 39

4.22 Average Mean-Square Error for Algorithm 2 and Algorithm 3 with $\breve{P}_0 = 100^2 I$ for $N = 25$, $\gamma = 0.004$ for a One-Dimensional Diffusion System. 39

4.23 Average Mean-Square Error for Algorithm 2 and Algorithm 3 with $\breve{P}_0 = 100^2 I$ for $N = 49$, $\gamma = 0.04$ for a Two-Dimensional Diffusion System. 40

Figure Page

4.24 Average Mean-Square Error for Algorithm 2 and Algorithm 3 with $\breve{P}_0 = 100^2 I$ for $N = 49$, $\gamma = 0.004$ for a Two-Dimensional Diffusion System. 41

Chapter 1

INTRODUCTION

1.1 Introduction

Modern systems for measuring spatial dynamic phenomena are often comprised of a distributed network of individual sensors that may be individually controlled. In many cases, all the sensors do not operate simultaneously due to constraints, such as bandwidth, sampling rate, and power [1, 2]. For certain systems, such as active sonar networks, only one sensor may be activated at a time to avoid signal interference [2]. The problem of sensor selection arises in many applications, such as defense and surveillance [1], management of wireless sensor networks [3], monitoring chemical plants [4], and robotics [5]. To accommodate such constraints, it is desirable to find optimal or near-optimal sensor schedules as the dynamical system propagates in time.

This thesis considers two related variants of this general sensor management problem, both of which entail known and deterministic linear system dynamics and linear measurements. In the first, the measurements are also deterministic while in the second, the measurements are affected by additive Gaussian noise. In both cases, determining the system's initial condition from the measurements will enable its entire state trajectory to be deduced. With deterministic measurements, reconstructing the initial condition from the measurements is seen to be an observability problem. The criterion for optimal sensor scheduling in this situation is taken to be the numerical conditioning of the equation for the initial condition in terms of the measurement sequence as in [6].

With noisy measurements, it will generally be impossible to obtain the initial

condition exactly. Rather, it must be estimated from the noisy measurement data, and the optimality criterion will be in terms of the statistics of this estimate. With Gaussian measurement noise, the problem of determining an optimal measurement sequence has the character of a Kalman filtering problem and has been studied in [7, 8]. The approach developed in this thesis is compared both qualitatively and quantitatively with the one described in [8].

1.2 Sensor Scheduling

An optimal sensor schedule can be resolved a priori when the state and noise properties are known [7]. With this knowledge, the schedule can be derived in an open-loop method before any measurements are taken [8]. The problem in sensor scheduling is selecting k sensors from n possible sensors at each time step to optimize a particular performance metric. Given a finite library of sensor configurations from which to choose a measurement in each time epoch, determining an optimal sequence of sensor configurations is generally NP-hard [9]. For this reason, sub-optimal "greedy" algorithms are proposed in this thesis for both problems described above in section 1.1.

1.3 Diffusion Applications

Many physical phenomena propagate through a medium by diffusion, and modeling of such propagation is well studied in mathematical and scientific literature. An important physical example is diffusion of chemical or biological particulates in aerosol, as might arise if such a contaminant is released into air or water. Such leaks can cause safety and environmental issues [10]. In the case of spills or leaks from remote storage facilities or pipelines, some time may pass between the initial release and commencement of the process of collecting measurements to determine the initial

release point.

This work assumes that the linear-stochastic diffusion model is known, the source is localized (e.g., a contaminant is leaking from a single rupture in a pipeline), and that the measurements may or may not be collected from the time of the initial release. Although this work focuses on the reconstruction of the initial state at the time of the first measurement, if measurements begin after some time has passed, running the system backward to the point of a single concentration can yield release location. The discrete diffusion models are in the class described in section 1.1.

1.4 Observability Basics

Two important concepts in modern control theory, introduced in the 1960s by Kalman [11, 12, 13, 14], are controllability and observability. Consider a discrete-time linear dynamical system with numerous inputs and outputs. This system is modeled by the following equations:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \tag{1.1}$$

where $k \geq 0$ denotes the discrete time step, $x_k \in \mathbb{R}^N$ represents the state vector, the system dynamics are given by the matrix $A \in \mathbb{R}^{N \times N}$, the control matrix (or input matrix) is given by $B \in \mathbb{R}^{N \times P}$, and the control vector (or input vector) is represented by $u_k \in \mathbb{R}^P$. Linear sensors are modeled by the matrix $C \in \mathbb{R}^{M \times N}$, and the measurements are given by $y_k \in \mathbb{R}^M$.

These two concepts answer significant questions regarding the state vector. Observability asks [15]: After measurements are taken over a finite time interval, can the state vector x_k be determined? Controllability asks [15]: Given an initial state vector x_0, is there a sequence of control vectors u_k which can transform x_0 into a desired x_k within a finite time? Observability describes the relationship between the state

vector and the measurements and is characterized by the matrices A and C. Controllability describes the relationship between the state vectors and control vectors and is characterized by the matrices A and B. The system in this work is uncontrolled diffusion; the control vector is zero for all k (or equivalently $B = 0$). Therefore, the remainder of this section will examine the relationship between the state vector and the measurements.

Definition 1.4.1. [16] The system (1.1) is said to be observable if, for any unknown initial state x_0, there exists a finite time K such that knowledge of the inputs u_k and the measurements y_k for $k = 0, \ldots, K$ suffices to uniquely determine the initial state x_0. Otherwise, the system is unobservable.

Consider the following uncontrolled discrete-time linear system:

$$\begin{aligned} x_{k+1} &= Ax_k \\ y_k &= Cx_k \end{aligned} \tag{1.2}$$

where the state vector is $x_k \in \mathbb{R}^N$, the system matrix is $A \in \mathbb{R}^{N\times N}$, the measurement matrix is $C \in \mathbb{R}^{M\times N}$, and the measurements are $y_k \in \mathbb{R}^M$. It can be shown that any state can be realized with knowledge of the system dynamics A and initial state x_0. For example, the following states are written in terms of x_0.

$$\begin{aligned} x_1 &= Ax_0 \\ x_2 &= Ax_1 = A(Ax_0) \ = A^2x_0 \\ x_3 &= Ax_2 = A(A^2x_0) = A^3x_0 \\ &\vdots \end{aligned}$$

This leads to a system expression equivalent to (1.2):

$$x_k = A^k x_0 \tag{1.3a}$$

$$y_k = CA^k x_0 \tag{1.3b}$$

The observability matrix Φ comes from the measurements expression (1.3b):

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} CA^0 \\ CA^1 \\ \vdots \\ CA^{n-1} \end{bmatrix} x_0.$$

Definition 1.4.2. [13] The system observability matrix $\Phi \in \mathbb{R}^{MN \times N}$ is defined by

$$\Phi = \begin{bmatrix} CA^0 \\ CA^1 \\ \vdots \\ CA^{n-1} \end{bmatrix}. \tag{1.4}$$

Theorem 1. [17] A linear discrete-time system is completely observable if and only if the observability matrix Φ has full rank.

The measurements vector $y = [y_0 \dots y_{n-1}]^T$ in (1.3b) is a linear combination of the columns of Φ, meaning that the measurements lie in the column space of Φ, which is expressed as

$$y = \Phi x_0.$$

In addition to lying in the column space, in order for a unique solution to exist, the dimension of the null space should be identically zero. Therefore, following Theorem 1, if the observability matrix Φ is full rank, then x_0 can be uniquely obtained by

$$x_0 = (\Phi^T \Phi)^{-1} \Phi^T y. \tag{1.5}$$

Theorem 2. [17] The discrete-time observability Gramian is defined as

$$\tilde{W} = \sum_{m=0}^{\infty} (A^T)^m C^T C A^m. \tag{1.6}$$

The discrete-time observability Gramian $W \in \mathbb{R}^{N \times N}$ for $k = 0, \dots, N-1$ equates to

$$W = \Phi^T \Phi. \tag{1.7}$$

1.5 Diffusion Matrix

The system model used in this thesis is a diffusive system. In one dimension and two dimensions, this system takes the form of a discrete-time heat equation. The following sections describe the realization of these systems as linear difference equations in the form (1.2).

1.5.1 Diffusion of a One-Dimensional System

Consider the following boundary value problem where the boundary conditions are given for two specific locations [18]:

$$u''(x) = f(x) \quad \text{for} \quad 0 < x < 1, \tag{1.8}$$

with Dirichlet boundary conditions $u(0) = 0$ and $u(1) = 0$. The finite difference method may be used to approximate the solution. The grid points are equally spaced by the distance h, where $h = 1/(N+1)$. The solution consists of points on the grid with values U_0, U_1, ..., U_{N+1}. The central difference approximation has the expression

$$\frac{1}{h^2}(U_{j-1} - 2U_j + U_{j+1}) = f(x_j) \quad \text{for} \quad j = 1, 2, ..., N, \tag{1.9}$$

where U_0 and U_{N+1} are excluded because these boundary values are known. The unknown values are expressed in the form of a set of linear equations $D_1 b = f$ where $b = [U_1, U_2, \ldots, U_N]^T$ and $f = [f(x_1), f(x_2), \ldots, f(x_N)]^T$. The second-order difference N x N matrix D_1 is written as

$$D_1 = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & 1 & -2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{bmatrix} \tag{1.10}$$

with the following eigenvalues and eigenvectors, respectively [18]:

$$\lambda_p = \frac{2}{h^2}(\cos(p\pi h) - 1) \quad \text{for} \quad p = 1, 2, \ldots, N, \tag{1.11a}$$

$$u_p = [\sin(p\pi h) \cdots \sin(p\pi N h)] \quad \text{for} \quad p = 1, 2, \ldots, N. \tag{1.11b}$$

This linear system is used to model the one-dimensional diffusion system. Consider the following diffusion equation:

$$u_t = \alpha u_{xx} \quad \text{for} \quad 0 \leq x \leq 1, \tag{1.12}$$

with Dirichlet boundary conditions $u(0) = 0$ and $u(1) = 0$, where α represents the diffusion coefficient with units $(length^2/time)$. The forward finite-difference method is expressed as

$$\frac{U_j^{k+1} - U_j^k}{\Delta t} = \alpha \frac{U_{j-1}^k - 2U_j^k + U_{j+1}^k}{h^2} \tag{1.13}$$

The superscript indicates the discrete time step. Solving for the unknown U_j^{k+1}, the equation gives

$$U_j^{k+1} = (1 - 2\gamma)U_j^k + \gamma U_{j+1}^k + \gamma U_{j+1}^k \tag{1.14}$$

where $\gamma = \alpha \frac{\Delta t}{h^2}$. The system matrix A is

$$A = \begin{bmatrix} 1-2\gamma & \gamma & 0 & \cdots & 0 \\ \gamma & 1-2\gamma & \gamma & \ddots & \vdots \\ 0 & \gamma & 1-2\gamma & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \gamma \\ 0 & \cdots & 0 & \gamma & 1-2\gamma \end{bmatrix}. \tag{1.15}$$

In order for the system A to be stable, $\Delta t = rh^2$ where $\alpha r < \frac{1}{2}$. The matrix A written in terms of the matrix D_1 (1.10) is expressed as [6]

$$A = I + \alpha \Delta t D_1. \tag{1.16}$$

Its eigenvalues are

$$\begin{aligned}\lambda_p &= I + \alpha\Delta t(\frac{2}{h^2}(\cos(p\pi h) - 1)) \\ &= 1 + 2\gamma(\cos(p\pi h) - 1) \quad \text{for} \quad p = 1, 2, \ldots, N,\end{aligned} \tag{1.17}$$

and the eigenvectors are equivalent to u_p in (1.11b).

This discrete-time linear dynamical system (1.2) has the state vector

$$x_k = [U_1^k, U_2^k, \ldots, U_N^k]^T.$$

1.5.2 Diffusion of a Two-Dimensional System

The discrete-time linear dynamical system for the two-dimensional case is found in a manner similar to the one-dimensional system. Consider the following second-order partial differential equation:

$$u_t = \alpha u_{xx} + \alpha u_{yy} \quad \text{for} \quad 0 \leq x \leq 1, 0 \leq y \leq 1$$

with Dirichlet boundary conditions $u(x, 0) = 0$ and $u(0, y) = 0$. The solution may be approximated using a finite difference in the form of a $(N + 2) \times (N + 2)$ grid with the equation

$$\alpha\frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j}}{(\Delta x)^2} + \alpha\frac{U_{i,j-1} - 2U_{i,j} + U_{i,j+1}}{(\Delta y)^2} = f(x_i, y_j)$$

where $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, N$ because there are $N \times N$ unknowns. Figure 1.1 illustrates the two-dimensional grid. With equally distanced grid points $\Delta x = \Delta y = h = 1/(N + 1)$, equation (1.5.2) yields

$$\alpha\frac{1}{h^2}(U_{i+1,j} + U_{i,j+1} - 4U_{ij} + U_{i-1,j} + U_{i,j-1}) = f(x_i, y_j).$$

This can be written as $D_2 b = f$, where $b = [U_{1,1}, U_{2,1}, \ldots, U_{N,1}, U_{1,2}, U_{2,2}, \ldots, U_{N,N}]^T$.

The matrix D_2 is $N^2 \times N^2$:

$$D_2 = \frac{\alpha}{h^2} \begin{bmatrix} G & I & 0 & \cdots & 0 \\ I & G & I & \ddots & \vdots \\ 0 & I & G & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & I \\ 0 & \cdots & 0 & I & G \end{bmatrix}$$

where the $N \times N$ matrix G has the form

$$G = \begin{bmatrix} -4 & 1 & 0 & \cdots & 0 \\ 1 & -4 & 1 & \ddots & \vdots \\ 0 & 1 & -4 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & 1 & -4 \end{bmatrix}$$

and I is the $N \times N$ identity matrix.

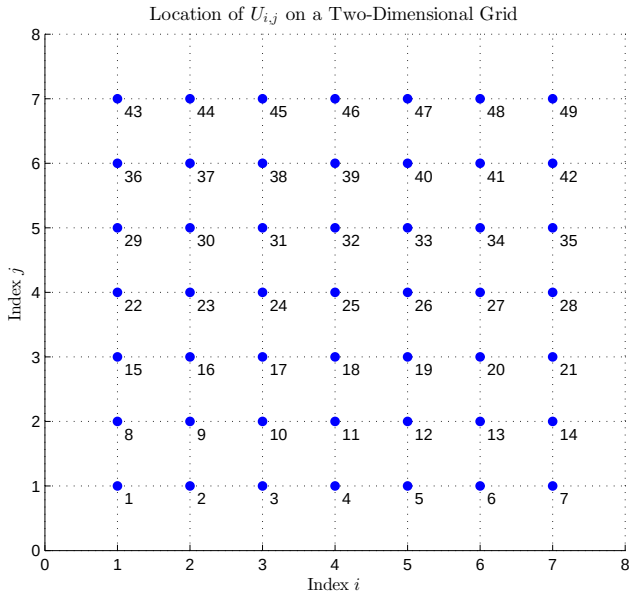

Figure 1.1: The figure illustrates the placement of $U_{i,j}$ on a two-dimensional grid where $N = 7$. The label on each point represents the position in the vector b.

The eigenvalues of D_2 with $\alpha = 1$ are [18]

$$\lambda_{p,q} = \frac{2}{h^2}((\cos(p\pi h) - 1) + (\cos(q\pi h) - 1)) \tag{1.18}$$

for $p = 1, 2, \ldots, N$ and $q = 1, 2, \ldots, N$ where the total p, q combinations are N^2. The corresponding $N^2 \times 1$ eigenvector is [18]

$$u_{p,q}^{i,j} = \sin(p\pi ih)\sin(q\pi jh). \tag{1.19}$$

Solving for the unknown, $U_{i,j}^{k+1}$, similar to equation (1.14), the $N^2 \times N^2$ system matrix A can be written as

$$A = \begin{bmatrix} B & \gamma I & 0 & \cdots & 0 \\ \gamma I & B & \gamma I & \ddots & \vdots \\ 0 & \gamma I & B & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \gamma I \\ 0 & \cdots & 0 & \gamma I & B \end{bmatrix} \tag{1.20}$$

where B is the $N \times N$ matrix:

$$B = \begin{bmatrix} 1-4\gamma & \gamma & 0 & \cdots & 0 \\ \gamma & 1-4\gamma & \gamma & \ddots & \vdots \\ 0 & \gamma & 1-4\gamma & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \gamma \\ 0 & \cdots & 0 & \gamma & 1-4\gamma \end{bmatrix}. \tag{1.21}$$

For a stable two-dimensional system, $\alpha r < \frac{1}{4}$. The eigenvalues are

$$\begin{aligned} \lambda_{p,q} &= I + \alpha\Delta t\frac{2}{h^2}((\cos(p\pi h) - 1) + (\cos(q\pi h) - 1)) \\ &= 1 + 2\gamma((\cos(p\pi h) - 1) + (\cos(q\pi h) - 1)) \end{aligned}$$

and the eigenvectors are equivalent to u_p in (1.19). This discrete-time linear dynamical system (1.2) has state vector

$$x_k = [U_{1,1}^k, U_{2,1}^k, \ldots, U_{N,1}^k, U_{1,2}^k, U_{2,2}^k, \ldots, U_{N,N}^k]^T.$$

Chapter 2

BACKGROUND

In the interest of determining the initial state of the system, the observability of the systems introduced in section 1.5. As noted in section 1.4, a system is said to be observable when the initial state can be determined after a finite number of measurements are taken as the system evolves over time. Theorem 1 gives a "yes-no" answer as to whether or not the linear dynamical system is observable. However, it does not specifically state to what degree the system is observable. To address this, different metrics of observability utilizing the observability matrix have been proposed. Equivalently, they can be applied to the observability gramian.

In many cases, the observability matrix or the observability gramian is built from a collection of predetermined sensor configurations. These gramians are then compared to each other against one or more performance metrics and the optimal configuration is selected [19]. Because these sensor configurations are predetermined, they do not clearly specify a method of choosing sensors as the system propagates to optimize one or more observability performance metrics. One method of sensor scheduling that has been proposed uses the condition number of the observability matrix Φ as a performance metric and Rank Revealing QR (RRQR) factorization for the scheduling algorithm [6].

2.1 Relation between the Observability Gramian and Estimation Covariance

A measure is needed to quantify the quality of the initial state estimate in the case of noisy measurements. One method used in optimal sensor selection is based

on the Fisher information matrix (FIM) [20] [21]. The FIM F is associated with a lower bound on the covariance of an unbiased estimate $\hat{x}$ known as the Cramér-Rao bound and given as

$$E[(x - \hat{x})(x - \hat{x})^T] \succcurlyeq F^{-1} \tag{2.1}$$

where $\hat{x}$ is the estimate of x. When equality is achieved, the estimator $\hat{x}$ is said to be efficient, in which case it gives a minimum-variance estimate [20]. In this case, and assuming x is Gaussian, the FIM is the inverse of the estimation covariance.

Consider the linear dynamical discrete-time system in (1.2), but with noise on the measurements:

$$\begin{aligned} x_{k+1} &= Ax_k \\ y_k &= Cx_k + \eta_k \end{aligned} \tag{2.2}$$

where $\hat{y}$ are linear measurements with noise η_k modeled as i.i.d. zero-mean white Gaussian vectors with variance ρ; i.e. $\eta_k \sim \mathcal{N}(0, \rho)$. The estimation uncertainty and the FIM are connected to the observably gramian $\Phi^T\Phi$ through the estimation covariance of the initial condition x_0. The estimation covariance of the initial condition x_0 is

$$E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] = (\rho^{-1}\Phi^T\Phi)^{-1}. \tag{2.3}$$

The least-square estimator is efficient, therefore the Cramér-Rao lower bound equality is achieved [22]. From Theorem 2, this shows equivalency between the discrete-time observability gramian and the FIM F:

$$F = \rho^{-1}\Phi^T\Phi = \rho^{-1}W. \tag{2.4}$$

Sensor selection using the observability gramian W is directly related to the estimation covariance of the initial condition.

2.2 Observability Measure Metrics

Three different metrics were introduced by Müller and Weber [23] for observability of linear systems. The first is

$$\mu_1 = \lambda_{\min}(W), \tag{2.5}$$

i.e., the smallest eigenvalue of the observability gramian W. The smaller μ_1, the less observable the system is for this metric. The eigenvector corresponding to μ_1 is the worst observable direction of the system. The second metric is

$$\mu_2 = \frac{N}{\text{trace}(W^{-1})} \tag{2.6}$$

where N is the number of elements in the $N \times 1$ state vector. Similar to (2.5), the smaller μ_2 is, the less observable the system according to this metric. The third metric is

$$\mu_3 = \det(W)^{\frac{1}{N}} \tag{2.7}$$

The higher μ_3 is, the more observable the system is based on this metric. An application using metrics (2.5), (2.6), and (2.7) is satellite attitude control [23].

Dochain *et al.* [4] utilized the condition number of the observability matrix Φ as a measure to determine sensor location:

$$\kappa(W) = \frac{\sigma_{\max}(W)}{\sigma_{\min}(W)} \tag{2.8}$$

where $\sigma_{\max}$ represents the largest singular value and $\sigma_{\min}$ represents the minimum singular value. The condition number gives insight as to how a perturbation in the input, the initial state vector, perturbs the output, the measurements. A smaller condition number indicates a higher degree of observablity. It also measures the

sensitivity in taking the inverse of the matrix. An application using metric (2.8) is in determining optimal sensor locations for fixed bed bioreactors [4].

Waldraff *et al.* [24] used the smallest singular value as a metric:

$$NS(W) = \sigma_{\min}(W). \tag{2.9}$$

This metric gives information about the weight of the least observable mode. An application using metric (2.9) is determining optimal sensor locations for a tubular reactor [24].

Van den Berg [25] proposed two methods which put focus is on the weight of the most observable mode. The first is the spectral radius

$$\rho(W) = \sigma_{\max}(W), \tag{2.10}$$

which gives information about the most observable mode. The second metric is

$$\text{trace}(W) = \sum_{i=1}^{N} \sigma_i(W), \tag{2.11}$$

where a larger metric indicates a higher degree of observability. It also gives information about the average degree of uncertainty in the estimate. These metrics ((2.10) and (2.11)) have also been used for sensor placement a in tubular reactor [25].

Another metric, called a Figure of Merit (FOM), is a weighted sum of metrics from the FIM: condition number, the trace, and the determinant of the observability gramian [26]. The condition number gives information about the sensitivity of taking the inverse; the trace gives the overall sensitivity; and the determinant gives the global estimation uncertainty [26]. The metric is given as:

$$FOM = -\beta_1 \log(\kappa(W)) + \beta_2 \log(\text{trace}(W)) + \beta_3 \log(\det(W)), \tag{2.12}$$

where β_i accounts for the normalization and weight of each term. A high metric indicates a favorable sensor configuration. An application is sensor placement for engine health monitoring [26].

Chapter 3

MATHEMATICAL FORMULATION

Three algorithms are described in this chapter, one of which is in literature. The first algorithm uses the condition number of the observability matrix $\kappa(\Phi)$ as a metric. The second algorithm uses the trace of the inverse of observability gramian trace(W^{-1}) as the metric. The third algorithm is one described in [8], which also uses trace(W^{-1}) as the metric and will be compared with the second algorithm in Chapter 4.

3.1 Sensor Scheduling Algorithm for Deterministic Measurements

The measure of observability used for this method is the condition number of the observability matrix $\kappa(\Phi)$ (2.8), which provides information about the computational tractability of performing the inverse of the observability matrix. To achieve the highest degree of linear independence, the N columns of the observability matrix should be orthogonal. The condition number of an orthonormal $N \times N$ real matrix is 1 [27], which in this case, is the optimal value of this metric of observability.

In order to construct Φ with columns which are as orthogonal as possible, the normalized Gram determinant will be utilized in formulating this algorithm. The equation of a diffusive system is characterized as in (1.2), but now $C_m \in \mathbb{R}^{1\times N}$ is chosen from a library $\mathcal{C} = \{C_1, \ldots, C_N\}$ where C_m has a one as its m^{th} element and is otherwise zero:

$$\begin{aligned} x_{k+1} &= Ax_k \\ y_k &= C_{m_k}x_k + \eta_k \end{aligned} \tag{3.1}$$

Physically, selecting C_m corresponds to taking a measurement at one point corre-

sponding to the position of the m^{th} sensor. The $N \times N$ observability matrix for this system is

$$\Phi = \begin{bmatrix} - & C_{m_0}A^0 & - \\ & \vdots & \\ - & C_{m_{n-1}}A^{n-1} & - \end{bmatrix}. \tag{3.2}$$

The Gram matrix [28] of the observability matrix (3.2) yields a matrix of inner products given as

$$\begin{aligned} G &= \Phi\Phi^T \\ &= \begin{bmatrix} \langle C_{m_0}A^0, C_{m_0}A^0\rangle & \langle C_{m_0}, C_{m_1}A^1\rangle & \dots & \langle C_{m_0}, C_{m_{n-1}}A^{n-1}\rangle \\ \vdots & \ddots & & \vdots \\ \langle C_{m_{n-1}}A^{n-1}, C_{m_0}A^0\rangle & \langle C_{m_{n-1}}A^{n-1}, C_{m_1}A^1\rangle & \dots & \langle C_{m_{n-1}}A^{n-1}, C_{m_{n-1}}A^{n-1}\rangle \end{bmatrix}, \end{aligned} \tag{3.3}$$

where the Gram matrix is $G \in \mathbb{R}^{N\times N}$. To emphasize the angles between the vectors, each vector is normalized to unit length; the normalized Gram matrix is given as

$$\tilde{G} = \begin{bmatrix} 1 & \left\langle \frac{C_{m_0}A^0}{||C_{m_0}A^0||}, \frac{C_{m_1}A^1}{||C_{m_1}A^1||} \right\rangle & \dots & \left\langle \frac{C_{m_0}A^0}{||C_{m_0}A^0||}, \frac{C_{m_{n-1}}A^{n-1}}{||C_{m_{n-1}}A^{n-1}||} \right\rangle \\ \vdots & \ddots & & \vdots \\ \left\langle \frac{C_{m_{n-1}}A^{n-1}}{||C_{m_{n-1}}A^{n-1}||}, \frac{C_{m_0}A^0}{||C_{m_0}A^0||} \right\rangle & \left\langle \frac{C_{m_{n-1}}A^{n-1}}{||C_{m_{n-1}}A^{n-1}||}, \frac{C_{m_1}A^1}{||C_{m_1}A^1||} \right\rangle & \dots & 1 \end{bmatrix}.$$

Using the bilinearity of the inner product, this becomes

$$\tilde{G} = \frac{\begin{bmatrix} \langle C_{m_0}A^0, C_{m_0}A^0\rangle & \langle C_{m_0}A^0, C_{m_1}A^1\rangle & \dots & \langle C_{m_0}A^0, C_{m_{n-1}}A^{n-1}\rangle \\ \vdots & \ddots & & \vdots \\ \langle C_{m_{n-1}}A^{n-1}, C_{m_0}A^0\rangle & \langle C_{m_{n-1}}A^{n-1}, C_{m_1}A^1\rangle & \dots & \langle C_{m_{n-1}}A^{n-1}, C_{m_{n-1}}A^{n-1}\rangle \end{bmatrix}}{||C_{m_0}A^0||^2 \quad \cdots \quad ||C_{m_{n-1}}A^{n-1}||^2}. \tag{3.4}$$

The Gram determinant is represented as $|G|$, which satisfies [29]

$$0 \quad \leq \quad |G| \quad \leq \quad \prod_{i=1}^{N} ||C_{m_{i-1}}A^{i-1}||^2 \tag{3.5}$$

The upper and lower bounds of the normalized Gram determinant equate to

$$0 \quad \leq \quad |\tilde{G}| \quad \leq 1$$

where zero signifies linear dependency and one indicates orthonormal vectors. With this knowledge, a greedy algorithm [30] is used to formulate a sensor schedule. The

algorithm begins by randomly selecting the sensor C_{m_0} for the initial time step $k = 0$. For the next time step q, where $q \leq N - 1$, a greedy algorithm tests each sensor C_{m_q} in the library. For each sensor, the normalized Gram determinant is built only from the row vectors of Φ with the time steps $k = 0, \ldots, q$. The sub-matrix of Φ for time steps $k = 0, \ldots, q$ is

$$\hat{\Phi} = \begin{bmatrix} - & C_{m_0}A^0 & - \\ & \vdots & \\ - & C_{m_q}A^q & - \end{bmatrix} \tag{3.6}$$

and the normalized Gram determinant is formalized as

$$\hat{G} = \frac{\hat{\Phi}\hat{\Phi}^T}{||C_{m_0}A^0||^2 \quad \cdots \quad ||C_{m_q}A^q||^2}. \tag{3.7}$$

The sensor C_{m_q} which gives the maximum determinant among all N normalized Gram determinants is assigned to that time step. This process is repeated until sensors have been selected for all the time steps $k = 0, \ldots, N - 1$. This schedule is then utilized to collect measurements and subsequently find the initial state x_0.

Algorithm 1 Sensor Scheduling Algorithm based on the Gram determinant

i Define the one-dimensional (1.15) or two-dimensional (1.20) system diffusion matrix A in (1.2).

ii Let $k = 0, 1, \ldots, N-1$ be the time steps and q represent the current time step for which the sensor is being selected.

iii From the library $\mathcal{C} = \{C_1, \ldots, C_N\}$, randomly choose the first sensor location C_{m_0}, to build the first row vector $C_{m_0}A^0$ of Φ. Note: When a sensor is selected from the library in any step of this algorithm, the library is replenished.

iv Let $\hat{\Phi}$ (3.6) represent the sub-matrix of the Φ matrix (3.2) which includes the row vectors for time steps $k = 1, \ldots, q$.

v Sequentially choose every C_m in the library to first compute the row vector $C_{m_q}A^q$ and then compute the normalized Gram determinant $\hat{G}$ (3.7) for the corresponding $\hat{\Phi}$ matrix.

vi From the N normalized Gram determinants, choose the sensor C_m, which yields the highest result.

vii If $q \neq N-1$, repeat steps v. - vi. to select a sensor for the next q time step.

3.2 Sensor Scheduling Algorithm for Noisy Measurements

3.2.1 Method based on the Pseudoinverse

The measure of observability used for this method is the trace of the inverse of observability gramian, trace(W^{-1}), which gives information about the average degree of uncertainty in the estimate. To minimize the degree of uncertainty of the initial state x_0, the sensor schedule focuses on minimizing the estimation covariance of the

initial state. The estimation covariance of the initial state x_0 for the system described in (2.2) is given as

$$\begin{aligned} P_{x_0} &= \rho(\Phi^{-1})(\Phi^{-1})^T \\ &= \rho(\Phi^T\Phi)^{-1} \end{aligned} \tag{3.8}$$

where Φ is the $N \times N$ observability matrix. For time steps $k = 0, \ldots, q$ where $q \leq N-1$, the Φ matrix is $(q+1) \times N$ expressed as $\hat{\Phi}$ in (3.6). In this case, the inverse of $\hat{\Phi}$ is expressed in terms of the right pseudoinverse [31]

$$H = \hat{\Phi}^T(\hat{\Phi}\hat{\Phi}^T)^{-1}. \tag{3.9}$$

The estimation covariance is now expressed as

$$\hat{P}_{x_0} = \rho H H^T. \tag{3.10}$$

In this algorithm, to avoid the inversion of $\hat{\Phi}\hat{\Phi}^T$ when the matrix is ill-conditioned, the Singular Value Decomposition (SVD) is used. Let the SVD of $\hat{\Phi}$ be

$$\hat{\Phi} = \begin{bmatrix} U \end{bmatrix}_{N\times N} \begin{bmatrix} \Sigma_N & 0 \end{bmatrix}_{N\times M} \begin{bmatrix} & V^T & \\ & & \\ & & \end{bmatrix}_{M\times M} \tag{3.11}$$

where U is an $N \times N$ unitary matrix, V^T is an $M \times M$ unitary matrix, and Σ_N is a diagonal matrix with the singular values of $\hat{\Phi}$ along the diagonal. The SVD of H is

formulated as

$$
\begin{aligned}
H &= (V \begin{bmatrix} \Sigma_N \\ 0 \end{bmatrix} U^T)(U \begin{bmatrix} \Sigma_N & 0 \end{bmatrix} V^T V \begin{bmatrix} \Sigma_N \\ 0 \end{bmatrix} U^T)^{-1} \\
&= (V \begin{bmatrix} \Sigma_N \\ 0 \end{bmatrix} U^T)(U \Sigma_N^{-2} U^T) \\
&= V \begin{bmatrix} \Sigma_N \\ 0 \end{bmatrix} \Sigma_N^{-2} U^T \\
&= V \begin{bmatrix} \Sigma_N^{-1} \\ 0 \end{bmatrix} U^T.
\end{aligned}
\tag{3.12}
$$

The SVD of the estimate covariance $\hat{\Phi}$ can be expressed in terms of the SVD of the right pseudoinverse H as

$$
\begin{aligned}
HH^T &= (V \begin{bmatrix} \Sigma_N^{-1} \\ 0 \end{bmatrix} U^T)(U \begin{bmatrix} \Sigma_N^{-1} & 0 \end{bmatrix} V^T) \\
&= V \begin{bmatrix} \Sigma_N^{-2} & 0 \\ 0 & 0 \end{bmatrix} V^T.
\end{aligned}
\tag{3.13}
$$

This relation shows that Σ_N can be found directly from $\hat{\Phi}$ and by minimizing the trace of Σ_N^{-2}, the trace of the the right pseudoinverse H is also minimized. Therefore, H does not need to be explicitly evaluated, meaning the inversion of $\hat{\Phi}\hat{\Phi}^T$ is not solved. In this formulation, the sensor C_{m_0} for the initial time step $k = 0$ is arbitrarily chosen. For the next time step q, a greedy algorithm tries each sensor C_{m_q} in the library by building $\hat{\Phi}$ (3.6). For each $\hat{\Phi}$ the trace of its SVD is found as

$$
\text{trace}(\Sigma_{q+1}^{-2}) = \sum_{i=1}^{q+1} \frac{1}{\sigma_i^2}. \tag{3.14}
$$

The sensor C_{m_q} which yields the minimum trace is assigned to that time step. This process is repeated until sensors have been selected for all the time steps $k =$

$0, \ldots, N-1$. This schedule is then utilized to collect measurements and subsequently estimate the initial state $\hat{x}_0$.

Algorithm 2 Sensor Scheduling Algorithm based on the trace of the observability matrix

i Define the one-dimensional (1.15) or two-dimensional (1.20) system diffusion matrix A in (1.2).

ii Let $k = 0, 1, \ldots, N-1$ be the time steps and q represent the current time step for which the sensor is being selected.

iii From the library $\mathcal{C} = \{C_1, \ldots, C_N\}$, randomly choose the first sensor location C_{m_0}, to build the first row vector $C_{m_0}A^0$ of Φ. Note: When a sensor is selected from the library in any step of this algorithm, the library is replenished.

iv Let $\hat{\Phi}$ (3.6) represent the sub-matrix of the Φ matrix (3.2) which includes the row vectors for time steps $k = 1, \ldots, q$.

v Sequentially choose every C_m in the library to first compute the row vector $C_{m_q}A^q$ and then compute the SVD for the $\hat{\Phi}$ matrix to find its trace using equation (3.14).

vi From the N computed traces, choose the sensor C_{m_q}, which yields the smallest trace.

vii If $q \neq N-1$, repeat steps v. - vi. to select a sensor for the next q time step.

3.2.2 Method based on the Kalman Filter

Similar to Algorithm 2, the measure used for this method is the trace of the inverse of observability gramian, trace(W^{-1}). This method utilizes the sensor scheduling

algorithm described in [8] to determine the initial state estimate $\hat{x}_0$. For the linear dynamical discrete-time system in (3.1), the Kalman filter gives the pre-measurement error covariance $\tilde{P}_k$, the post-measurement error covariance $\breve{P}_k$, and the Kalman gain matrix K as [32]

$$\tilde{P}_k = A\breve{P}_{k-1}A^T \tag{3.15a}$$

$$\breve{P}_k = [I - K_k C_{m_k}]\tilde{P}_k \tag{3.15b}$$

$$K = \tilde{P}_k C_{m_k}^T [C_{m_k}\tilde{P}_k C_{m_k}^T + \rho]^{-1}. \tag{3.15c}$$

In the method described in [8], through the means of a greedy algorithm, the sensor at each time step is selected from the library $\mathcal{C}$ which minimizes the trace of the post-measurement error covariance $\breve{P}_k$. This schedule is then utilized to find the estimate of the initial state $\hat{x}_0$. There is no noise on the dynamical system, only on the measurements. In the Kalman filter, an initial $\breve{P}_0 = \nu I$ needs to be specified. With a deterministic system, $\nu = 0$. In order to use this algorithm, a non-zero ν should be defined. Different variances of ν can be tested to find the value which best optimizes the algorithm for the given set of parameters of the system A. In general, it was observed that a larger ν should be chosen for a diffusion system that is closer to being unstable.

Algorithm 3 Sensor Scheduling Algorithm described in [8]

i Define the one-dimensional (1.15) or two-dimensional (1.20) system diffusion matrix A in (1.2).

ii Let $k = 0, 1, \ldots, N-1$ be the time steps and q represent the current time step for which the sensor is being selected.

iii As the system propagates from time steps $k = 0, 1, \ldots, N-1$, for each time step, sequentially test every C_m from the library $\mathcal{C} = \{C_1, \ldots, C_N\}$ to see which C_{m_q} minimizes the trace of the post-measurement error covariance $\breve{P}_q$.

Chapter 4

RESULTS

In both the deterministic and noisy measurements cases, the results are presented in a similar manner. Two different γ parameters are considered, $\gamma = 0.04$ and $\gamma = 0.004$, for Algorithms 1 and 2 to illustrate the effects on the metric as the size N increases in the one-dimensional case. Algorithms 1 and 2 are then used to show the reconstruction of the initial state $\hat{x}_0$ using $\gamma = 0.04$ for both the one-dimensional and two-dimensional systems, with $N = 25$ and $N = 49$, respectively. Algorithms 2 and 3 are compared for different variances for the one-dimensional system ($N = 25$) and for the two-dimensional system ($N = 49$). The parameters $\gamma = 0.04$ and $\gamma = 0.004$ are used in this comparison.

4.1 Simulation Parameters

In this simulation, two different γ parameters are considered, $\gamma = 0.04$ and $\gamma = 0.004$. The parameter γ is determined by the diffusion rate of α and the ratio r. The diffusion rate of $\alpha = 0.135$ will be utilized, which models the diffusion rate of ethanol in air is $\alpha = 0.135(cm/sec^2)$ [33]. The two different r values which are as follows: $r_1 = \frac{8}{27}$ and $r_2 = \frac{4}{135}$. This results in γ values of $\gamma = 0.04$ and $\gamma = 0.004$, respectively.

4.2 Deterministic Measurements

Algorithm 1 is used to determine different schedules for different N values, which represent the size of the one-dimensional system matrix A. For each schedule, the same sensor, C_{3_0} is chosen for time step $k = 0$ and the condition number of the observability matrix $\kappa(\Phi)$ is evaluated. Figures 4.1 and 4.2 demonstrate the results

of Algorithm 1 for $\gamma = 0.04$ and $\gamma = 0.004$, respectively.

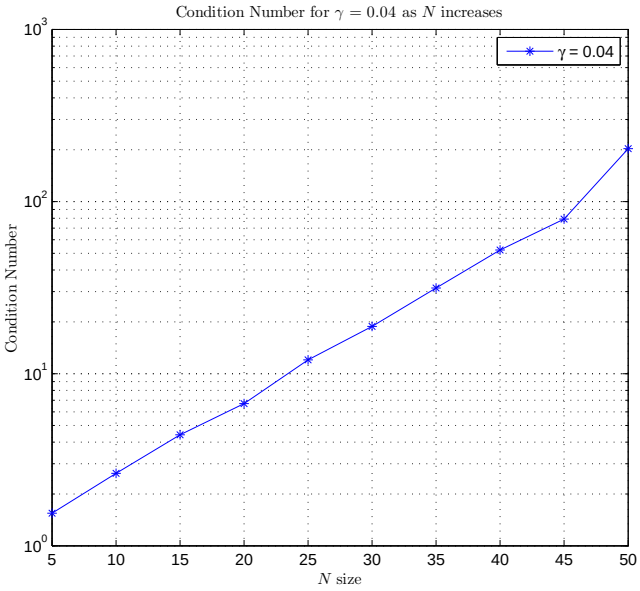

Figure 4.1: Results of Algorithm 1 illustrate the condition number $\kappa(\Phi)$ as a function of the $N \times N$ dimension of the one-dimensional diffusive system A for a $\gamma = 0.04$.

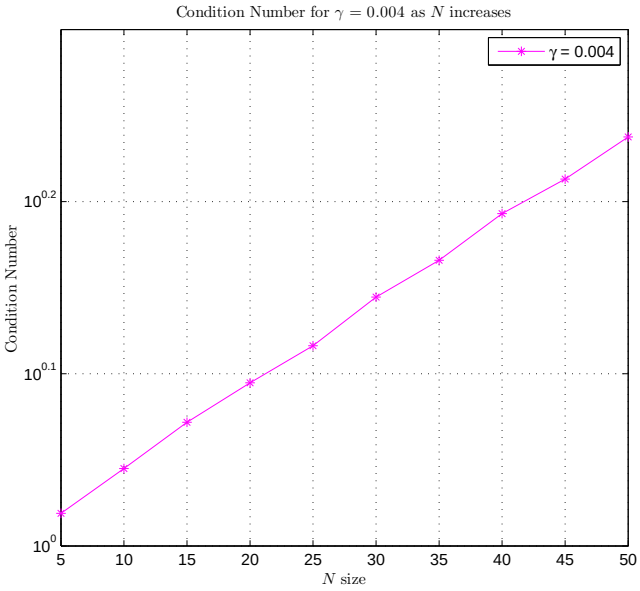

Figure 4.2: Results of Algorithm 1 illustrate the condition number $\kappa(\Phi)$ as a function of the $N \times N$ dimension of the one-dimensional diffusive system A for a $\gamma = 0.004$.

Figures 4.1 and 4.2 both demonstrate a linear logarithmic relationship between

N and the condition number; as N increases, so does the the condition number. The results for $\gamma = 0.004$ yield smaller condition numbers compared to $\gamma = 0.04$. This may be a result of how far each γ is from instability.

4.2.1 One-Dimensional Diffusion System

The following simulation shown in Figures 4.3, 4.4, 4.5, and 4.6 is an example of Algorithm 1 for a one-dimensional diffusive system with $N = 25$ and $\gamma = 0.04$.

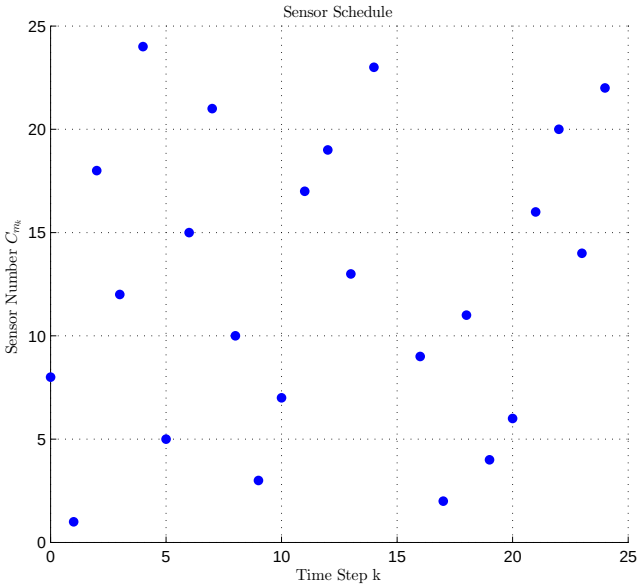

Figure 4.3: Sensor schedule determined by Algorithm 1 for the one-dimensional diffusive system with $\gamma = 0.04$ and $N = 25$. The graph illustrates the sensor chosen as the system propagates for time steps $k = 0, \ldots, 24$.

In the sensor schedule illustrated in Figure 4.3, it can be observed that once the algorithm chooses a sensor, the same sensor is not picked again for any of the remaining time steps.

The Φ matrix is built from the sensor schedule and is used to reconstruct the initial state $\hat{x}_0$ as

$$\hat{x}_0 = \Phi^{-1} y. \tag{4.1}$$

The reconstructed initial state $\hat{x}_0$ is used to solve for $\hat{x}_{24}$ as $\hat{x}_{24} = A^{24}\hat{x}_0$, and $\hat{x}_{12}$ as $\hat{x}_{12} = A^{12}\hat{x}_0$ which is shown in Figure 4.4 and Figure 4.5, respectively.

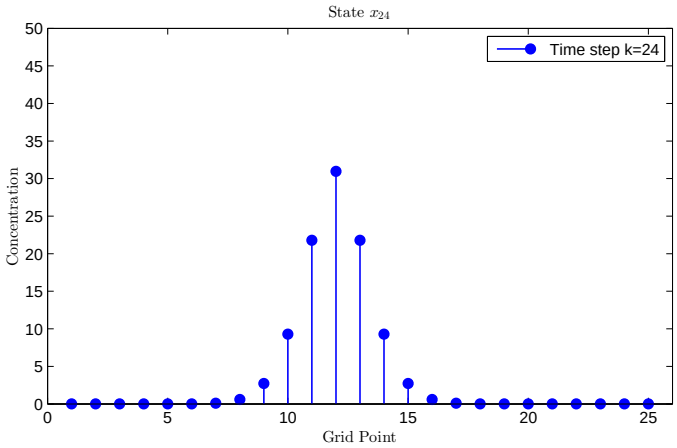

Figure 4.4: Algorithm 1 for the one-dimensional case with $\gamma = 0.04$ and $N = 25$. The graph illustrates the state $\hat{x}_{24}$.

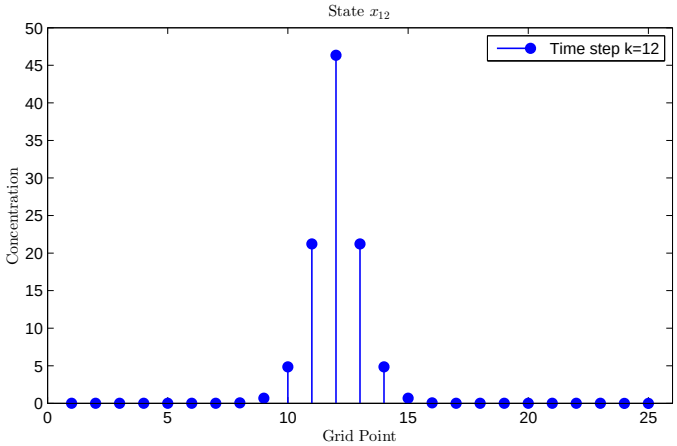

Figure 4.5: Algorithm 1 for the one-dimensional case with $\gamma = 0.04$ and $N = 25$. The graph illustrates the state $\hat{x}_{12}$.

The reconstructed initial state $\hat{x}_0$ is shown in Figure 4.6.

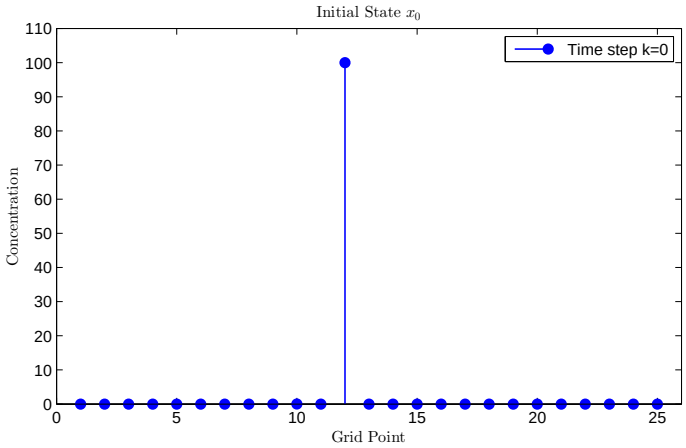

Figure 4.6: Algorithm 1 for the one-dimensional case with $\gamma = 0.04$ and $N = 25$. The graph illustrates the reconstructed initial state $\hat{x}_0$.

Figure 4.6 shows a single release point with the mean-square error (MSE) of 2.10×10^{-28} (essentially machine precision).

4.2.2 Two-Dimensional Diffusion System

The following simulation shown in Figures 4.7, 4.8, 4.9, and 4.10 is an example of Algorithm 1 for a two-dimensional diffusive system with $N = 49$ and $\gamma = 0.04$.

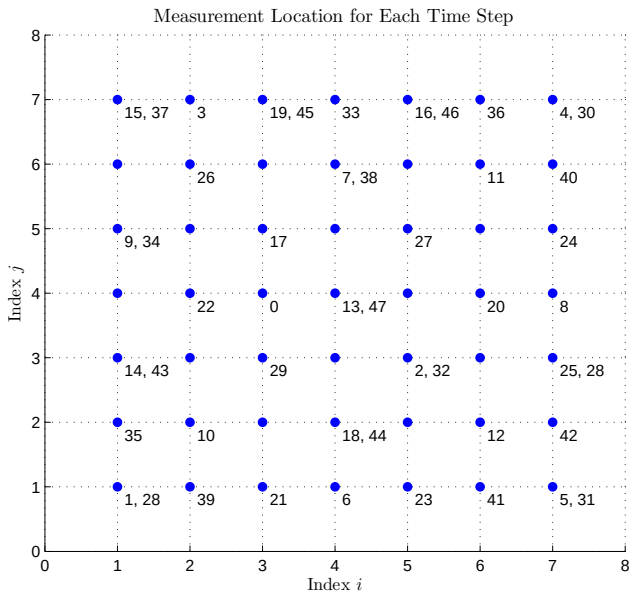

Figure 4.7: Sensor schedule determined by Algorithm 1 for the two-dimensional diffusive system with $\gamma = 0.04$ and $N = 49$. The graph illustrates the time steps at which a sensor was chosen at each grid point corresponding to $U_{i,j}$.

Unlike the results shown in Figure 4.3, it can be observed in Figure 4.7 for these parameters that the use of some sensors are repeated for a different time step and some locations are not measured at any time step.

The reconstructed initial state $\hat{x}_0$, found using equation (4.1), is used to solve for the $\hat{x}_{48}$ as $\hat{x}_{48} = A^{48}\hat{x}_0$, and $\hat{x}_{24}$ as $\hat{x}_{24} = A^{24}\hat{x}_0$ which is shown in Figure 4.8 and Figure 4.9, respectively.

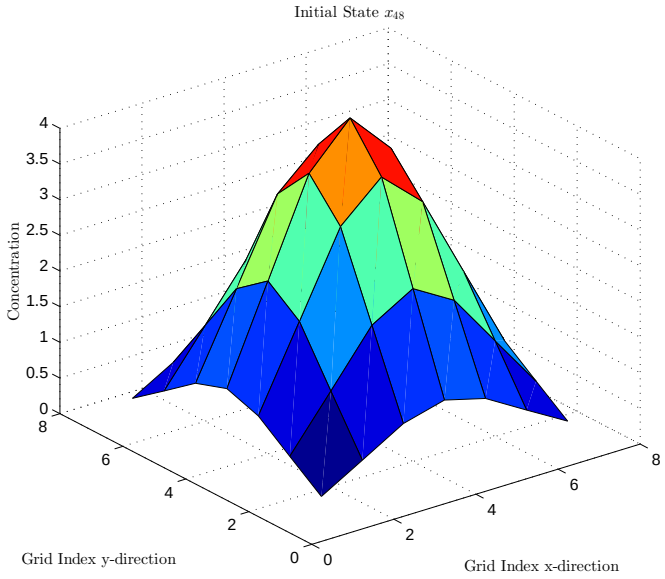

Figure 4.8: Algorithm 1 for the two-dimensional case with $\gamma = 0.04$ and $N = 49$. The graph illustrates the state $\hat{x}_{48}$.

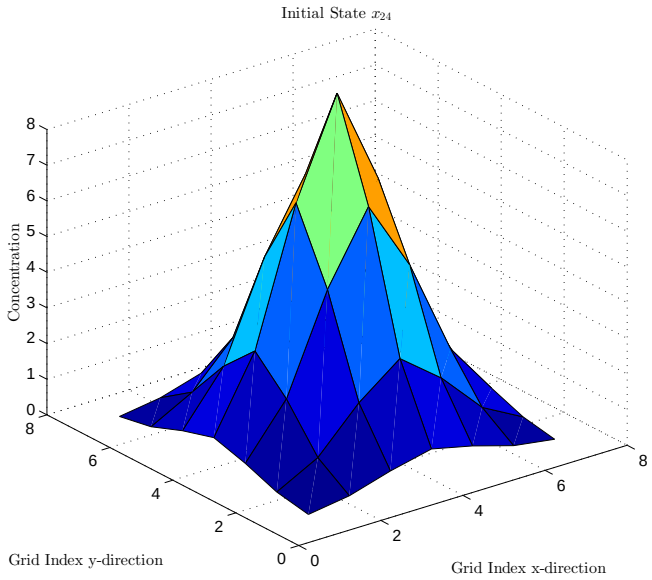

Figure 4.9: Algorithm 1 for the two-dimensional case with $\gamma = 0.04$ and $N = 49$. The graph illustrates the state $\hat{x}_{24}$.

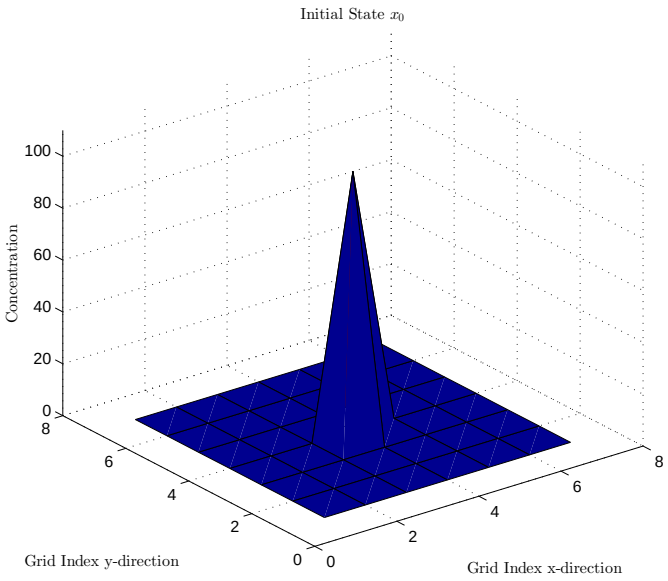

Figure 4.10: Algorithm 1 for the two-dimensional case with $\gamma = 0.04$ and $N = 49$. The graph illustrates the reconstructed initial state $\hat{x}_0$.

Figure 4.10 shows a single release point with a MSE of 1.7×10^{-23}.

4.3 Noisy Measurements

Algorithm 2 is used to determine different schedules for different N values, which represent the size of the one-dimensional system matrix A. For each schedule, the same sensor, C_{5_0} is chosen for time step $k = 0$ and the trace of the inverse of the observability gramian trace$((\Phi^T\Phi)^{-1})$ is evaluated. Figures 4.11 and 4.12 demonstrate the results of Algorithm 1 for $\gamma = 0.04$ and $\gamma = 0.004$, respectively.

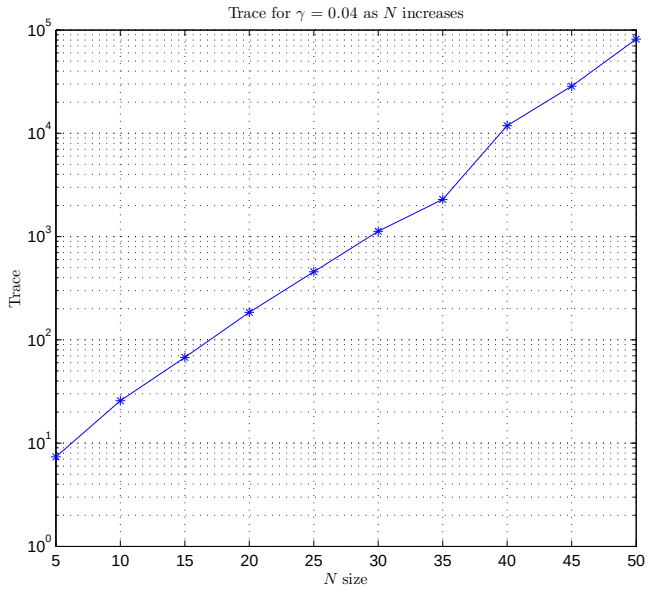

Figure 4.11: Results of Algorithm 2 illustrate the trace of the inverse of the observability gramian trace$((\Phi^T\Phi)^{-1})$ as a function of the $N \times N$ dimension of the one-dimensional diffusive system A for a $\gamma = 0.04$.

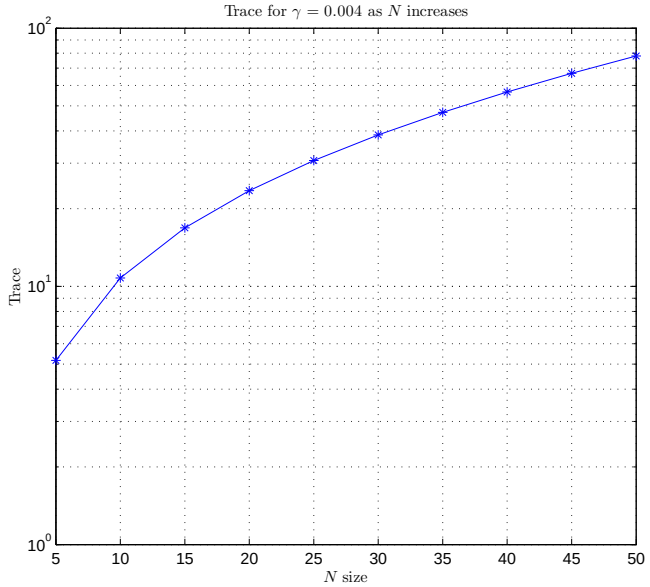

Figure 4.12: Results of Algorithm 2 illustrate the trace of the inverse of the observability gramian trace$((\Phi^T\Phi)^{-1})$ as a function of the $N \times N$ dimension of the one-dimensional diffusive system A for a $\gamma = 0.004$.

Similar to Figures 4.1 and 4.2, Figure 4.11 demonstrates a linear logarithmic

relationship between N and the trace; as N increases, so does the the trace. In Figure 4.12, the slope of the curve seems to decrease as N increases. This may be a result of the ill-conditioning of $(\Phi^T\Phi)^{-1}$ as N increases. As the matrix begins to become ill-conditioned, it will produce larger singular values. The reciprocal of a squared large singular value is small. Adding smaller numbers will decrease the growth of the trace.

4.3.1 One-Dimensional Diffusion System

The following simulation shown in Figures 4.13, 4.14, 4.15, and 4.16 is an example of Algorithm 2 for a one-dimensional diffusive system with $N = 25$ and $\gamma = 0.04$ and noise variance $\rho = 0.1^2$.

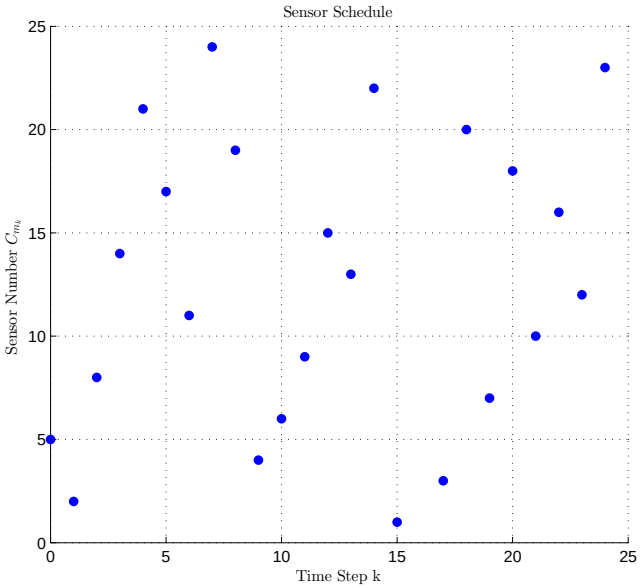

Figure 4.13: Sensor schedule determined by Algorithm 2 for the one-dimensional diffusive system with $\gamma = 0.04$, $N = 25$, and measurement noise variance of 0.1^2. The graph illustrates the sensor chosen as the system propagates for time steps $k = 0, \ldots, 24$.

Similar to the results shown in Figures 4.3 and 4.7, it can be observed in Figure 4.13 for these parameters that once the algorithm chooses a sensor, the same sensor

is not picked again for any of the remainder time steps.

The reconstructed initial state $\hat{x}_0$, found using equation (4.1), is used to solve for the $\hat{x}_{48}$ as $\hat{x}_{24} = A^{24}\hat{x}_0$, and $\hat{x}_{12}$ as $\hat{x}_{12} = A^{12}\hat{x}_0$ which is shown in Figure 4.14 and Figure 4.15, respectively.

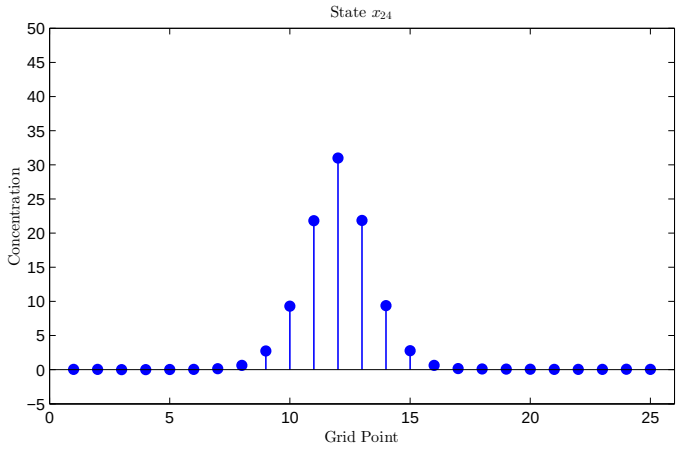

Figure 4.14: Algorithm 2 for the one-dimensional case with $\gamma = 0.04$, $N = 25$, and measurement noise variance of 0.1^2. The graph illustrates the state $\hat{x}_{24}$.

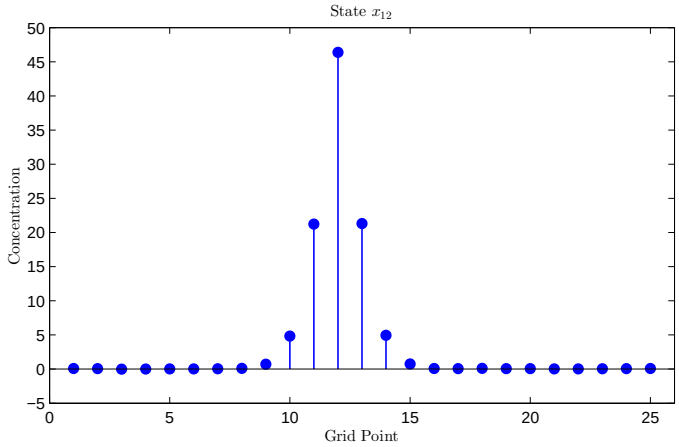

Figure 4.15: Algorithm 2 for the one-dimensional case with $\gamma = 0.04$, $N = 25$, and measurement noise variance of 0.1^2. The graph illustrates the state $\hat{x}_{12}$.

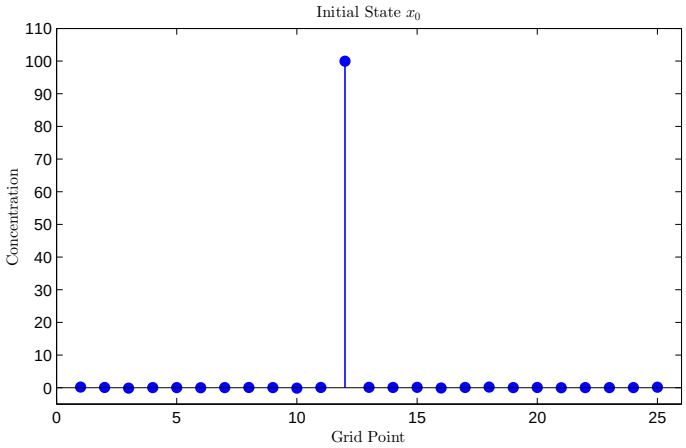

Figure 4.16: Algorithm 2 for the one-dimensional case with $\gamma = 0.04$, $N = 25$, and measurement noise variance of 0.1^2. The graph illustrates the reconstructed initial state $\hat{x}_0$.

Figure 4.16 shows a single release point with a MSE of 1.8×10^{-2}.

4.3.2 Two-Dimensional Diffusion System

The following simulation shown in Figures 4.17, 4.18, 4.19, and 4.20 is an example of Algorithm 2 for a two-dimensional diffusive system with $N = 49$ and $\gamma = 0.04$.

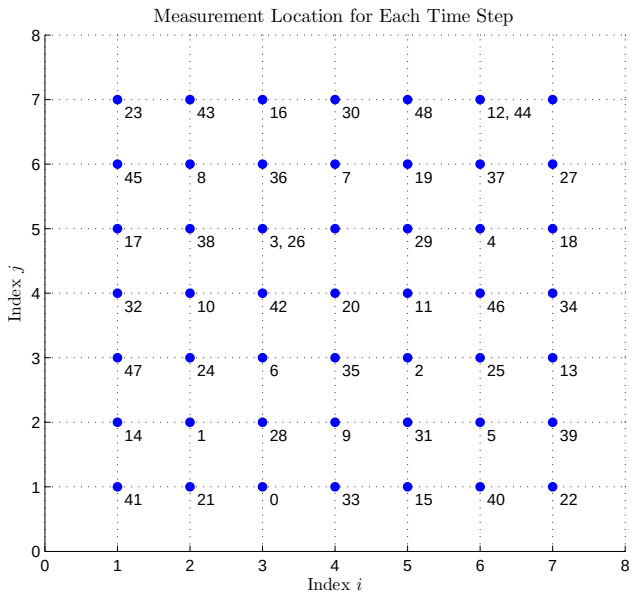

Figure 4.17: Sensor schedule determined by Algorithm 2 for the two-dimensional diffusive system with $\gamma = 0.04$, $N = 49$, and measurement noise variance of 0.01^2. The graph illustrates the time steps at which a sensor was chosen at each grid point corresponding to $U_{i,j}$.

Similar to Figure 4.7, it can be observed in Figure 4.17 for these parameters that the use of some sensors are repeated for a different time step and some locations are not measured at any time step.

The reconstructed initial state $\hat{x}_0$, found using equation (4.1), is used to solve for the $\hat{x}_{48}$ as $\hat{x}_{24} = A^{24}\hat{x}_0$, and $\hat{x}_{12}$ as $\hat{x}_{12} = A^{12}\hat{x}_0$ which is shown in Figure 4.18 and Figure 4.19, respectively.

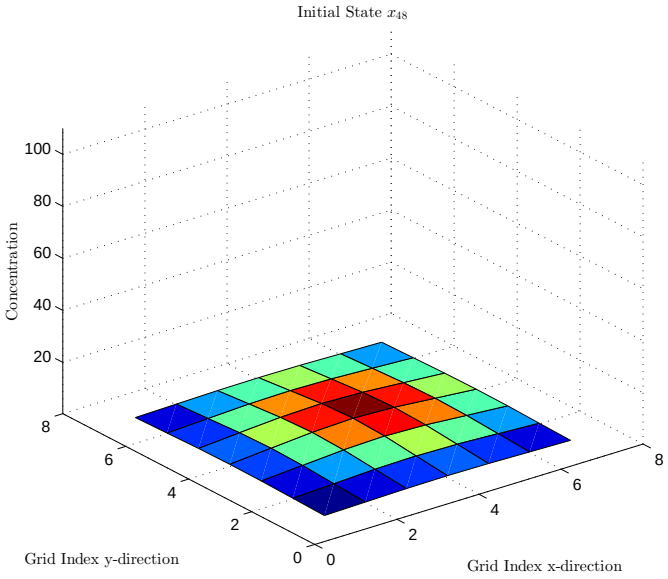

Figure 4.18: Algorithm 2 for the two-dimensional case with $\gamma = 0.04$, $N = 49$, and measurement noise variance of 0.1^2. The graph illustrates the state $\hat{x}_{48}$

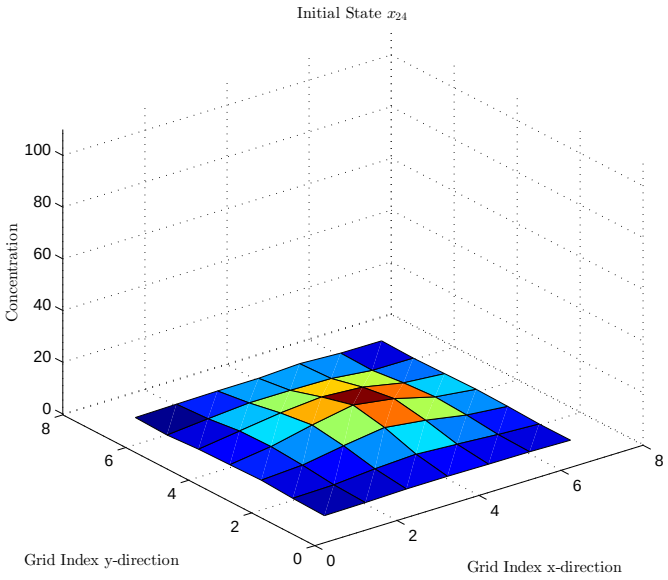

Figure 4.19: Algorithm 2 for the two-dimensional case with $\gamma = 0.04$, $N = 49$, and measurement noise variance of 0.1^2. The graph illustrates the state $\hat{x}_{24}$

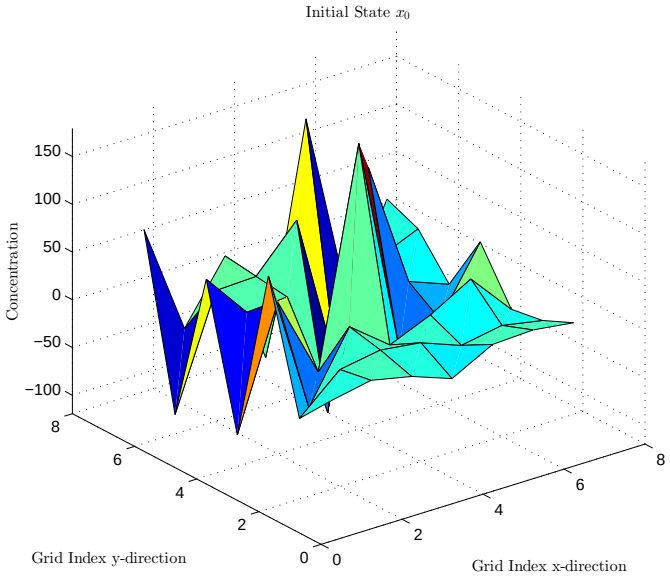

Figure 4.20: Algorithm 2 for the two-dimensional case with $\gamma = 0.04$, $N = 49$, and measurement noise variance of 0.1^2. The graph illustrates the reconstructed initial state $\hat{x}_0$.

Figure 4.16 does not show a clearly visible single release point, which is reflected in MSE value of 8.8×10^3. Not only does the addition of noise effect the reconstruction of the initial state, but the repeated use of a sensor, as seen in Figure 4.17, may also increase the MSE value.

4.4 Comparison of Algorithm 2 and Algorithm 3

In the following simulations, Algorithm 2 and Algorithm 3 are compared over different noise variances from 10^{-12} to 10^{-1}. For each algorithm and variance, the simulation is run 100 times and the MSE is solved for each simulation. The average of the MSE is then plotted against its variance.

4.4.1 One-Dimensional Diffusion System

Two different sets of parameters are used for Figure 4.21 and Figure 4.22. Both experiment setups utilize a one-dimensional diffusion system of size $N = 25$ and the initial $\breve{P}_0 = 100^2 I$ for Algorithm 3. Figure 4.21 uses $\gamma = 0.04$ while Figure 4.22 uses $\gamma = 0.004$.

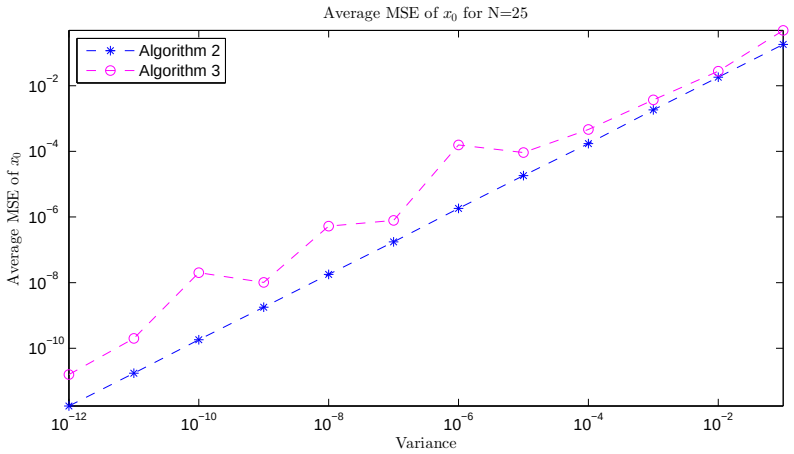

Figure 4.21: Average Mean-Square Error for Algorithm 2 and Algorithm 3 with $\breve{P}_0 = 100^2 I$ for $N = 25$, $\gamma = 0.04$ for a one-dimensional diffusion system.

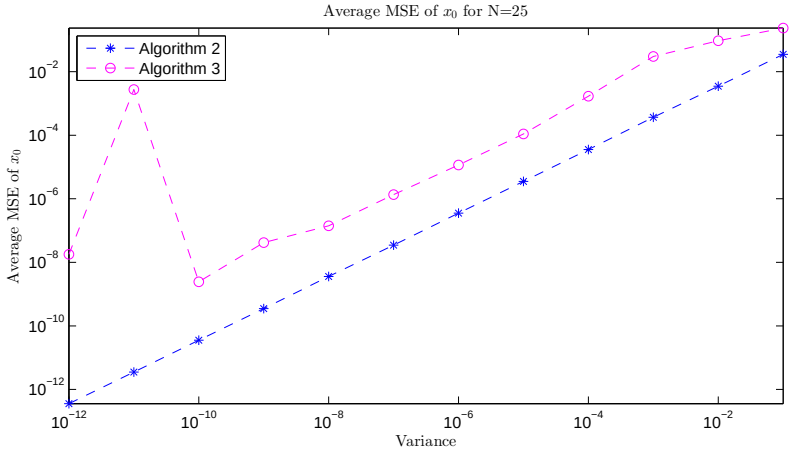

Figure 4.22: Average Mean-Square Error for Algorithm 2 and Algorithm 3 with $\breve{P}_0 = 100^2 I$ for $N = 25$, $\gamma = 0.004$ for a one-dimensional diffusion system.

In both experiments for Figures 4.21 and 4.22, Algorithm 2 yields a smaller average MSE compared to Algorithm 3. When the system is closer to instability ($\gamma = 0.04$),

Algorithm 3 performs closer to Algorithm 2, especially for smaller noise variances. When the system is further away from instability ($\gamma = 0.004$), the Algorithm 3 performs slightly more further away to Algorithm 2 compared to Figure 4.21. There is even more of a difference for smaller variances.

4.4.2 Two-Dimensional Diffusion System

Two different sets of parameters are used for Figure 4.23 and Figure 4.24. Both experiment setups utilize a two-dimensional diffusion system of size $N = 49$ and the initial $\breve{P}_0 = 100^2 I$ for Algorithm 3. Figure 4.23 uses $\gamma = 0.04$ while Figure 4.24 uses $\gamma = 0.004$.

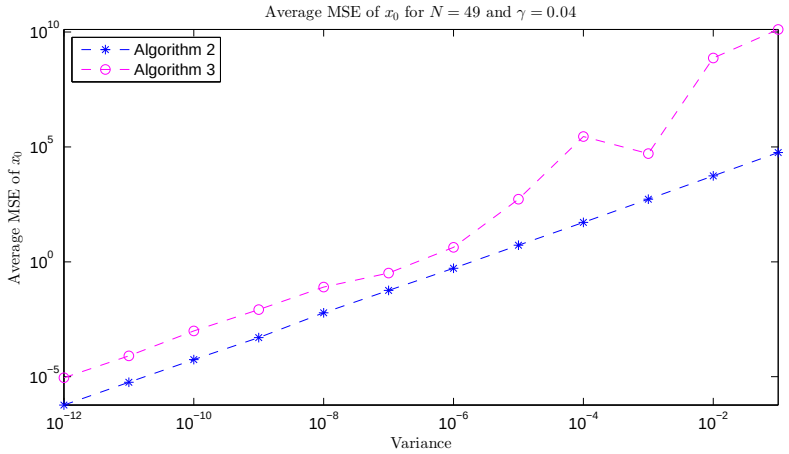

Figure 4.23: Average Mean-Square Error for Algorithm 2 and Algorithm 3 with $\breve{P}_0 = 100^2 I$ for $N = 49$, $\gamma = 0.04$ for a two-dimensional diffusion system.

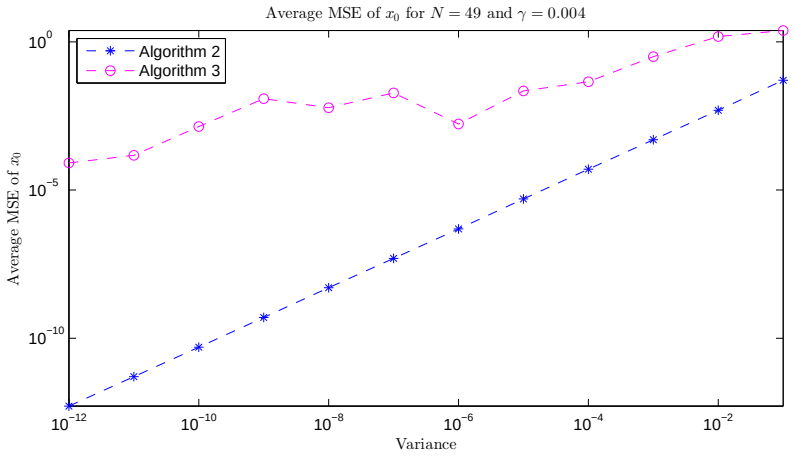

Figure 4.24: Average Mean-Square Error for Algorithm 2 and Algorithm 3 with $\breve{P}_0 = 100^2 I$ for $N = 49$, $\gamma = 0.004$ for a two-dimensional diffusion system.

In both experiments for Figures 4.23 and 4.24, Algorithm 2 yields a smaller average MSE compared to Algorithm 3. When the system is closer to instability ($\gamma = 0.04$), Algorithm 3 performs closer to Algorithm 2 for smaller variances. When the system is further away from instability ($\gamma = 0.004$), Algorithm 3 performs closer to Algorithm 2 for larger variances. However, for smaller variances, Algorithm 3 gives a much higher average MSE value.

Chapter 5

CONCLUSIONS

In modern networked distributed sensor systems, not all sensors operate simultaneously due to constraints. A sensor scheduling problem of estimating a localized source from one-dimensional and two-dimensional discrete diffusion processes was investigated as an observability problem. The definition of observability gives a "yes-no" answer as to whether or not a system is observable. In order to describe the extent of observability, different metrics of have been purposed to address this in literature. Most of the sensor selection methods choose the optimal or sub-optimal sensor location from predetermined sensor schedules. To determine a schedule as the system propagates in time rather than choosing from a predetermined sensor combination, two greedy algorithms, Algorithm 2 and Algorithm 3 were purposed for the one-dimensional and two-dimensional discrete diffusion processes.

In the case with no noise on the measurements, with interest in preforming numerically stable inversion of the observability matrix, the condition number of the observability matrix $\kappa(\Phi)$ was the metric. As the system propagates, Algorithm 1 builds the normalized Gram matrix for the observability matrix of each sensor in the library, and the sensor yielding the highest condition number is selected for that time step. The results of Algorithm 1 shows that as the size of the diffusion matrix N increases, so does the condition number. Also, the further away from instability, the lower the condition number.

In the case with with noise on the measurements, in order to focus on the average degree of uncertainty in the estimated initial state, the trace of the observability Gramian, trace(W^{-1}) was used as a metric. As the system propagates, Algorithm

2 builds the observability matrix and its SVD decomposition of each sensor in the library, and the sensor yielding the smallest trace is selected for that time step. Algorithm 2 is compared to Algorithm 3 and shows to produce a lower average MSE value.

Future work could investigate the effects of allowing the algorithms to take more than one point measurement. This also may allow editing or refinement of the algorithm by investigating the interaction of the eigenstructure of the diffusive system and the selection of sensors at any given time step.

REFERENCES

[1] A. O. Hero and D. Cochran, "Sensor management: Past, present, and future," *IEEE Sensors Journal*, vol. 11, no. 12, pp. 3064–3075, Dec 2011.

[2] M. P. Vitus, W. Zhang, A. Abate, J. Hu, and C. J. Tomlin, "On efficient sensor scheduling for linear dynamical systems," *Automatica (Oxford)*, vol. 48, no. 10, pp. 2482–2493, Oct 2012.

[3] M. Zoghi and M. H. Kahaei, "Adaptive sensor selection in wireless sensor networks for target tracking," *IET Signal Processing*, vol. 4, no. 5, pp. 530–536, 2010.

[4] D. Dochain, N. Tali-Maamar, and J. Babary, "On modelling, monitoring and control of fixed bed bioreactors," *Computers & Chemical Engineering*, vol. 21, no. 11, pp. 1255 – 1266, 1997.

[5] G. E. Hovland and B. J. McCarragher, "Dynamic sensor selection for robotic systems," in *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, vol. 1, 1997, pp. 272–277.

[6] U. Ilkturk, "Observability methods in sensor scheduling," Ph.D. dissertation, Arizona State University, July 2015.

[7] A. V. Savkin, R. J. Evans, and E. Skafidas, "The problem of optimal robust sensor scheduling," *Systems & Control Letters*, vol. 43, no. 2, p. 149–157, June 2001.

[8] D. Sinno and D. Cochran, "Estimation with configurable and constrained sensor systems," in *Defence Applications of Signal Processing*, D. Cochran, W. Moran, and L. B. White, Eds. Elsevier Science B.V., 2001, pp. 203–212.

[9] B. T. Hinson and K. A. Morgansen, "Observability-based optimal sensor placement for flapping airfoil wake estimation," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 5, pp. 1477–1486, Oct 2014.

[10] F. Chraim, Y. Erol, and K. Pister, "Wireless gas leak detection and localization," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2015.

[11] R. Kalman, "On the general theory of control systems," *IRE Transactions on Automatic Control*, vol. 4, no. 3, pp. 110–110, Dec 1959.

[12] R. E. Kalman, "Contributions to the theory of optimal control," in *Control Theory:Twenty-Five Seminal Papers (1932-1981)*, T. Basar, Ed. Wiley-IEEE Press, 2001, pp. 147–166.

[13] D. N. Burghes and A. Graham, *Introduction to control theory, including optimal control*. Ellis Horwood Limited, 1980.

[14] R. E. Kalman, P. L. Falb, and M. A. Arbib, *Topics in mathematical system theory.* McGraw-Hill, 1969.

[15] F. J. Hale, *Introduction to control system analysis and design.* Prentice-Hall, 1988.

[16] C. Chen, *Linear system theory and design*, 3rd ed. Oxford University Press, Inc., 1999.

[17] T. E. Fortmann and K. L. Hitz, *An introduction to linear control systems.* Marcel Dekker, 1977.

[18] R. J. LeVeque, *Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems.* SIAM, 2007, vol. 98.

[19] V. Lystianingrum, B. Hredzak, V. G. Agelidis, and V. S. Djanali, "Observability degree criteria evaluation for temperature observability in a battery string towards optimal thermal sensors placement," in *Proceedings of the Ninth IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, April 2014, pp. 1–6.

[20] F. E. Udwadia1, "Methodology for optimum sensor locations for parameter identification in dynamic systems," *Journal of Engineering Mechanics*, vol. 120, no. 2, pp. 368–390, Feb 1994.

[21] F. Zhao and L. J. Guibas, *Wireless sensor networks - An information processing approach.* Elsevier, 2004.

[22] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*, 2nd ed. Chapman & Hall/CRC, 2004.

[23] P. C. Müller and H. I. Weber, "Analysis and optimization of certain qualities of controllability and observability for linear dynamical systems," *Automatica*, vol. 8, no. 3, pp. 237 – 246, 1972.

[24] W. Waldraff, D. Dochain, S. Bourrel, and A. Magnus, "On the use of observability measures for sensor location in tubular reactor," *Journal of Process Control*, vol. 8, no. 5–6, pp. 497 – 505, 1998.

[25] F. van den Berg, H. Hoefsloot, H. Boelens, and A. Smilde, "Selection of optimal sensor position in a tubular reactor using robust degree of observability criteria," *Chemical Engineering Science*, vol. 55, no. 4, pp. 827 – 837, 2000.

[26] S. Borguet and O. Léonard, "The Fisher information matrix as a relevant tool for sensor selection in engine health monitoring," *International Journal of Rotating Machinery*, vol. 2008, 2008.

[27] T. A. Driscoll and R. J. Braun, *Fundamentals of numerical computation: An introduction using MATLAB.* T.A. Driscoll and R.J. Braun, 2015.

[28] E. Peekalska and R. P. W. Duin, *Dissimilarity representation for pattern recognition: Foundations and applications*. World Scientific Publishing Co., 2005.

[29] G. E. Shilov, *Linear algebra*. Dover Publications, 2012.

[30] R. Stephens, *Essential algorithms: A practical approach to computer algorithms*. John Wiley & Sons, 2013.

[31] P. J. Antsaklis and A. N. Michel, *Linear systems*. Birkhäuser Boston, 2006.

[32] G. E. Shilov, *Kalman filtering: Theory and practice using MATLAB*, 2nd ed. John Wiley & Sons, 2001.

[33] M. Lapuerta, J. P. Hernández, and J. R. Agudelo, "An equation for the estimation of alcohol-air diffusion coefficients for modelling evaporation losses in fuel systems," *Applied Thermal Engineering*, vol. 73, no. 1, pp. 539 – 548, 2014.