Sensor Placement Considering the Observability of Traffic Dynamics: on the Algebraic and Graphical Perspectives

By

XINYUE HU
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Civil and Environmental Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Yueyue Fan, Chair

Michael H. Zhang

Shiqian Ma

Committee in Charge

2024

Contents

List of Figures

List of Tables

Abstract

Traffic data is crucial for traffic operation and management. Traffic sensors serve as one of the most important sources for such data. One important mission of the traffic sensor is to provide an accurate and reliable general picture of the traffic system.

In this dissertation, a new sensor location model is developed to maximize the observability of link densities in a dynamic traffic network described using a piecewise linear ordinary differential equation system. We develop an algebraic approach based on the eigenstructure to determine the sensor location for achieving full observability with a minimal number of sensors. The proposed Algebraic Approach is efficient and generic and it can be applied to any dynamical system with direct state observation. Additionally, a graphical approach based on the concept of structural observability is developed. By exploiting the special property of flow conservation in traffic networks, we derive a simple analytical result that can be used to identify observable components in a partially observable system.

The graphical and algebraic properties of observability are then integrated into a sensor location optimization model considering a wide range of traffic conditions. Through numerical experiments, we demonstrate the good performance of our sensor deployment strategies in terms of the average observability and estimation errors.

Acknowledgments

The five years of my PhD journey at UC Davis have been a time of immense happiness, challenges, and growth.

First and foremost, I wish to express my deepest gratitude to my advisor, Professor Yueyue Fan. Her mentorship, patience, and intellectual insights have guided me throughout this journey. I am incredibly fortunate to have had her unwavering support and encouragement, which have been invaluable to my development as a researcher.

I am also deeply appreciative of the members of my dissertation committee, Professor Michael Zhang and Professor Shiqian Ma, as well as my QE committee member, Professor Jia Li. Their constructive suggestions and feedback have greatly enriched this work.

I would like to extend my sincere thanks to my colleagues and friends Ran Sun, Hang Gao, Chenlu Pu and Han Yang. The technical discussions, support, and the exchange of ideas have played a significant role in shaping this dissertation. I am also grateful to my wonderful roommates Yue Wu, Ji Wang, Likang Yin, and Ziwen Kan for their companionship and support throughout this fun journey.

Lastly, my heartfelt thanks go to my parents, Xuefen Xie and Hao Hu, and my sister Xiaoyue Hu for the endless patience, love, and support.

Chapter 1

Introduction

In the era of booming data, we are surrounded by an abundance of information. Within the realm of transportation, intelligent transportation systems (ITS) are transforming our daily lives—ranging from e-hailing taxi services and real-time navigation to the advent of self-driving vehicles. These innovations are powered by data and the insights derived from it. However, critical questions remain regarding what information to collect, where to collect it, and when it should be gathered, as we still face the challenge of incomplete data coverage.

Accurate and reliable traffic data is crucial for effective traffic operation and management. For instance, data on vehicle speeds, traffic volumes, and congestion patterns are essential for optimizing traffic flow, reducing delays, and improving road safety. Traffic sensors, which capture these vital pieces of information, are among the most important data sources for transportation planners. However, given the constraint of limited budgets, deploying these sensors poses a significant challenge: How can we strategically place a finite number of sensors to maximize the value of the data collected? The objective is to ensure that the information gathered is as comprehensive and insightful as possible, enabling more informed decisions and efficient management of transportation systems.

In the context of maximizing the richness of information in a traffic network, the concept of observability has been a key focus. Observability refers to the ability to determine whether a

given set of measurements is sufficient to uniquely infer the states of a traffic network, such as flows, speeds, and densities. This concept has driven much of the existing research on traffic sensor placement.

Most traffic sensor placement literature on observability focused on aggregated flow counts and are thus referred to as the static observability problem in Castillo et al. (2015). Previous studies have provided simple and efficient exact solutions for achieving both full and partial observability in static scenarios. This body of work has significantly advanced our understanding of how to strategically place sensors to gather aggregated flow data across a network. However, these approaches are primarily focused on static, time-invariant data, which limits their applicability in dynamic traffic environments.

In dynamical systems, observability refers to the ability to infer the internal states of a system based on its external outputs from sensors. This concept, alongside its counterpart controllability, has been extensively explored in control theory. Recent research has applied the concept of observability to sensor placement in dynamic traffic systems. Studies have explored sensor placement strategies within nonlinear dynamic traffic network models, addressing the challenges posed by nonlinearity either through linearization around an equilibrium point or by assuming a known sample path of the dynamic process. However, due to the variability in traffic conditions and the limitations of fixed equilibrium points or strong prior information, there is still a need for a systematic and effective sensor placement method that can maximize information gain while accommodating a wide range of dynamic traffic conditions.

An extended concept of observability, structural observability, has emerged, focusing on the structural aspects of the algebraic conditions of a system. Structural observability simplifies the need for precise knowledge of system dynamics and can often be equivalent to standard observability in some contexts. This concept has also been applied to traffic network studies. However, most results have been derived for general dynamical systems. This study aims to explore specific properties related to traffic network density dynamics and develop new theories incorporating domain knowledge.

This dissertation seeks to address these limitations by proposing sensor location optimization methods that consider the observability of traffic density under various dynamic traffic conditions. The dynamics of a highway network are captured using a piecewise-linear macroscopic traffic model, and the research explores both algebraic and graphical properties of such systems. The main contributions of this research are as follows:

- We establish new understanding of graphical and algebraic properties of observability beyond what has been known for general linear dynamic systems by exploiting unique characteristics of traffic networks. Our results advance the current understanding of observability concept in the transportation science literature.
- We incorporate the structural observability concept to the design of an optimal sensor location problem to maximize information gain from sensors under various traffic conditions. Numerical experiments demonstrate the effectiveness of our optimal sensor location strategies in terms of observability and traffic estimation quality.
- The analytical and computational advantages of our results are made possible by exploiting special features of the dynamic traffic networks, which showcases how domain expertise from the transportation field could be integrated with and extend the current state of knowledge of complex systems.

The rest of the dissertation is organized as follows. In Chapter 3, we introduce the mathematical modeling of the traffic network density dynamics and develop the Algebraic Approach to find sensor locations for full (exact) observability with a minimum number of sensors. In Chapter 4, we establish the mathematical properties and analyses associated with structural observability to identify the observable components in a partially observable system. We then incorporate this understanding to develop a sensor location optimization based on the Graphical Approach. In Chapter 5, we present numerical examples to evaluate the quality of sensor location solutions derived from our models, demonstrating that the proposed optimization strategies result in high-quality traffic estimations. Finally, the dissertation concludes with a discussion of our findings and potential directions for future research.

Chapter 2

Literature Review

2.1 Traffic sensor location problem

In the early literature, the sensor location problem has been mostly addressed as a sub-problem of the broader OD demand estimation problem (Yang and Zhou (1998), Bianco et al. (2001), Ehlert et al. (2006)), rather than as an independent problem. In such problems, some prior knowledge such as prior OD demand or link usage proportion matrix is needed. The common practice is to formulate the sensor location problem as an integer programming to guarantee a reliable OD demand estimation. A famous example is the OD covering rule (Yang and Zhou, 1998) that a certain portion of trips between any OD pair should be observed by a sensor. More rule-based optimization examples can be found in the synthesis paper in Gentili and Mirchandani (2012).

In addition to the optimization framework, assuming some statistical priori distribution, Fei et al. (2007), Zhou and List (2010) and Fei et al. (2013) applied the Kalman filter to maximizes the information gain while minimizing the uncertainties of the estimated OD demand matrix. Under this framework, however, it is not feasible to derive a global optimal set of sensor location when the dimension of problem becomes large. Recent sensor location problems can be found in the synthesis paper of Gentili and Mirchandani (2012) and Castillo et al. (2015).

In Gentili and Mirchandani (2012), the traffic sensor location problem is divided into two

categories: (a) flow estimation problems and (b) observability problems. In the estimation problem, we are not able to directly observe or infer all the flows. Instead, some prior knowledge about the flow itself (OD trips, route flow or link flow) is needed. Such information is obtained through survey or historical data. The aforementioned OD estimation sensor location problem literature belong to this category.

With the continuing advances in traffic surveillance, the technology is becoming more reliable and cost-effective. The growing need to fully understand traffic pattern and performance urges people to deploy sensors that we can infer all the information. As a result, the observability concept from the control theory is introduced to the transportation community by Castillo et al. (2008a). In their work, observability is referred to the ability to infer a subset of OD pair or link flows based on a subset of observed OD pair and link flows. The observability in traffic networks can either refer to the static case (states satisfy a linear system of equations based on flow conservation) or the dynamic case (states is described using a dynamical system). In the following two subsections, we will briefly discuss the related literature in both cases.

2.2 Static observability

Under the static condition, flow conservation holds as a linear system of equations. After brought up in Castillo et al. (2008a), observability analysis under the static case is being extensively studied. Many of the discussion centered on the algebraic techniques of the null-space method (Castillo et al. (2006), Castillo et al. (2008b), Castillo et al. (2010), Castillo et al. (2013)) since the observability problem is to find the inverse from a system of equations.

Existing approaches already provide simple and elegant exact solutions for full observability in the static case. Hu et al. (2009) first introduced the problem of finding the smallest subset of links to locate counting sensors for full observability. The matrix algebraic method based on the reduced row echelon form (RREF) of the link-path incidence matrix is very efficient to solve. To avoid the large dimension of the link-path incidence matrix, Ng (2012) developed a node-based formulation of the problem. He (2013) addressed the same problem from a graphical point of view and showed

that a minimum spanning tree of the modified network will give exactly the smallest subset for full observability.

Full observability requires 60%-70% (Hu et al. (2009)) of the network links to be equipped with sensors. Viti et al. (2014) argued that this number is too large for realistic traffic networks and therefore studied the partially observable systems. They proposed a null space metric to evaluate the degree of partial observability and developed a heuristic sensor deployment algorithm to maximize the null space metric. Also, some researchers addressed the stochasticity in the sensor location problem. Salari et al. (2019), Li and Ouyang (2011) and Danczyk et al. (2016) examined the sensor locations considering the failure of sensors.

2.3 Dynamic observability

For dynamical systems, observability is a measure of how well the internal states of a system can be inferred from knowledge of external outputs (from sensors). Observability and its mathematical dual controllability have been well-studied in the control theory since introduced by Kalman (1960).

To distinguish it from the static observability, Castillo et al. (2015) referred to the observability in dynamic case as the dynamic observability. They also pointed out that unlike the static case which has been widely discussed in the existing literature, the dynamic observability in traffic systems remains as an interesting and important direction for future exploration. Compared with the static case where a linear relationship is used to describe the system, the dynamic case considers the observability throughout the time, indicating that we need to consider the traffic network as a dynamical system.

It is worth noting that in Fei et al. (2007), time-varying network flow is considered by slicing the time in pieces and apply a dynamic traffic assignment in each time slice. However, this is still inherently different from considering a dynamic evolving traffic system.

Contreras et al. (2015) first addressed the dynamic observability in detail on highway segments. In this work, traffic dynamics is modeled using a nonlinear ODE system considering flow conservation and the link density is the state variable. Observability condition is investigated

for local densities of highway segments by utilizing linearized traffic dynamics about steady state flows. To extend the discussion to network level observability, Agarwal et al. (2015) considered both density dynamics and routing dynamic in an ODE traffic model. The knowledge of routing dynamics makes it possible to keep track of real-time split ratio. However, in real traffic systems, routing dynamics is often unknown since few sensing technologies can give such information. In both Agarwal et al. (2015) and Contreras et al. (2015), the observability condition is examined. Both highway corridor and network lead to the similar conclusion that under uncongested case, downstream links are more important for observability; and under congested situation, upstream links are more important. **However,in the existing literature for dynamic observability, there is no systematic and efficient algorithm to obtain the sensor location scheme that guarantees full observability.**

2.3.1 Structural observability and graph-theoretic approaches

More recently, graph-theoretic approaches have been developed to explore an extended concept of observability, structural observability, which focuses on the structure of the algebraic conditions (Lin, 1974; Liu et al., 2011, 2013). Structural observability bypasses the need for accurate/exact knowledge of system dynamics and in many cases, it is equivalent to observability. This concept and the graphical approaches are also borrowed in recent traffic network observability studies. Bekiaris-Liberis et al. (2017) developed traffic state estimation approach for per-lane density estimation and investigated the structural observability property based on the graphical approach. Rostami-Shahrbabaki et al. (2020) studied the state estimation problem in urban traffic networks and studied the sensor location problem based on the Strongly Connected Component method in Liu et al. (2013). Mousavi and Kouvelas (2020). Mousavi and Kouvelas (2020) discussed the structural observability of density dynamics on a motorway ring road, also based on graph-theoretic approaches.

In recent years, more sensor placement studies incorporated the observability concept in a dynamic traffic system (Agarwal et al., 2015; Bekiaris-Liberis et al., 2017; Contreras et al., 2015). For example, based on nonlinear dynamic traffic network modeling, sensor placement strategies

have been studied in Contreras et al. (2015) and Nugroho et al. (2021). The non-linearity has to be addressed by either linearizing the problem around an equilibrium point or assuming a known sample path of the dynamic process. However, traffic conditions are subject to variations and a fixed equilibrium point may not exist. A known sample path is too strong as a prior information. **Therefore, a systematic and effective sensor placement method that could maximize information gain while being capable of handling a large spectrum of dynamic traffic conditions is still in need.**

Chapter 3

Algebraic Approach for Full Observability

3.1 Traffic network as a dynamical system

In order to investigate an optimal sensor deployment for information collection on dynamic traffic networks, we first need a dynamical model to describe the traffic system. Our choice here are macroscopic traffic models since we are only interested in the aggregate behavior of vehicles. To derive the mathematical properties, simple analytical models are desired. Although it might oversimplify the highly nonlinear, complicated real traffic network, the capability to analyze the system properties can provide valuable insights.

The famous LWR model proposed by Lighthill and Whitham (1955) and Richards (1956) is a macroscopic traffic flow model that captures shock and rarefaction waves of fluid-like traffic. The flow conservation equation can be written as:

$$\frac{\partial k(x,t)}{\partial t} + \frac{\partial q(x,t)}{\partial x} = 0 \tag{3.1}$$

where k and q denote the traffic density and flow, respectively. In addition, an equilibrium speed-flow relationship is assumed which refers to the fundamental diagram. This model is a partial

differential equation model.

The cell transmission model (CTM) is developed to approximate the LWR model (Daganzo, 1994, 1995). The CTM model divides a link into small cells with the same length and defines each cell's demand (maximum sending flow) and supply (maximum receiving flow) to describe interactions between adjacent freeway cells as well as shockwaves. The actual flow transmitted is determined by the minimum of the upstream demand and downstream supply. There are different ways to define demand and supply functions based on the assumption of the fundamental diagram. To avoid the overwhelmingly large number of cells for better efficiency, Yperman (2007) proposed the Link Transmission Model which requires only one sending flow and one receiving flow for an entire link. However, such demand and supply functions are delayed functions in in- and out-flows, making it less suitable for observability-based sensor location problems.

To avoid the nonlinearity in CTM, the switching-mode model (SMM) was proposed by Muñoz et al. (2003). The SMM is a piecewise affine system that switches among different sets of linear difference equations (representing different traffic states of the freeway), depending on boundary flow and the congestion status of the cells in a freeway segment and assuming a triangular fundamental diagram. Adopting traffic density as the state variable, the SMM permits the uneven cell lengths, which greatly reduces the dimension of the system. However, this model works only for highway segments as it does not include merge and diverge behaviors.

Link Queue Model

To develop a traffic network model whose junction model is realistic and link model is mathematically tractable, Jin (2021) developed a deterministic queueing model, the link queue model (LQM). The LQM is a system of ordinary differential equations and the state variable is the queue length (number of vehicles on a link). Similar to the CTM, the LQM defines the demand and supply of a queue as functions of the queue length to determine the actual flow between 2 links. The LQM is defined as:

$$\dot{\mathbf{K}}(t) = \mathbf{f}(\mathbf{K}(t)) - \mathbf{g}(\mathbf{K}(t)) \tag{3.2}$$

where $\mathbf{K}(t)$ is the set of the numbers of vehicles on all links, and $\mathbf{f}(\mathbf{K}(t))$ and $\mathbf{g}(\mathbf{K}(t))$ are the sets of in- and out-flows computed from the $\mathbf{K}(t)$ based on the demand and supply.

We adopt the equivalent LQM by using link density as the state variable instead of link queue length. Let k_a and l_a, respectively, denote the traffic density and length of link a, Eq.(3.2) is rewritten as:

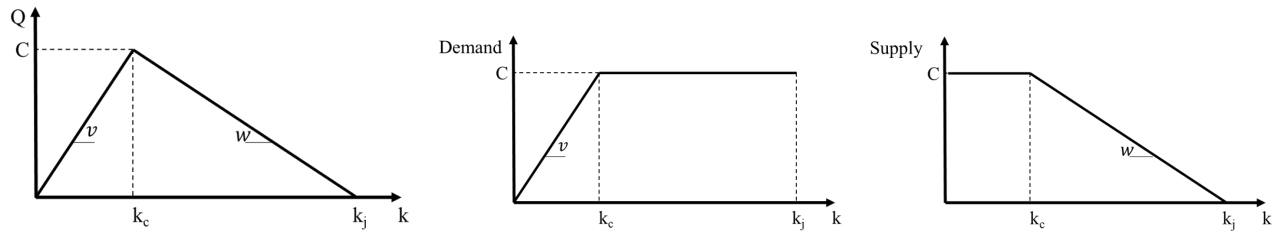
$$\dot{k}_a(t) = \frac{f_a(t) - g_a(t)}{l_a} \tag{3.3}$$

Though LQM allows for various types of fundamental diagrams, we use triangular fundamental diagrams in the further discussion for simplicity. A fundamental diagram depicts the flow-density relationship. A triangular fundamental diagram is shown in Figure 3.1a, where v is the free-flow speed and w is the congestion wave speed. The maximum possible flow rate C, also known as the capacity, is achieved at the critical density k_c. As shown in Figure 3.1b 3.1c, the demand and supply functions of link of density k is defined as:

$$d(t) = \begin{cases} vk(t), \text{ if not congested}(k(t) < k_c) \\ C, \text{ if congested}(k(t) \geq k_c) \end{cases} \tag{3.4a}$$

$$s(t) = \begin{cases} C, \text{ if not congested}(k(t) < k_c) \\ -w(k(t) - k_j), \text{ if congested}(k(t) \geq k_c) \end{cases} \tag{3.4b}$$

where k_c, k_j are the critical density and jam density of the link, respectively.



(a) Triangular fundamental diagram (b) Link demand (maximum sending flow) (c) Link supply (maximum receiving flow)

Figure 3.1: Fundamental diagram, demand and supply functions

Compared with the SMM, the LQM defines the merge and diverge behaviors and therefore is capable of describing the traffic network dynamics. We will briefly illustrate the formulation of three types of fundamental junctions in highway networks, namely the linear, merge, and diverge junctions, as shown in Figure 3.2.

For a linear junction with upstream link 1 and downstream link 2 in Figure 3.2a, the flux between link 1 and 2 can be derived as follows:

$$g_1(t) = f_2(t) = \min\{d_1(t), s_2(t)\} \tag{3.5}$$

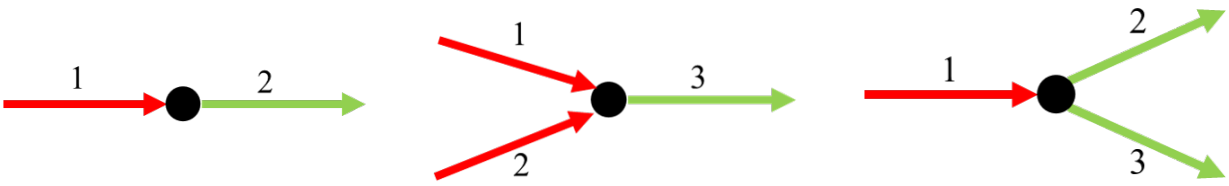
For a merge with two upstream links, 1 and 2, and a downstream link, 3, as shown in Figure 3.2b, the flux can be derived if we assume a merging priority:

$$f_3(t) = \min\left\{d_1(t) + d_2(t), s_3(t)\right\}, \tag{3.6a}$$

$$g_1(t) = \min\left\{d_1(t), \max\left\{s_3(t) - d_2(t), \alpha s_3(t)\right\}\right\}, \tag{3.6b}$$

$$g_2(t) = f_3(t) - g_1(t). \tag{3.6c}$$

where α is the merging priority of link 1. If link 1 and link 2 have the same priority to merge, the situation is called a fair merge[1] and $\alpha = \frac{C_1}{C_1+C_2}$. Note that, any choice of α will be fine and fair merge is not the only choice.



(a) A linear junction (b) A merge (c) A diverge

Figure 3.2: Three Types of Junctions

[1]Note that fair merge is only presented as an example. Our model and analyses presented in this paper can accommodate various node merge models, not only fair merge. As long as the merge node model after the min, max, median function results in piecewise linear functions, the results are applicable.

For a diverge with an upstream link 1, and two downstream links, 2 and 3, as shown in Figure 3.2c, the flux can be derived if all vehicles have predefined route choices and follow the density-proportional diverging rule:

$$g_1(t) = \min\left\{d_1(t), \frac{s_2(t)}{\xi_{1\to 2}(t)}, \frac{s_3(t)}{\xi_{1\to 3}(t)}\right\}, \tag{3.7a}$$

$$f_2(t) = \xi_{1\to 2}(t) g_1(t), \tag{3.7b}$$

$$f_3(t) = \xi_{1\to 3}(t) g_1(t). \tag{3.7c}$$

where ξ denotes the split ratio at the junction. In our following discussion, we assume the split ratio is known and remains as constant throughout the time.

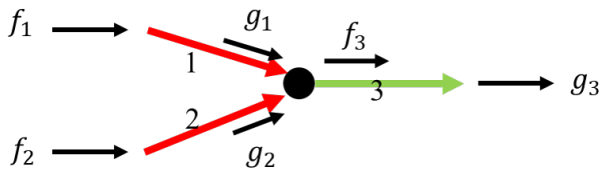


Figure 3.3: A toy merge junction

Now we see that the LQM is a system of first-order, nonlinear ordinary differential equations. However, similar to the SMM, the LQM system equations will remain the same when state variables stay within some range. The range here refers to the polyhedron formed by the state variables. Therefore, the LQM is a piecewise affine system. To illustrate the idea, we show a toy example of a merge junction.

Example In a toy network shown in Figure 3.3, we know all the boundary flows f_1, f_2, g_3, and the free-flow speed v_1, v_2, v_3 and the congestion wave speed w_1, w_2 and w_3. Assume the link lengths l_1, l_2 and l_3 are the same, and equal to 1 for simplicity. Assume the initial density of each link is zero and the in-flow is large enough for us to observe a bottleneck at link 3.

Before bottleneck occurs, all the links are uncongested and there is no bottleneck:

$s_3 > d_1 + d_2$, therefore $f_3 = \min(s_3, d_1 + d_2) = d_1 + d_2 = v_1 k_1 + v_2 k_2$, $g_1 = d_1 = v_1 k_1$, $g_2 = d_2 = v_2 k_2$.

The system of equations is:

$$\begin{aligned}
\dot{k_1} &= \frac{f_1 - g_1}{l_1} = f_1 - v_1 k_1 \\
\dot{k_2} &= \frac{f_2 - g_2}{l_2} = f_2 - v_2 k_2 \\
\dot{k_3} &= \frac{f_3 - g_3}{l_3} = v_1 k_1 + v_2 k_2 - g_3
\end{aligned} \tag{3.8}$$

In matrix form,

$$\begin{bmatrix} \dot{k_1} \\ \dot{k_2} \\ \dot{k_3} \end{bmatrix} = \begin{bmatrix} -v_1 & 0 & 0 \\ 0 & -v_2 & 0 \\ v_1 & v_2 & 0 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \\ -g_3 \end{bmatrix} \tag{3.9}$$

which is a linear time-invariant system.

After bottleneck occurs, assume all the links are still uncongested but the difference is: $s_3 < d_1 + d_2$, therefore $f_3 = \min(s_3, d_1 + d_2) = s_3 = k_3 v_3$, if we assume a fair merge rule $g_1 = \frac{C_1}{C_1+C_2} s_3 = \frac{C_1}{C_1+C_2} C_3$, $g_2 = \frac{C_2}{C_1+C_2} s_3 = \frac{C_2}{C_1+C_2} C_3$. The system of equations is:

$$\begin{aligned}
\dot{k_1} &= \frac{f_1 - g_1}{l_1} = f_1 - \frac{C_1}{C_1 + C_2} C_3 \\
\dot{k_2} &= \frac{f_2 - g_2}{l_2} = f_2 - \frac{C_2}{C_1 + C_2} C_3 \\
\dot{k_3} &= \frac{f_3 - g_3}{l_3} = C_3 - g_3
\end{aligned} \tag{3.10}$$

In matrix form,

$$\begin{bmatrix} \dot{k_1} \\ \dot{k_2} \\ \dot{k_3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} + \begin{bmatrix} f_1 - \frac{C_1}{C_1+C_2} C_3 \\ f_2 - \frac{C_2}{C_1+C_2} C_3 \\ C_3 - g_3 \end{bmatrix} \tag{3.11}$$

The dynamic system will remain the same until link 3 becomes congested and the supply is no longer C_3; instead, it will be a function of k_3 again.

This toy example illustrate the point that the LQM can be considered as a piecewise affine system. When the link density is within a certain range, the LQM is a linear ODE system. As a result, many methodologies developed in the linear systems analysis can be used.

LQM validation using PeMS data

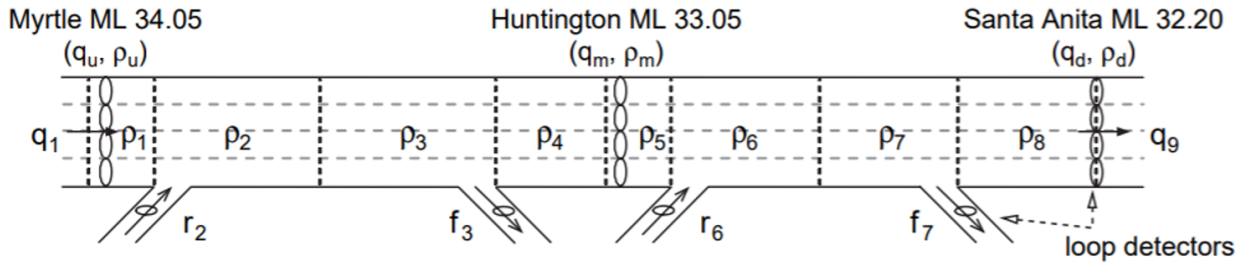
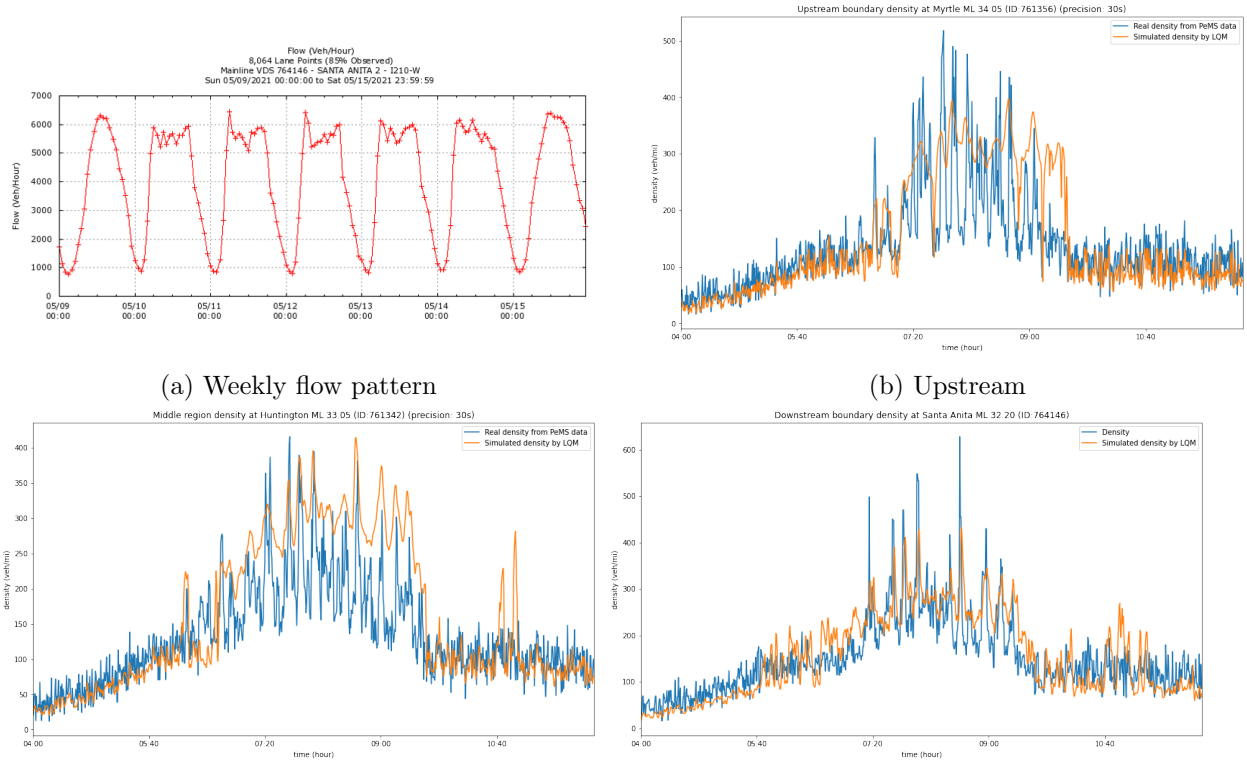


Figure 3.4: Segment of I280W divided into several links



(a) Weekly flow pattern

(b) Upstream

(c) Middle region

(d) Downstream

Figure 3.5: Simulated density using LQM compared with real data

To validate the LQM, we also examined its performance on describing traffic dynamics using real traffic data on a subsection of I-280 West. As shown in Figure 3.4, the road segment we choose is approximately 2 miles in length, with four mainline lanes, three mainline loop detector stations labeled Myrtle (ML 34.05), Huntington (ML 33.05), Santa Anita (ML 32.20), and additional detector stations on each ramp. The loop detector collects the data of the boundary flows as well as the occupancy (which can be used to compute density) at a precision of 30 seconds. We use data from Caltrans Performance Measurement System (PeMS) during 4:00AM to 12:00PM on May 11th, 2021 and verified there is no special events affecting the flow pattern in that day which is shown in Figure 3.5a. The boundary flows are adjusted using the method in Muñoz et al. (2003) to maintain realistic flows. The simulated traffic density by the LQM is compared with the ground truth in Figure 3.5. Despite being less accurate when there are large spikes in density, We find that the LQM provides satisfactory result, capturing the overall tendency of traffic density evolution. However, we need to note that in this example, the link length is quite short. For large networks, we need to divide long links into smaller links to guarantee better performance.

Here, we want to re-emphasize why we choose the link queue model: (1) the junction model is realistic and link model is mathematically tractable; (2) the piecewise affine property greatly simplifies further analysis and the rich literature on linear systems can be borrowed; (3) it captures the traffic dynamics at least qualitatively.

3.2 Preliminary on observability

For the reader's convenience, this section introduces the concept of observability for linear time-invariant systems. It includes a review of existing theorems and lemmas related to this topic.

Consider the following time-invariant dynamical system with $\boldsymbol{x}$ being the state variable (link densities) and $\boldsymbol{u}$ being the known input,

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + \boldsymbol{u}(t),\ \boldsymbol{x}(t_0) = \boldsymbol{x}_0 = \text{unknown} \tag{3.12}$$

With sensors, we obtain the corresponding measurements

$$\boldsymbol{y}(t) = C\boldsymbol{x}(t) \tag{3.13}$$

of dimensions $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^p$, $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{p\times n}$. C is the mapping from the states to measurements and p is the number of measurements.

Since the control input $\boldsymbol{u}(t)$ is known, we conclude that knowing $\boldsymbol{x}(t_0)$ is sufficient to determine $\boldsymbol{x}(t)$ at any time. To solve for $\boldsymbol{x}(t_0)$, we can solve a linear system of equations:

$$\begin{aligned}
&\boldsymbol{y}(t_0) = C\boldsymbol{x}(t_0)\\
&\dot{\boldsymbol{y}}(t_0) = C\dot{\boldsymbol{x}}(t_0) = CA\boldsymbol{x}(t_0) + C\boldsymbol{u}(t_0)\\
&\ddot{\boldsymbol{y}}(t_0) = C\ddot{\boldsymbol{x}}(t_0) = CA^2\boldsymbol{x}(t_0) + CA\boldsymbol{u}(t_0) + C\dot{\boldsymbol{u}}(t_0)\\
&\vdots\\
&\boldsymbol{y}^{(n-1)}(t_0) = C\boldsymbol{x}^{(n-1)}(t_0) = CA^{n-1}\boldsymbol{x}(t_0) + CA^{n-2}\boldsymbol{u}(t_0) + \cdots + C\boldsymbol{u}^{n-2}(t_0)
\end{aligned}$$

This is a system of np linear equations. In matrix form,

$$\begin{bmatrix} \boldsymbol{y}(t_0) \\ \dot{\boldsymbol{y}}(t_0) \\ \ddot{\boldsymbol{y}}(t_0) \\ \vdots \\ \boldsymbol{y}^{(n-1)}(t_0) \end{bmatrix}^{(np)\times 1} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}^{(np)\times n} \times \boldsymbol{x}(t_0) + D \begin{bmatrix} \boldsymbol{u}(t_0) \\ \dot{\boldsymbol{u}}(t_0) \\ \ddot{\boldsymbol{u}}(t_0) \\ \vdots \\ \boldsymbol{u}^{n-2}(t_0) \end{bmatrix}^{n(n-1)\times 1} \tag{3.14}$$

where $D = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ C & 0 & 0 & \dots & 0 \\ CA & C & 0 & \dots & 0 \\ \vdots & \vdots & & & \vdots \\ CA^{n-2} & CA^{n-3} & CA^{n-4} & \dots & C \end{bmatrix}^{(np)\times n(n-1)}$.

Notice that, the only unknown in Eq. (3.14) is $\boldsymbol{x}(t_0)$. Rearrange the terms, we obtain

$$\begin{bmatrix} \boldsymbol{y}(t_0) \\ \dot{\boldsymbol{y}}(t_0) \\ \ddot{\boldsymbol{y}}(t_0) \\ \vdots \\ \boldsymbol{y}^{(n-1)}(t_0) \end{bmatrix} - D \begin{bmatrix} \boldsymbol{u}(t_0) \\ \dot{\boldsymbol{u}}(t_0) \\ \ddot{\boldsymbol{u}}(t_0) \\ \vdots \\ \boldsymbol{u}^{n-2}(t_0) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \times \boldsymbol{x}(t_0) = \mathcal{O}\boldsymbol{x}(t_0) \tag{3.15}$$

The LHS of Eq. (3.15) is known. In order to be able to solve for $\boldsymbol{x}(t_0)$, matrix $\mathcal{O}$ has to be full ranked ($= n$).

Kalman's rank test

The observability matrix $\mathcal{O}$ is defined as $\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$. The system is observable when $rank(\mathcal{O}) = n$ (the observability matrix has full column rank) and $\boldsymbol{x}_0$ can be uniquely determined. This condition is called the **Kalman's rank test**. This is also referred to as the **exact observability** since the exact parameterization of the system dynamics is known. One might recall that in a static problem, observability is also measured by the rank of a matrix. The difference is the matrix - in a static network problem, one focuses on the matrix from the system of linear equations based on flow conservation, which only reflects the connection of various information pieces (such as link flows) over the spatial dimension. While in a dynamic problem, the focus is on the observability matrix $\mathcal{O}$, which reflects the connection of various information pieces (i.e. state variables such as link densities) in both spatial and time dimensions described by the system of ordinary differential equations (ODEs).

Example We now show a simple example to illustrate observability in dynamic traffic systems. In the toy network in Figure 3.3, when all the links are uncongested and sum of the demands is

less than the supply, we would get a system like in Eq. (3.9):

$$\begin{bmatrix} \dot{k_1} \\ \dot{k_2} \\ \dot{k_3} \end{bmatrix} = \begin{bmatrix} -v_1 & 0 & 0 \\ 0 & -v_2 & 0 \\ v_1 & v_2 & 0 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \\ -g_3 \end{bmatrix}$$

Now, if we have a sensor on link 3, the corresponding matrix $C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$. The observability matrix is

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ v_1 & v_2 & 0 \\ -v_1^2 & -v_2^2 & 0 \end{bmatrix}, \tag{3.16}$$

$\mathcal{O}$ is of full rank iff $v_1 \neq v_2$.

From this example, we recognize that the parameter of the system does affect the observability conclusion. Placing a sensor on link 3 is sufficient to guarantee full observability in most cases. We would like to point out that if we only consider the static network and topological relationship, we would never be able to fully observe the system with only one sensor. The traffic dynamics give us richer information although it is more demanding in terms of the prior knowledge of system parameters.

Observability and similarity transformation

An important concept of observability is the observable/unobservable subspace. The similarity transformation is needed for such a separation. Before introducing the similarity transformation, the invariant subspace of the observability matrix $\mathcal{O}$ is discussed.

Definition 3.1 (Invariant subspace)**.** *A subspace $\mathcal{V}$ of $\boldsymbol{R}^n$ or $\boldsymbol{C}^n$ (if A is complex) is called A-invariant or an invariant subspace of A, if for every $v \in \mathcal{V}$ the vector Av is also in $\mathcal{V}$.*

Lemma 3.2. *Null space of $\mathcal{O}$, $\mathcal{N}(\mathcal{O})$ is A-invariant: $\forall v \in \mathcal{N}(\mathcal{O}) \Rightarrow Av \in \mathcal{N}(\mathcal{O})$.*

Proof. Based on Cayley-Hamilton theorem, $\mathcal{N}\left(\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n+k} \end{bmatrix}\right), \forall k \geq 0.$

We know $\forall v \in \mathcal{N}(\mathcal{O}), \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} v = 0$. To check whether $Av \in \mathcal{N}(\mathcal{O})$, we compute

$$\mathcal{O}Av = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} Av = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{n} \end{bmatrix} v.$$

It is obvious that the null spaces have the following relationship

$$\mathcal{N}\left(\begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{n} \end{bmatrix}\right) \subseteq \mathcal{N}\left(\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n} \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}\right)$$

Therefore $v \in \mathcal{N}(\mathcal{O})$ implies $Av \in \mathcal{N}(\mathcal{O})$. □

Lemma 3.3. *If rank$(\mathcal{O}) = n_0 < n$, then there exists a similarity transformation matrix T such that $\hat{x} = T^{-1}x$, $\hat{A} = T^{-1}AT$, $\hat{C} = CT$, and $\hat{A}$ and $\hat{C}$ take the form:*

$$\hat{A} = \begin{bmatrix} A_o & 0 \\ A_{21} & A_u \end{bmatrix}, \quad \hat{C} = \begin{bmatrix} C_o & 0 \end{bmatrix},$$

where A_o has dimension $n_0 \times n_0$ and (C_o, A_o) is observable.

Proof. Let $\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$ denote the observability matrix, and let $\mathcal{N}(\mathcal{O})$ be its null space, which has dimension $n - n_0$. Let $\{v_1, \ldots, v_{n-n_0}\}$ be a basis for $\mathcal{N}(\mathcal{O})$. Define

$$T = \begin{bmatrix} T_{n_0} & v_1 & \ldots & v_{n-n_0} \end{bmatrix},$$

where T_{n_0} is a matrix with n_0 linearly independent columns, chosen so that T is non-singular.

To show that $\hat{A} = T^{-1}AT$ has the desired form, note that

$$AT = \begin{bmatrix} AT_{n_0} & Av_1 & \ldots & Av_{n-n_0} \end{bmatrix}.$$

Since $\mathcal{N}(\mathcal{O})$ is A-invariant, Av_i lies in $\mathcal{N}(\mathcal{O})$, so each Av_i can be expressed as a linear combination of $\{v_1, \ldots, v_{n-n_0}\}$, resulting in the 0's in the block form for $\hat{A}$.

Similarly, for $\hat{C} = CT$, we have

$$\hat{C} = \begin{bmatrix} CT_{n_0} & Cv_1 & \ldots & Cv_{n-n_0} \end{bmatrix}.$$

Since $v_i \in \mathcal{N}(\mathcal{O})$, $Cv_i = 0$, which leads to $\hat{C} = \begin{bmatrix} C_o & 0 \end{bmatrix}$. □

Lemma 3.4. *Observability is not affected by similarity transformations.*

Proof. Let $\hat{A} = T^{-1}AT$ and $\hat{C} = CT$ for some nonsingular T, then:

$$\hat{\mathcal{O}} = \begin{bmatrix} \hat{C} \\ \hat{C}\hat{A} \\ \hat{C}\hat{A}^2 \\ \vdots \\ \hat{C}\hat{A}^{n-1} \end{bmatrix} = \begin{bmatrix} CT \\ CAT \\ CA^2T \\ \vdots \\ CA^{n-1}T \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} T = \mathcal{O}T$$

Nonsingularity of T implies that $\text{rank}(\hat{\mathcal{O}}) = \text{rank}(\mathcal{O})$ $\square$

The Popov-Belevitch-Hautus Test (PBH rank test)

With the above lemmas, we are ready to show a very important theorem for observability condition, which is equivalent to the Kalman's rank test.

Theorem 3.5. *The pair (A, C) is observable if and only if there exists no $x \neq 0$ such that*

$$Ax = \lambda x, \quad Cx = 0$$

Proof. We prove this by proving its contrapositive statement:

The pair (A, C) is unobservable if and only if there exists $x \neq 0$ such that (1) holds.

$\Longleftarrow$: if there exists $x \neq 0$ such that $Ax = \lambda x, Cx = 0$, then

$$\begin{aligned} CAx &= \lambda Cx = 0, \\ CA^2x &= \lambda CAx = 0, \\ &\vdots \\ CA^{n-1}x &= \lambda CA^{n-2}x = 0 \end{aligned}$$

Therefore, $\mathcal{O}(A, C)x = 0$ for some $x \neq 0$. $\mathcal{O}(A, C)$ cannot be full rank, which implies unobservability.

$\Longrightarrow$: Assume that (A, C) is not observable. Based on Lemma 3.3, we can transform it into the equivalent observable/unobservable realization where

$$\hat{A} = \begin{bmatrix} A_o & 0 \\ A_{21} & A_u \end{bmatrix} \qquad \hat{C} = \begin{bmatrix} C_o & 0 \end{bmatrix}$$

Consider $x \neq 0$ such that

$$A_u x = \lambda x$$

Then

$$\begin{bmatrix} A_o & 0 \\ A_{21} & A_u \end{bmatrix} \begin{pmatrix} 0 \\ x \end{pmatrix} = \lambda \begin{pmatrix} 0 \\ x \end{pmatrix}, \quad \begin{bmatrix} C_o & 0 \end{bmatrix} \begin{pmatrix} 0 \\ x \end{pmatrix} = 0$$

which is

$$T^{-1}AT \begin{pmatrix} 0 \\ x \end{pmatrix} = \lambda \begin{pmatrix} 0 \\ x \end{pmatrix}, \quad CT \begin{pmatrix} 0 \\ x \end{pmatrix} = 0$$

$$AT \begin{pmatrix} 0 \\ x \end{pmatrix} = \lambda T \begin{pmatrix} 0 \\ x \end{pmatrix}, \quad CT \begin{pmatrix} 0 \\ x \end{pmatrix} = 0$$

That is there exists a vector $z = T \begin{pmatrix} 0 \\ x \end{pmatrix}$, such that

$$Az = \lambda z$$

$$Cz = 0$$

□

The PBH test implies that to evaluate observability, we only need to examine the eigenvectors of the system matrix A. If there does not exist any eigenvector x of A such that $Cx = 0$, the system is observable. Recall that the mapping C between states and measurements in our case is simple and special since we directly observe a single state with one sensor. Any row of C is a unit vector $\boldsymbol{e_j}$, with 1 at the jth element if the jth link is observed. For example, if we observe link 1 and link 3 in the toy network in Figure 3.3, we would have $C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

To ensure observability, every eigenvector x of A must satisfy that $Cx \neq 0$. This is equivalent to say, for any eigenvector of A, the elements corresponding to the sensor location cannot all be

zero. We will illustrate this statement by the following example.

Example: Again, let us consider the 3-link merge toy network. If we only observe link 3, the free flow travel speed of link 1 and link 2 is crucial to determine observability as shown in Eq.(3.16). With $C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$,

$$A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 1 & 2 & -3 \end{bmatrix} \text{ is observable, and } A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 1 & -3 \end{bmatrix} \text{ is not observable}$$

Let us examine the first case, where $A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 1 & 2 & -3 \end{bmatrix}$.

The eigenvalues are -3 -2 and -1. The corresponding eigenvectors are $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$, $\begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$ and $\begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}$. The matrix $C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$ will choose the 3rd element and all the eigenvectors have nonzero value. Therefore, the system is observable.

Let us examine the second case, where $A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 1 & -3 \end{bmatrix}$.

The eigenvalues are -3 -1 and -1. The corresponding eigenvectors are $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$, $\begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}$ and $\begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}$. There seems to be no problem here, as the 3rd elements are all non zero. However, we know this case is not observable.

Here, we have repeated eigenvalue -1. There are 2 linearly independent eigenvectors corresponding

to this eigenvalue, and any linear combination of these 2 eigenvectors is also an eigenvector. For example, $\begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$ is an eigenvector. For this eigenvector, the 3rd element is 0 and therefore the PBH test concludes that the system is unobservable.

If we know the sensor location, it is easy to verify the observability using Kalman's test. Unfortunately, that does not give us a sensor deployment strategy. PBH test seems to be a better method for determining sensor location to ensure observability. It is clear at this point that the observability is strongly related to the eigenvectors of the system matrix. Thanks to our simple structure of C, the problem can be further simplified. However, the repeated eigenvalue seems to be problematic. We cannot enumerate all linear combination of the eigenvectors and check all the zero locations. This is infeasible when the dimension is large. This calls for a systematic approach to find a sensor location scheme that guarantees full observability.

3.3 Algebraic Approach for full observability

In the current subsection, we propose a new Algebraic Approach to identify sensor locations that guarantee full observability of an LTI traffic system. Furthermore, we demonstrate the optimality of this approach by proving its ability to achieve the minimal number of sensors. Note that the proposed Algebraic Approach works for any possible realization of the parameters in A. Our theoretical contribution is made possible by exploring the unique algebraic structure of the observability test within the transportation network, where each sensor directly measures its corresponding link density.[2]

We build our Algebraic Approach on the Popov-Belevitch-Hautus Test (PBH rank test). PBH rank test is equivalent to Kalman's rank test and it reveals the connection between observability and eigenvalues as discussed in the last section. Based on the PBH rank test, we restate the PBH rank test in the following equivalent theorem:

[2]This is different from the context studied in the general complex systems literature where a sensor may measure a linear combination of the system states.

Theorem 3.6. *A linear time-invariant system pair (A, C) is observable, if and only if for any eigenvalue λ of A, we have rank* $\left(\begin{bmatrix} A - \lambda I \\ C \end{bmatrix} \right) = n$.

The sensor location problem now reduces to finding an observation matrix C to satisfy the above condition. Yuan et al. (2013) developed an algebraic procedure to minimize the number of driver nodes in complex networks for exact controllability. Due to the mathematical duality of observability and controllability, the result can be applied to find the minimum number of sensors.

However, in complex networks, a driver node has the ability to control multiple components' states simultaneously. When applying this approach to observer design, it means sensors can measure a linear combination of the component states, resulting in rows in C that are not unit vectors. Traffic sensors, on the other hand, can only observe the state of one link, not a combination of states. Thus, each row of the matrix C is a unit vector and this extra constraint makes this problem harder to solve.

The rows of C are unit vectors with ith element to be 1 if the ith link is equipped with a sensor and its density is directly observed. Due to the simple yet special structure of C, we developed an algebraic algorithm to find sensor location that guarantees full observability using the minimal number of sensors. This approach can be summarized in the following 4 steps:

Algebraic Approach

Step 1: Compute the eigenvalues of matrix A.

Step 2: For each eigenvalue λ, choose C s.t. $\begin{bmatrix} A - \lambda I \\ C \end{bmatrix}$ is of full rank. This is done by finding the non-pivot columns in the reduced row echelon form (RREF) of $A - \lambda I$.

Step 3: Combine the necessary sensor locations found for each eigenvalue to finalize the sensor location for full system observability

Step 4: Examine if any sensor is redundant. This is to compute the column rank of $\mathcal{O}$ (Kalman's rank test) with a modified observation matrix. This modification involves removing the ith

row in C for all i, one at a time. If the column rank remains n after excluding a sensor, it will be dropped from the necessary sensor list.

The Algebraic Approach ensures full observability. **Step 2** and **Step 3** return a sensor set that is sufficient for full However, **Step 2** and **Step 3** alone might induce redundant sensors, as shown in the example in Appendix A. **Step 4** will guarantee the sensor locations that are necessary for full observability, without including redundant sensors.

In contrast to optimization model-based methods, which often encounter numerical challenges due to their combinatorial nature, this Algebraic Approach is numerically efficient, making it particularly suitable for tackling large-scale problems.

Theorem 3.7. *The Algebraic Approach gives an optimal sensor placement strategy with* ***minimum number of sensors*** *for full observability.*

Proof. We prove our claim by contradiction. $C \in R^{p\times n}$ is the observation matrix obtained by our Algebraic Approach. Assume there is an optimal observation matrix $C^* \in R^{q\times n}$ $(q < p)$ that uses fewer sensors and yet satisfies the PBH rank test.

Based on **Step 2**, for all λ of A, we have rank $\left(\begin{bmatrix} A-\lambda I \\ C \end{bmatrix}\right) = n$. Also, **Step 4** ensures that removing any row in C will leads to unobservability, i.e., $\exists \lambda_i, \text{rank}\left(\begin{bmatrix} A-\lambda_i I \\ C \end{bmatrix}\right) = n$ and each row in C is linearly independent of the rows in $A-\lambda_i I$. Since C contains orthogonal unit vectors, $A-\lambda_i I$ has rank $n-p$.

For the optimal observation matrix C^*, we also have rank $\left(\begin{bmatrix} A-\lambda_i I \\ C^* \end{bmatrix}\right) = n$. However, the maximum rank of matrix $\begin{bmatrix} A-\lambda_i I \\ C^* \end{bmatrix} = \text{rank}\,(A-\lambda_i I) + \text{rank}\,(C^*) = n-p+q < n$, which leads to a contradiction.

Therefore, we conclude that there cannot exist any C^* that has a fewer dimension than C constructed through steps 1-4, meaning C constructed in our Algebraic Approach is optimal in the

sense of a minimal number of sensors used.

□

The Algebraic Approach works for any realization of the parameters in A without imposing restrictions on the structure of the system matrix. Thus, this approach is applicable for any dynamical system, not limited to transportation network density dynamics. The only requirement is that the rows of observation matrix C consist of unit vectors.

Next, we apply our approach to the following example and demonstrate that caution is needed while translating conclusions from other fields to a domain specific context where problem structure and assumptions may differ.

Consider the system matrix $A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 2 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$.

Step 1

Compute the eigenvalues of A. The eigenvalues are -1, -2 and 0, all with algebraic multiplicity 2.

Step 2

1. For $\lambda = -1$,

$A - \lambda I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$, our sensor placement (choice of C) should ensure $\begin{bmatrix} A - \lambda I \\ C \end{bmatrix}$ to be full column ranked. Since each row in C is a unit vector, we can identify the columns that

need a leading 1 by using the row reduction. To make $\begin{bmatrix} A-\lambda I \\ C \end{bmatrix}$ full ranked, we can construct the row reduced echelon form of RREF $\left(\begin{bmatrix} A-\lambda I \\ C \end{bmatrix} \right)$ since elementary row operations do not change the rank.

RREF $(A-\lambda I) = \begin{bmatrix} \mathbf{1} & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 1 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$, for columns without a leading 1 (column 2 and column 5), we need to include a 1 in matrix C. Thus, C has to include rows $\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$, meaning placing sensors on link 2 and 5.

2. For $\lambda = -2$,

$A - \lambda I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{bmatrix}$ and RREF $(A - \lambda I) = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 2 & -4 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

C has to include rows $\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$.

3. For $\lambda = 0$,

$$A - \lambda I = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 2 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \text{ and RREF } (A - \lambda I) = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

C again has to include rows $\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$.

Step 3

Combining all the necessary rows in C, we obtain the sensor locations that guarantee full observability. In this example, final choice of C is $\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$.

Step 4

Not observing any one of the link 2, 5 and 6 will lead to unobservability (remove the each row on at a time and check the rank of observability matrix). Thus, there is no redundant sensor.

It concludes that for full system observability, sensors are needed on Link 2, 5, and 6 (after the ordering of node indices). Note that the conclusion in Yuan et al. (2013) of using the maximum geometric multiplicity (which is $\mu(\lambda^M) = 2$ in this example) to determine the minimum sensor requirement for full system observability is not applicable here. This is due to the more strict constraint for our observer design, i.e. one sensor can only observe the density of one link.

Remark on computing the eigenvalues

In the Algebraic Approach, the first step is to calculate the eigenvalues of the system matrix A. Due to the special structure of A for traffic network dynamics, the eigenvalues are likely to be the diagonal elements of A. This point is illustrated in the next Chapter section 4.2 where the graphical properties are introduced.

Chapter 4

Graphical Approach for Structural Observability

4.1 Structural observability and inference diagram

Structural observability is an extended concept of observability stemmed from structural controllability, first introduced by Lin (1974). Compared to the exact observability, whose analysis requires knowing the value of matrix A, the structural observability only considers the structure of matrices A and C, meaning whether the matrices' elements are either fixed zeros or independent free parameters.

Definition 4.1 (Structural observability)**.** *The system* (A, C) *is observable for at least some realizations of parameters, without changing the structure (i.e. the location of zero and nonzero elements) of the system matrix and observation model* (A, C)*.*

This concept is particularly relevant to the context of this study for the following reasons. Firstly, for a dynamic traffic system, the exact values of parameters in the system matrix A may change depending on the traffic condition, and one may not be able to always measure the parameters. Secondly, sensor deployment strategies are typically designed for a long-term planning period, during which the exact traffic situation (which determines the exact values of system parameters)

may evolve but the connectivity of the physical road infrastructure that would affect the structure of the system is likely to be stable. Thirdly, even though a structurally observable system may not be observable in theory, for such occasion to happen in a dynamic traffic network is rare (ex. 2 merging links have exactly the same free-flow travel speed).

Structural observability/controllability is found to be closely related to the graph theory (Lin, 1974; Liu et al., 2011, 2013). We adopt the same method in Liu et al. (2013) to construct an **inference diagram** from the system matrix A.

Definition 4.2 (Inference diagram)**.** *Each state variable is considered as a node in the inference diagram and we draw a directed edge from node i pointing to node j if x_j appears in x_i's state update equation ($a_{ij} \neq 0$ in A).*

The edge indicates that the information of state j is captured by state i.

Let us use a 6-link toy network (shown in Figure 4.1) as an example to show how we construct the inference diagram. This toy network, even though is small, has a linear junction, a merge, and a diverge, which are building blocks of a highway network.

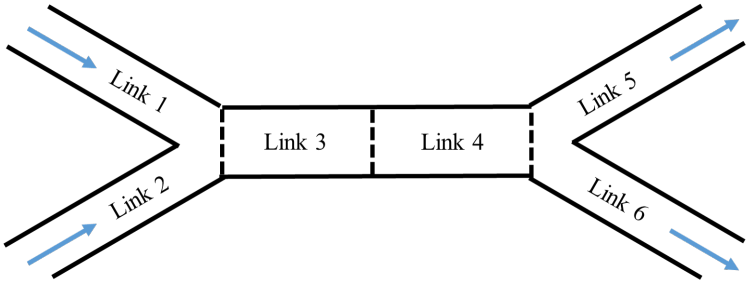


Figure 4.1: 6-link network example

Consider a mode where link 5 is congested and all other links are uncongested. In this case, the downstream links of uncongested links capture information about the upstream links. Link 5 is congested and the flow from link 4 and 5 is determined by supply of link 5, which is a function of link 5 density. The inflow for link 6 is thus also a function of link 5 density. The inference diagram corresponding to this mode, and the matrix A, with the (i, j)th element denoted as a_{ij}, are plotted in Figure 4.2. Apparently, different traffic modes can result in different matrix A and inference

diagram, as demonstrated by the examples given in subsection 4.4.1.

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 0 \\ 0 & 0 & a_{43} & 0 & a_{45} & 0 \\ 0 & 0 & 0 & 0 & a_{55} & 0 \\ 0 & 0 & 0 & 0 & a_{65} & 0 \end{bmatrix}$$

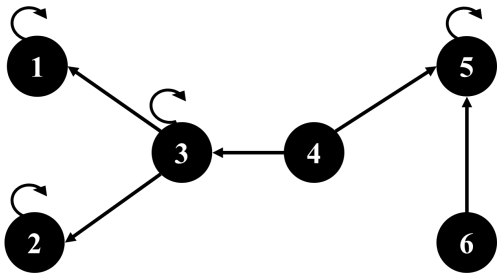


Figure 4.2: Inference diagram for the 6-link network

Definition 4.3 (Self-edge)**.** *A self-edge is an edge starting from a node and pointing to the same node.*

In the above example, node 1,2,3,5 have a self-edge. In some literature, self-edge is called a self-loop.

Definition 4.4 (Reachable node)**.** *A reachable node in the Inference Diagram is a node such that there exists a sequence of adjacent nodes (i.e. walk) from a given observed node (such as a sensor location) to it. A k-step reachable node is a node that is reached from an observed node by a walk with k edges.*

Using Figure 4.2 as an example, node 1 is reachable from node 4. It is 2-step reachable from node 4 through edge (4,3) → edge (3,1). Also, it is 3-step reachable from node 4 through edge (4,3) →edge (3,3) → edge (3,3), or edge (4,3) → edge (3,1) → edge (1,1).

4.2 Notes on computation of the Algebraic Approach exploiting the network structure

In the Algebraic Approach, the first step is to calculate the eigenvalues of the system matrix A based on the special structure of A for traffic network dynamics. The following assumption is made for the inference diagram we construct for the system dynamics:

Assumption: The inference diagram of the traffic density dynamics is an acyclic graph.

This assumption holds true if the information does not flow in one direction that forms a loop. If the physical network does not contain any cycle, this assumption will be true. Even if it contains cycles, the information is unlikely to flow in one direction (all links are free-flow or congested), which is also a case that we are not very interested in.

With this assumption, our Algebraic Approach has some advantageous properties.

Proposition 1. *Since the inference diagram is acyclic, there exists a topological ordering such that the matrix A is lower-triangular. Moreover, the eigenvalues are simply the diagonal elements of A.*

A (reverse) topological ordering is a vertex ordering for a directed acyclic graph such that for every directed edge (i, j) from vertex i to vertex j, i comes after j in the ordering. After the topological sort, $a_{ij} = 0$ if $i < j$ and A is lower-triangular. This is for analytical convenience only, as the eigenvalues are the diagonal elements for an upper or lower triangular matrix.

We implement the (reverse) topological ordering and obtain matrix A as a lower triangular matrix. For the 6-link network example in Figure 4.1, the reordered graph and its system matrix A is shown in Figure 4.3. The node 4 and 5 are switched to satisfy the ordering requirement. Matrix A is now an lower triangular matrix.

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{44} & 0 & 0 \\ 0 & 0 & a_{53} & a_{54} & 0 & 0 \\ 0 & 0 & 0 & a_{64} & 0 & 0 \end{bmatrix}$$

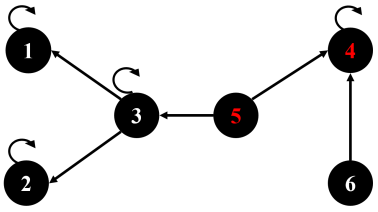


Figure 4.3: 6-link network example: after the topological ordering

4.3 A Graphical Approach to identify observable components

The existence of self-edges in an inference diagram (see Proposition 2 and its proof) is a special feature in the dynamic traffic network context, which may not be true for general complex networks. The existence of self-edges greatly increases observability, as we shall see later.

Proposition 2. *In an inference diagram of a traffic system, any node with an incoming edge also*

has a self-edge.

Proof. A node with an incoming edge implies the flow between this node and its neighbor(s) is determined by this node's own density. Due to the flow conservation, this flow will also act on the node itself, causing a self-edge. □

As an example, in Figure 4.2, the nonzero entry a_{43} of system matrix A indicates the change of density of link 4 is a function of density of link 3. This implies the flow between link 4 and 3 is determined by the density of link 3. This flow also affects link 3 itself, leading to a nonzero entry a_{33} in A. Consequently, we observe a self-edge in the inference diagram, reflecting the influence of link 3's density on its own dynamics.

We are now ready to show the following important theorem to identify observable states (network link densities):

Theorem 4.5. *Based on the inference diagram, any reachable nodes from observed nodes are also (structurally) observable. Moreover, if each node has at most one outgoing edge (excluding self-edge), the observability is exact.*

Proof. Each row of observation matrix C is a unit vector. Assume the ith row of C is unit vector $\boldsymbol{e}_{\boldsymbol{k}}^T$ where all the elements are zero except for the kth element. The ith sensor is used to observe the kth state. Consider the observability matrix $\mathcal{O}_i = \begin{bmatrix} C_{i,:} \\ C_{i,:}A \\ C_{i,:}A^2 \\ \vdots \\ C_{i,:}A^{n-1} \end{bmatrix}$ where $C_{i,:}$ is the ith row of C (i.e., $\boldsymbol{e}_{\boldsymbol{k}}^T$). Since multiplying $C_{i,:}$ to a matrix is simply selecting the kth row of the matrix, the rows of matrix $\mathcal{O}_i$ contain kth row of $A, A^2, ..., A^{n-1}$, leading to a matrix whose columns are $\mathbf{0}$ for unreachable nodes starting from node k. In the case where each node has at most one outgoing edge, the leading nonzero element positions of the nonzero columns are different. Therefore, the nonzero columns are linearly independent.

In the general case, due to the existence of the self-edges, entries under the first nonzero entry will not vanish for each column. The structural assumption on our system leads to linearly independent nonzero columns (more discussion on this is provided in the remark below). Thus, the nonzero linearly independent columns of $\mathcal{O}_k$ form a full column rank observability matrix, representing an observable subsystem consisting of the observed node and its reachable nodes. This is true for any k. The observed nodes and the consequent reachable nodes are the observable components. For unreachable nodes, we can consider them as if they are not part of the system and they will not affect the observability of the reachable nodes. □

Remark on structural assumption: Readers familiar with the complex systems literature may recall that the structural assumption leading to the conclusion that the nonzero columns are linearly independent was based on the assumption that all entries in matrix A are independent variables. This is certainly not the case in the traffic dynamic network case. For example, consider the problem shown in Figure 4.2, $a_{11} = -\frac{v_1}{l_1}$ and $a_{31} = \frac{v_1}{l_3}$ where v_1 is the free-flow travel speed on link 1 and l_1, l_3 are the link lengths of link 1 and link 3, respectively. Regardless of the choice of the free parameter v_1, entries a_{11} and a_{31} are dependent: $a_{11} = -\frac{l_3}{l_1}a_{31}$. With these dependent elements in A, can we still assume linear independence of the nonzero columns in the observability matrix? We answer this question next.

Proposition 3. *We denote the inference diagram as graph $G(V, E)$. $\forall (i_1, j_1), (i_2, j_2) \in E$, the corresponding entries in A, $a_{i_1 j_1}$ and $a_{i_2 j_2}$ are independent parameters given $j_1 \neq j_2$.*

Proof. For nonzero parameters $a_{i_1 j_1}$ and $a_{i_2 j_2}$, the flows between (i_1, j_1) and (i_2, j_2) are functions of the link densities of j_1 and j_2 respectively. Consequently, the parameters $a_{i_1 j_1}$ and $a_{i_2 j_2}$ are solely dependent on the fundamental diagrams of different links j_1 and j_2, each possessing independent characteristics such as free-flow travel speed. Thus, the two parameters $a_{i_1 j_1}$ and $a_{i_1 j_2}$ are independent. □

To prove Theorem 4.5, we claimed that the structural assumption will make the column vectors linearly independent in the observability matrix $\mathcal{O}$. It is evident that if all the nonzero leading

entries of the columns are positioned in distinct rows, these columns will be linearly independent. The problem remains to verify for the case where the nonzero leading entries of two columns reside in the same row, and the columns can still be linearly independent. The (i,j)th entry of A^k is the sum of the products of all weights of all walks from node i to node j of length exactly k. For each of the two columns (denoted by j_1 and j_2) in $\mathcal{O}$ to be verified, assuming they are k-step reachable, the leading nonzero entries is the product of all weights of the k-length walk from the observed node (denoted by i) to the column node. Thus, **the leading nonzero entries for column j_1 and j_2 are functions of a_{ij_1} and a_{ij_2}** respectively, which are independent parameters based on Proposition 2. The entries under the leading nonzero entries are the products of the weights in $(k+1)$-length, which are functions of $a_{ij_1} \cdot a_{j_1 j_1}$ and $a_{ij_2} \cdot a_{j_2 j_2}$. Since Proposition 2 implies $a_{jj} \neq 0$ if $a_{ij} \neq 0$, moreover, a_{ij} and a_{jj} are dependent parameters, scaled by a factor, **the two entries under the leading nonzero entires of column j_1 and j_2 are functions of $a_{ij_1}^2$ and $a_{ij_2}^2$**. Therefore, there exist choices of parameters a_{ij_1} and a_{ij_2}, such that columns j_1 and j_2 are linearly independent.

Remark on the importance of self-edges: In the example of 6-link network with the inference diagram of Figure 4.2, if a sensor is placed on link 4, the observability matrix is then in the following form, where we use $\times$ to indicate a nonzero entry:

$$\begin{bmatrix} 0 & 0 & 0 & \times & 0 & 0 \\ 0 & 0 & \times & 0 & \times & 0 \\ \times & \times & \times & 0 & \times & 0 \\ \times & \times & \times & 0 & \times & 0 \\ \times & \times & \times & 0 & \times & 0 \\ \times & \times & \times & 0 & \times & 0 \end{bmatrix} \begin{array}{l} \longleftarrow C_{i,:} \text{ (note: link 4 is directly observed by the sensor)} \\ \longleftarrow C_{i,:}A \text{ (links 3 and 5 are 1-step reachable from link 4)} \\ \longleftarrow C_{i,:}A^2 \text{ (links 1, 2, 3 and 5 are 2-step reachable from link 4)} \\ \longleftarrow C_{i,:}A^3 \\ \longleftarrow C_{i,:}A^4 \\ \longleftarrow C_{i,:}A^5 \end{array}$$

The structural observability assumption leads to the conclusion that the nonzero columns are linearly independent. Thus, the observability matrix of the subsystem consisting of link 1,2,3,4, and

5 has full column rank. This subsystem will be observable. Now imagine if the inference diagram did not have the self-edges. All the nodes in the graph that are previously k-step reachable are no longer $(k+1)$-step reachable, leading to the following observability matrix:

$$\begin{bmatrix} 0 & 0 & 0 & \times & 0 & 0 \\ 0 & 0 & \times & 0 & \times & 0 \\ \times & \times & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} \longleftarrow C_{i,:} \\ \longleftarrow C_{i,:}A \\ \longleftarrow C_{i,:}A^2 \\ \longleftarrow C_{i,:}A^3 \\ \longleftarrow C_{i,:}A^4 \\ \longleftarrow C_{i,:}A^5 \end{matrix}$$

where the first 5 columns can never be linearly independent no matter what values the non-zero elements take. This example shows the importance of the self-edges in increasing the observability.

In summary, Theorem 4.5 provides an efficient way to identify the observable subspace of the states - the number of reachable nodes in the inference diagram measures the observability, and the reachable nodes correspond to the observable states. This is important since it provides an explicit quantification of the quality of observability (unlike other metrics such as observability Gramian). It enables us to evaluate a sensor placement strategy for a partially observable system. A natural follow-up result is that to fully observe all link densities, traffic sensors should be located at links corresponding to the source nodes (nodes that have only outgoing edges) in the inference diagram. The result is consistent with the graph-theoretical approaches developed in Liu et al. (2011) and Liu et al. (2013): observing the source nodes is indeed the solution applying the maximum matching algorithm. Note that the graph-theoretical approaches developed in Liu et al. (2011) and Liu et al. (2013) find sensor location for full observability only. Our finding on the partial observability property is blessed by the special graphical properties of the traffic density dynamics. To our knowledge, this is the first study to reveal the partial observability property of the dynamic traffic system in the literature.

4.4 A mathematical programming model maximizing number of observable components under various traffic conditions

In the above sections, we have established algebraic and graphical approaches for observability analysis of an LTI traffic system. However, the dynamics of traffic network density can only be modeled by an LTI dynamical system for short periods of time with consistent traffic conditions. A single set of linear equations is inadequate for capturing the evolving nature of traffic conditions within networks. In this section, we integrate observability and mode switching concepts in a mathematical programming modeling framework to maximize observable components under various traffic conditions.

Let us first explain the mode switching concept. Due to the min function to determine the actual flow in Eq (3.5), (3.6) and (3.7), the system is nonlinear. However, with a triangular fundamental diagram, the whole system can be considered as a piecewise linear system, switching within different linear models, which we call **modes**. Using Fig 3.2a as an example, when the two links are free-flow with length 1 and the demand of link 1 is less than supply of link 2, the system dynamics is described as:

$$\begin{bmatrix} \dot{k_1} \\ \dot{k_2} \end{bmatrix} = \begin{bmatrix} -v_1 & 0 \\ v_1 & 0 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} + \begin{bmatrix} f_1 \\ -g_2 \end{bmatrix} \tag{4.1}$$

When link 2 becomes congested, the supply of link 2 is less than the demand of link 1, the system dynamics then evolves to:

$$\begin{bmatrix} \dot{k_1} \\ \dot{k_2} \end{bmatrix} = \begin{bmatrix} 0 & w_2 \\ 0 & -w_2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} + \begin{bmatrix} f_1 - w_2 k_{j,2} \\ -g_2 + w_2 k_{j,2} \end{bmatrix} \tag{4.2}$$

In a more concise form, the above system with two modes can be expressed as a piecewise affine system:

$$\dot{\boldsymbol{x}}(t) = A(\lambda_t)\boldsymbol{x}(t) + \boldsymbol{b}(\lambda_t) + \boldsymbol{u}(t) \tag{4.3}$$

where $\boldsymbol{x}$ is the continuous state variables, $\boldsymbol{u}(t) = \begin{bmatrix} f_1 & -g_2 \end{bmatrix}^T$ is the boundary flow vector. A and

$\boldsymbol{b}$ are mode specific, which switch among K different modes $\lambda_t \in \{1, 2, \cdots, K\}$. In **??**, we show a bigger example of mode switching in the 6-link network.

Now we are ready to present our sensor optimization model that considers a wide spectrum of traffic modes. Let the binary decision variable z_i control the sensor location and the auxiliary binary variable $x_{k,i}$ indicate the inferred observability.

$$\text{let } z_i = \begin{cases} 1, \text{if a sensor is put on link } i, \\ 0, \text{otherwise} \end{cases}, \; x_{k,i} = \begin{cases} 1, \text{if link } i \text{ is observable in mode } k, \\ 0, \text{if link } i \text{ is not observable in mode } k \end{cases}$$

where $i = 1, 2, ..., n$ and $k = 1, 2, ..., K$, K is the total number of different modes of the system under consideration. Note that in a typical decision environment, K does not need to include all possible modes; one may only include representative modes that are more likely to appear. The importance of each mode k may be expressed by a non-negative weight factor w_k. In practice, this can refer to the frequency of occurrence of a mode. A practical way of obtaining major modes and their occurrence rates is introduced in the case study in Section 4.5.

We then have the following mixed-integer program formulation:

$$\max_{z,x} \sum_{k=1}^{K} w_k \sum_{i=1}^{n} x_{k,i} \tag{4.4a}$$

$$s.t. \sum_{i} z_i = p \tag{4.4b}$$

$$x_{k,i} \leq \sum_{(i^-,i) \in \delta_k^-(i)} x_{k,i^-} + z_i, \forall k, i \tag{4.4c}$$

$$z_i \in \{0, 1\}, \forall i \tag{4.4d}$$

$$x_{k,i} \leq 1, \forall k, i \tag{4.4e}$$

where $\delta_k^-(i)$ is the set of incoming edges of node i (here, the self-edges are **not** considered as incoming edges) in the inference diagram of the k-th mode. The objective function (4.4a) maximizes the average number of observable states across all possible modes, in a weighted sense. Constraint (4.4b) specifies the total number of sensors to be placed where p is pre-specified. Constraints

(4.4c) impose that a node is not observable if it is not directly observed or it has no observable predecessor nodes in the inference diagram. One might notice that x is not enforced to be binary in the formulation. This is not a mistake as the objective function and constraints will naturally push x to be binary. With a non-negative weight $w_{k,i}$, $x_{k,i}$ will be binary given constraints (4.4c) and (4.4e). This could provide computational advantages because handling integer constraints typically adds burden to large-scale linear programming problems.

It is beneficial to recognize that the mathematical model (4.4) is indeed a stochastic program when the weight factor w_k is set to be probability of mode k. Recall a general two-stage stochastic nonlinear program (Ruszczyński and Shapiro, 2003)

$$\begin{aligned} \min_{x} \quad & L(x) + E_{\xi}[Q(x,\xi)] \\ \text{s.t.} \quad & g_i^1(x) \leq 0 \\ \text{with} \quad & \\ Q(x,\xi) = \min_{y} \quad & f(y;x,\xi) \\ \text{s.t.} \quad & g_i^2(y;x,\xi) \leq 0 \end{aligned} \tag{4.5}$$

Typically, the goal can be interpreted as to make a planning decision that minimizes the total costs including the current and the expected future costs. The first stage planning decision x with cost $L(x)$ has to be made before any possible outcome of the random parameter ξ becomes certain. In the second stage, the actual realization of ξ becomes known and a recourse decision y can be taken. $Q(x,\xi)$ is the objective function of the second stage problem given a particular choice of x and a realization of ξ.

In the current context, each mode k corresponds to an uncertain scenario with a probability of w_k. The sensor placement decision z corresponds to the planning decision. Note we do not impose a sensor deployment cost, so there is no $L(x)$. Instead, we limit the budget of sensor deployment by imposing a hard constraint on the total number of sensors. The observability of links x corresponds to the operational decisions in a classic stochastic programming (SP) framework. The second-stage subproblem $Q(x,\xi) = \min_y f(y;x,\xi)$ in the generic formulation (4.5) becomes

$Q(z,k) = \max_x \sum_{i=1}^{n} x_{k,i}$ subject to the feasibility constraints of x (4.4c and 4.4e) in the current context.

We make an association of our model with classic SP framework so that one can exploit the very rich SP literature such as the various decomposition approaches (Carøe and Schultz, 1999; Collado et al., 2012; Rockafellar and Wets, 1991) developed for solving large-scale SP models if needed, even though numerical treatment is not an emphasis of this study.

4.4.1 A numerical example of mode switching and the MILP model for sensor location

In this numerical example, we use the aforementioned 6-link network example as shown in Figure 4.1. The mode-switching is illustrated in this example. The system dynamics are shown in the discrete time-invariant system derived from the LQM. There are two modes in consideration, with mode importance weights w_1 and w_2:

Mode 1: all links are uncongested and the system is operating at free-flow condition:

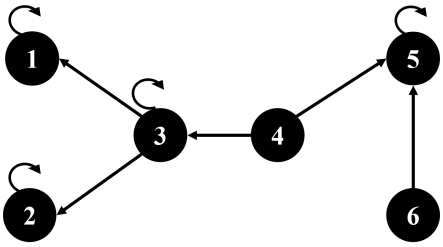
$$\begin{bmatrix} k_1(t+1) \\ k_2(t+1) \\ k_3(t+1) \\ k_4(t+1) \\ k_5(t+1) \\ k_6(t+1) \end{bmatrix} = \begin{bmatrix} 1-v_1\frac{\Delta T}{l_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-v_2\frac{\Delta T}{l_2} & 0 & 0 & 0 & 0 \\ v_1\frac{\Delta T}{l_3} & v_2\frac{\Delta T}{l_3} & 1-v_3\frac{\Delta T}{l_3} & 0 & 0 & 0 \\ 0 & 0 & v_3\frac{\Delta T}{l_4} & 1-v_4\frac{\Delta T}{l_4} & 0 & 0 \\ 0 & 0 & 0 & \xi_{4\to5}(t)v_4\frac{\Delta T}{l_5} & 1 & 0 \\ 0 & 0 & 0 & \xi_{4\to6}(t)v_4\frac{\Delta T}{l_6} & 0 & 1 \end{bmatrix} \times \begin{bmatrix} k_1(t) \\ k_2(t) \\ k_3(t) \\ k_4(t) \\ k_5(t) \\ k_6(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} f_1(t)/l_1 \\ f_2(t)/l_2 \\ 0 \\ 0 \\ -g_5(t)/l_5 \\ -g_6(t)/l_6 \end{bmatrix} \tag{4.6}$$

Mode 2: link 5 is congested and becomes the bottleneck. The flow from link 4 to link 5 is now dominated by the supply of link 5, which is $(k_{j,5} - k_5)w_5$. The outflux of link 4 is $g_4 = s_5/\xi_{4\to5}(t)$.

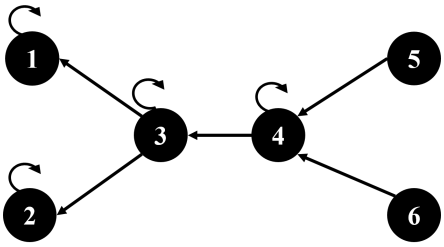
The influx of link 6 is $\xi_{4\to6}(t)g_4$:

$$\begin{bmatrix} k_1(t+1) \\ k_2(t+1) \\ k_3(t+1) \\ k_4(t+1) \\ k_5(t+1) \\ k_6(t+1) \end{bmatrix} = \begin{bmatrix} 1-v_1\frac{\Delta T}{l_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-v_2\frac{\Delta T}{l_2} & 0 & 0 & 0 & 0 \\ v_1\frac{\Delta T}{l_3} & v_2\frac{\Delta T}{l_3} & 1-v_3\frac{\Delta T}{l_3} & 0 & 0 & 0 \\ 0 & 0 & v_3\frac{\Delta T}{l_4} & 1 & \frac{w_5\Delta T}{\xi_{4\to5}(t)l_4} & 0 \\ 0 & 0 & 0 & 0 & 1-\frac{w_5\Delta T}{l_5} & 0 \\ 0 & 0 & 0 & 0 & -\frac{w_5\Delta T}{l_6}\frac{\xi_{4\to6}(t)}{\xi_{4\to5}(t)} & 1 \end{bmatrix} \times \begin{bmatrix} k_1(t) \\ k_2(t) \\ k_3(t) \\ k_4(t) \\ k_5(t) \\ k_6(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{k_{j,5}w_5\Delta T}{\xi_{4\to5}(t)l_4} \\ \frac{k_{j,5}w_5\Delta T}{l_5} \\ \frac{k_{j,5}w_5\Delta T}{l_6}\frac{\xi_{4\to6}(t)}{\xi_{4\to5}(t)} \end{bmatrix} + \begin{bmatrix} f_1(t)/l_1 \\ f_2(t)/l_2 \\ 0 \\ 0 \\ -g_5(t)/l_5 \\ -g_6(t)/l_6 \end{bmatrix} \tag{4.7}$$

Figure 4.4 compares the inference diagram for the two modes. Observing links 1,2,3,4 leads to 4 links observable in mode 1 and 5 links observable in mode 2. Note that when we construct the inference diagram, a self-edge may only be added if the diagonal element of the continuous system matrix is nonzero. Here, since Eqs. (4.6) and (4.7) correspond to the discretized system matrix, if the diagonal element is 1 (the corresponding continuous system matrix will have a zero diagonal element), self-edges will not be added.



(a) Mode 1



(b) Mode 2

Figure 4.4: Inference diagram for mode 1&2

Integer programming formulation

Here we want to locate 4 sensors on the 6-link network to maximize observability, thus we have $n = 6, p = 4$.

$$\text{let } z_i = \begin{cases} 1, \text{if a sensor is put on link } i, \\ 0, \text{otherwise} \end{cases}, \; x_{k,i} = \begin{cases} 1, \text{if link } i \text{ is observable in mode } k, \\ 0, \text{if link } i \text{ is not observable in mode } k \end{cases}$$

where $i = 1, 2, ..., 6$ and $k = 1, 2$ the total number of modes in this case is 2, for simplicity. Let w_1 and w_2 denote the mode frequencies. The observability maximization problem is formulated as:

$$\begin{aligned}
\max \quad & w_1(x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} + x_{1,5} + x_{1,6}) + w_2(x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} + x_{2,5} + x_{2,6}) \\
\text{s.t.} \quad & z_1 + z_2 + z_3 + z_4 + z_5 + z_6 = 4 \\
& \left.\begin{aligned} & x_{1,1} \le x_{1,3} + z_1 \\ & x_{1,2} \le x_{1,3} + z_2 \\ & x_{1,3} \le x_{1,4} + z_3 \\ & x_{1,4} \le z_4 \\ & x_{1,5} \le x_{1,4} + x_{1,6} + z_5 \\ & x_{1,6} \le z_6 \end{aligned}\right\} \textbf{mode 1} \\
& \left.\begin{aligned} & x_{2,1} \le x_{2,3} + z_1 \\ & x_{2,2} \le x_{2,3} + z_2 \\ & x_{2,3} \le x_{2,4} + z_3 \\ & x_{2,4} \le x_{2,5} + x_{2,6} + z_4 \\ & x_{2,5} \le z_5 \\ & x_{2,6} \le z_6 \end{aligned}\right\} \textbf{mode 2} \\
& z_1, z_2, z_3, z_4, z_5, z_6 \in \{0, 1\} \\
& x_{k,i} \le 1, \forall k \in \{1, 2\}, i \in \{1, \cdots, 6\}
\end{aligned}$$

4.5 Mode importance: a prior knowledge that can be obtained from other data sources

To formulate the optimization in 4.4, the mode importance w_k is needed. In this section, we present a case study conducted using real traffic data collected on the I80 East Davis - Sacramento highway corridor, which experiences regular afternoon peak congestion. This case study is included to demonstrate how one might implement our model in practice, including how to prepare information regarding traffic modes and their weights that is required in the sensor location model.

The highway network consists of 21 mainstream links and 11 on- and off-ramps that form merging and diverging junctions, as depicted in Figure 4.5. The 10.6-mile long mainstream is segmented into links that are separated using the orange pins on the map. We collected real-time travel speed information through the Google Maps API every 10 minutes from 2 PM to 7 PM, Monday through Thursday, throughout the months of April to June 2023. The speed profiles for three days in June are visualized in Figure 4.6. We observe a recurring pattern of congestion propagation every day and the data serves as the prior knowledge of the various traffic conditions. The weight factor w_k is obtained based on the occurrence frequency of the conditions. From all the traffic modes, we choose the ones that occurred more than 10 times. A total of 47 different modes are selected, which takes approximately 44% of all-time occurrence. Note that even though we have link speed information as ground truth, it does not directly translate to link density information as there is not a one-to-one relationship between speed and density under the uncongested region. Therefore, the question of observing link density is not trivial.

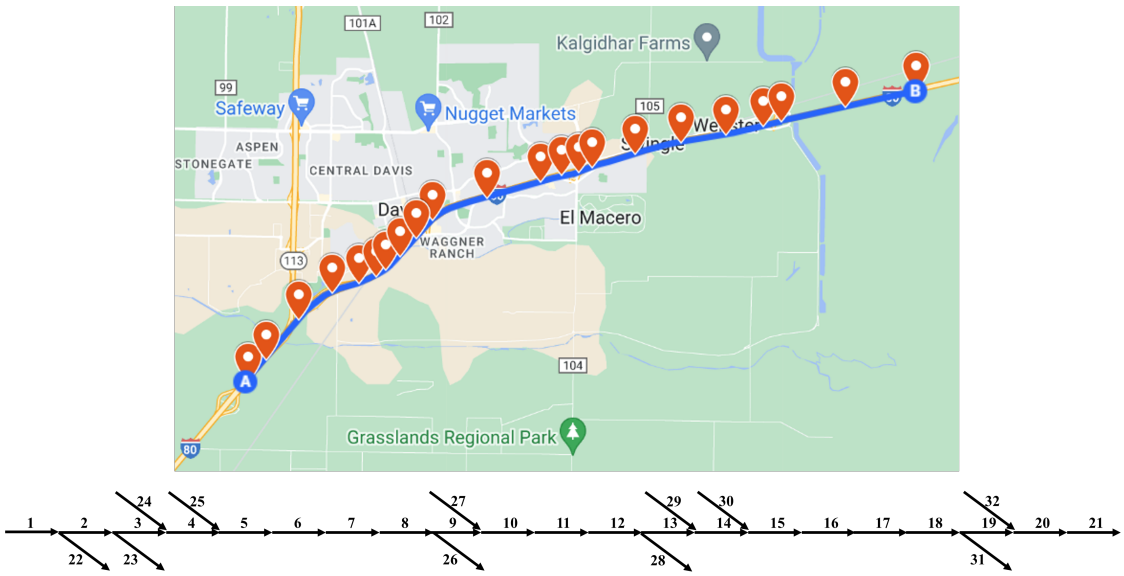


Figure 4.5: I80 East Davis-Sacramento

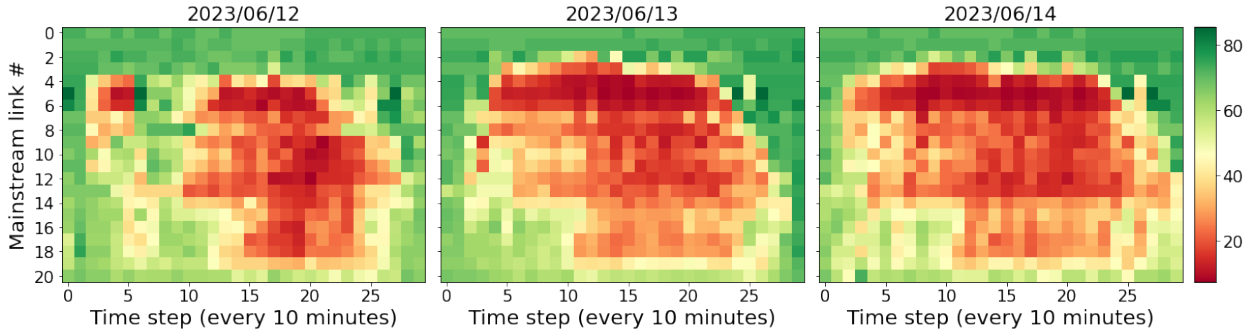


Figure 4.6: Speed profile of I80 East evolution in time

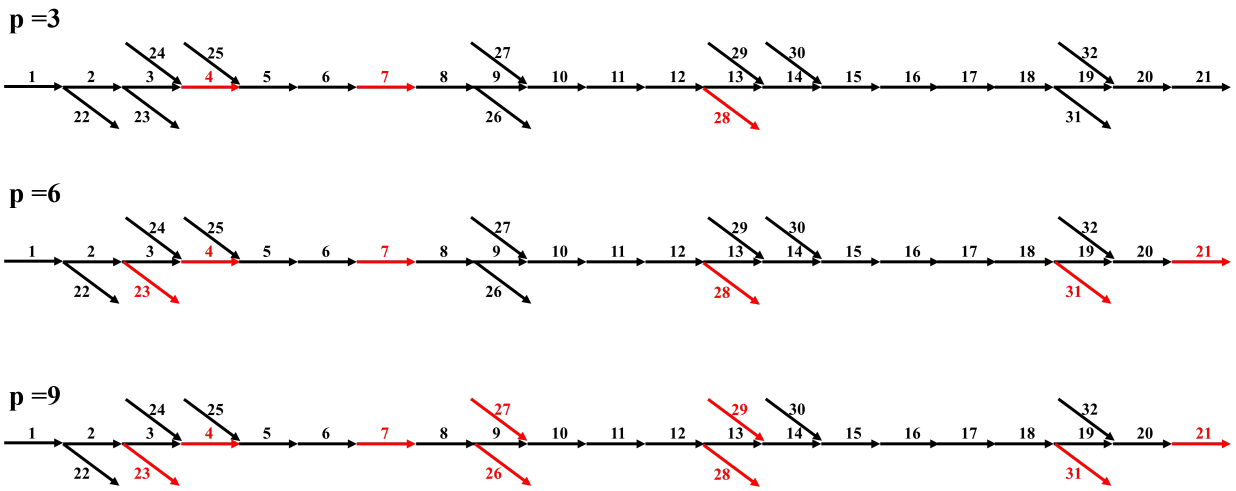


Figure 4.7: Optimal sensor locations for varying numbers of sensors to be deployed

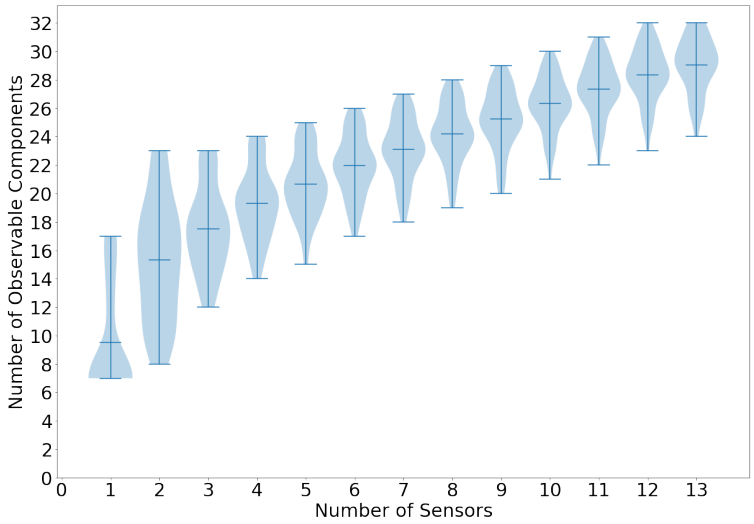


Figure 4.8: Violin plot for observability under different numbers of sensor to be deployed

We solve the optimization problem in (4.4) to obtain the best sensor location solutions under various p. As an illustration, the optimal sensor locations for $p = 3, 6, 9$ are plotted in Figure 4.7. Links 4 and 7 are of highest priority since there is usually free-flow traffic before link 4 and congested traffic after link 7. Off-ramps are also prioritized since their information is not captured by the mainstream when the main congestion source is the downstream mainstream segment.

Note that while preparing modes as an input to the optimization model (4.4), we take a quite

coarse approach to translate speed to density under the uncongested regime – for an uncongested link (a link where speed is within 90% of its free-flow speed), we use half of its critical density to determine the supply and demand relation between this link and its immediate downstream links. The reason we take such a coarse approach is that before sensors are placed, we are not supposed to have much precise information about the traffic conditions. One can see from the optimal sensor location solutions that even though the information of modes entered to the optimization model is prepared in a quite coarse manner, the sensor placement strategies made good sense. For example, no sensor is suggested for links 1, 2, 3, which is consistent with the fact that they are never congested based on the Google Maps data.

Figure 4.8 reports the performance of the optimal sensor location strategies (obtained under different p) across various traffic modes. Each violin corresponds to a given p. The shape of violin shows the distribution of the number of observed links across different modes, together with the max, min, and the average number of observability. Based on the results shown in Figure 4.8, we have the following findings:

- When sensors are strategically placed, they can lead to more inferred information. For example, placing 13 sensors in the 32-link network could lead to inference of link densities of about 28 links on average.
- More sensors could not only improve the average observability, but also reduce the risk of not observing many links (indicated by the narrower observability range and heavier skew towards larger observability as the number of sensors increases).
- The marginal benefit on observability gained from additional sensors slows down as more sensors are deployed, indicating that strategic budgeting may be considered along with strategic location problem.

Notably, when only one sensor can be deployed, the optimal sensor will at most observe 17 links and at least observe 7 links among the 47 different modes. One should be mindful that the knowledge of the exact mode itself contains a significant amount of information. Without such information, a

single sensor observing multiple segment downstream may be questionable, particularly considering traffic density dynamics is subject to noise.

Chapter 5

Traffic Observability and State Estimation

This chapter includes numerical experiments on different networks, with a purpose of showcasing the applicability of the proposed observability measures and sensor placement approach under different network settings.

Traffic state estimation (TSE) refers to the process of the inference of traffic state variables using partially observed traffic data. The state variables can be traffic flow, density, speed, etc. Observability is the property of uniquely determining the state variables from the observation. Observability is a prerequisite for effective state estimation. If a system is not observable, it is impossible to uniquely determine the system's state from its outputs, making reliable state estimation impossible.

In this chapter, the observability is quantified as the average number of observable components across all possible modes, computed in Eq.(4.4a). We aim to show that the sensor location with our observability maximization will lead to better state estimation quality.

In Section 5.1, the TSE algorithm Interacting Multiple Model Filtering is introduced. In Section 5.2, a small toy network is used to illustrate the idea of mode switching as the system gets congested. This example also showcases that better observability leads to better estimation quality

even when the actual traffic dynamics deviate from the LQM model on which the observability analysis is based. In Section 5.3, we study the observability in a larger network to further investigate the benefit of higher observability considering various possible modes for state estimation.

5.1 Traffic state estimation

Traffic state estimation plays an important role in traffic operations and planning. It is the primary step to understand the traffic system before any control (such as ramp metering and dynamic pricing) can be implemented.

The highway state estimation has been studied for decades and various estimation methods have been proposed. Seo et al. (2017) examines the highway state estimation methods in 3 categories: model-driven, data-driven, and streaming-data-driven. The model-driven approaches are based on traffic flow models describing traffic dynamics. The parameters in the traffic flow model need to be calibrated before estimation. The data-driven approach relies extensively on historical data. Recent popular machine learning models belong to this category. It does not require prior knowledge of the physical model. However, it requires a large amount of historical data.

Among the many traffic state estimation methodologies (Seo et al., 2017), we adopt the Interacting Multiple Model (IMM) filtering, which has demonstrated its effectiveness in traffic state estimation, both in urban and freeway traffic systems (Panda et al., 2019; Thai and Bayen, 2014; Wang and Work, 2014; Zhang and Mao, 2015). The IMM filtering is a powerful yet cost-effective filtering algorithm with the capability of estimating the state of a dynamic system that may switch across different traffic modes, which makes it a suitable choice for our needs. To implement the IMM filtering, one needs to describe the stochastic process to be estimated. For the estimation purpose, the stochastic process is described following stochastic LQM to incorporate process noise and measurement noise:

$$\boldsymbol{x}_{t+1} = A_{\lambda_t}\boldsymbol{x}_t + \boldsymbol{b}_{\lambda_t} + \boldsymbol{u}_t + \boldsymbol{\omega}_t \tag{5.1}$$

$$\boldsymbol{y}_t = C\boldsymbol{x}_t + \boldsymbol{\nu}_t \tag{5.2}$$

where $\boldsymbol{x}$ is the discretized state variables, whose dynamics is switching among a number of K different modes $\lambda_t \in \{1, 2, \cdots, K\}$. $\boldsymbol{\omega}_t$ and $\boldsymbol{\nu}_t$ are zero-mean Gaussian noises with known covariances Q and R. This model is similar to the one in Thai and Bayen (2014).

In IMM filtering, at each time step, the model is no longer in a single true mode; instead, it is a probabilistic combination of all possible modes (K different modes). This filtering algorithm dynamically recalibrates and updates the likelihood of different modes for the current state based on the observed data. The filtering operates in parallel, employing Kalman filters for each mode, and the resulting individual estimates are combined to produce the final estimate.

Interacting Multiple Model filtering overview

The Interacting Multiple Model (IMM) algorithm is a widely used estimation technique in systems with multiple dynamic models. It is effective in tracking system states where the system can switch between different modes, each represented by a different model. The IMM filtering operates by running multiple filters in parallel, each corresponding to a different mode, and then combining their outputs based on probabilistic mode weights. The nature of this estimation algorithm fits well with our mode-switching traffic dynamics.

The IMM algorithm operates by maintaining a set of K models, each represented by a Kalman filter. **To avoid notation confusion with the Kalman Gain K, we denote the total number of modes in consideration as N_m.**

The state estimate at time t is given by the following distribution:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{i=1}^{N_m} \mu_t^i \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t}^i, \Sigma_{t|t}^i)$$

where:

- $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is the posterior distribution of the system state $\mathbf{x}_t$ given the observations $\mathbf{y}_{1:t}$ up to time t. It represents the probability of the state being $\mathbf{x}_t$ given all the observations so far.

- $\mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}^i_{t|t}, \Sigma^i_{t|t})$ is a multivariate Gaussian distribution with mean $\hat{\mathbf{x}}^i_{t|t}$ and covariance $\Sigma^i_{t|t}$ for the ith mode. This distribution describes how likely different values of $\mathbf{x}_t$ are, assuming the ith mode is the true mode.

- $\mu^i_t = P(\lambda_t = i|\mathbf{y}_{1:t})$ is the posterior mode probability, which represents the probability that the ith mode $\lambda_t = i$ is the true mode at time t, given the observations $\mathbf{y}_{1:t}$.

- The sum $\sum_{i=1}^{N_m}$ over N_m modes implies that the overall state distribution is approximated by a weighted sum of the individual Gaussian distributions for each mode, with the weights being the mode probabilities μ^i_t.

The overall posterior mean $\hat{\mathbf{x}}_{t|t}$ and covariance $\Sigma_{t|t}$ can be calculated as follows:

$$\hat{\mathbf{x}}_{t|t} = \sum_{i=1}^{N_m} \mu^i_t \hat{\mathbf{x}}^i_{t|t}$$

$$\Sigma_{t|t} = \sum_{i=1}^{N_m} \mu^i_t \left[\Sigma^i_{t|t} + (\hat{\mathbf{x}}^i_{t|t} - \hat{\mathbf{x}}_{t|t})(\hat{\mathbf{x}}^i_{t|t} - \hat{\mathbf{x}}_{t|t})^\top\right]$$

- $\hat{\mathbf{x}}_{t|t}$ is the overall state estimate at time t, calculated as the weighted sum of the state estimates $\hat{\mathbf{x}}^i_{t|t}$ from each mode.

- $\Sigma_{t|t}$ is the overall covariance of the state estimate at time t, reflecting the uncertainty in the estimate. It is computed as a weighted sum of the covariances $\Sigma^i_{t|t}$ of each mode, plus discrepancy between the individual mode estimates $\hat{\mathbf{x}}^i_{t|t}$ and the overall mean $\hat{\mathbf{x}}_{t|t}$.

IMM details

Suppose we have the previous sufficient statistics $\{\hat{x}^i_{t-1|t-1}, \Sigma^i_{t-1|t-1}, \mu^i_{t-1}\}_{i=1}^{N_m}$. Then, a single step of the IMM algorithm to obtain the current sufficient statistics $\{\hat{x}^i_{t|t}, \Sigma^i_{t|t}, \mu^i_t\}_{i=1}^{N_m}$ is given as follows:

- **Mixing:**

– Calculate the mixing probabilities $\{\mu^{ji}_{t-1|t-1}\}^{N_m}_{i,j=1}$ as:

$$\mu^{ji}_{t-1|t-1} = \frac{\pi^{ji}\mu^{j}_{t-1}}{\sum_{\ell=1}^{N_m}\pi^{\ell i}\mu^{\ell}_{t-1}}$$

Here, the mode switching probability $\pi_{ij} = P(\lambda_t = j|\lambda_{t-1} = i)$ is the ij th element in the mode transition matrix Π. The mode transition is governed by a finite state Markov chain. The mode transition matrix is obtained by incorporating the mode evolution in our simulation. However, the performance of the IMM algorithm is insensitive to the mode switching probabilities as reported in Thai and Bayen (2014).

– Calculate the mixed estimates $\{\hat{x}^{0i}_{t-1|t-1}\}^{N_m}_{i=1}$ and covariances $\{\Sigma^{0i}_{t-1|t-1}\}^{N_m}_{i=1}$ as:

$$\hat{x}^{0i}_{t-1|t-1} = \sum_{j=1}^{N_m}\mu^{ji}_{t-1|t-1}\hat{x}^{j}_{t-1|t-1}$$

$$\Sigma^{0i}_{t-1|t-1} = \sum_{j=1}^{N_m}\mu^{ji}_{t-1|t-1}\left[\Sigma^{j}_{t-1|t-1} + (\hat{x}^{j}_{t-1|t-1} - \hat{x}^{0i}_{t-1|t-1})(\hat{x}^{j}_{t-1|t-1} - \hat{x}^{0i}_{t-1|t-1})^{\top}\right]$$

- **Mode Matched Filtering:**

  Mode Matched Prediction Update:

  – For the ith model, $i = 1, \ldots, N_m$:

  Calculate the predicted estimate $\hat{x}^{i}_{t|t-1}$ and covariance $\Sigma^{i}_{t|t-1}$ from the mixed estimate $\hat{x}^{0i}_{t-1|t-1}$ and covariance $\Sigma^{0i}_{t-1|t-1}$ as:

$$\hat{x}^{i}_{t|t-1} = A^{(i)}\hat{x}^{0i}_{t-1|t-1} + b^{(i)} + u_t$$

$$\Sigma^{i}_{t|t-1} = A^{(i)}\Sigma^{0i}_{t-1|t-1}A^{(i)\top} + Q$$

  Note, the update is following Eq.5.1. $A^{(i)}$ and $b^{(i)}$ correspond to mode i.

  Mode Matched Measurement Update:

  – For the ith model, $i = 1, \ldots, N_m$:

Calculate the updated estimate $\hat{x}^i_{t|t}$ and covariance $\Sigma^i_{t|t}$ from the predicted estimate $\hat{x}^i_{t|t-1}$ and covariance $\Sigma^i_{t|t-1}$ as:

$$\hat{x}^i_{t|t} = \hat{x}^i_{t|t-1} + K^i_t(y_t - \hat{y}^i_{t|t-1})$$

$$\Sigma^i_{t|t} = \Sigma^i_{t|t-1} - K^i_t S^i_t K^{i\top}_t$$

Here, the predicted measurement $\hat{y}^i_{t|t-1}$, innovation covariance S^i_t, and Kalman gain K^i_t are given by:

$$\hat{y}^i_{t|t-1} = C\hat{x}^i_{t|t-1}$$

$$S^i_t = C\Sigma^i_{t|t-1}C + R$$

$$K^i_t = \Sigma^i_{t|t-1}C(S^i_t)^{-1}$$

Note, in our current setting, $D^{(i)}$ is also an identity matrix.

Calculate the updated mode probability μ^i_t as:

$$\mu^i_t = \frac{\mathcal{N}(y_t; \hat{y}^i_{t|t-1}, S^i_t)\sum_{j=1}^{N_m}\pi^{ji}\mu^j_{t-1}}{\sum_{\ell=1}^{N_m}\mathcal{N}(y_t; \hat{y}^\ell_{t|t-1}, S^\ell_t)\sum_{j=1}^{N_m}\pi^{j\ell}\mu^j_{t-1}}$$

- **Output Estimate Calculation:**
  - Calculate the overall estimate $\hat{x}_{t|t}$ and covariance $\Sigma_{t|t}$ as:

$$\hat{x}_{t|t} = \sum_{i=1}^{N_m}\mu^i_t\hat{x}^i_{t|t}$$

$$\Sigma_{t|t} = \sum_{i=1}^{N_m}\mu^i_t\left[\Sigma^i_{t|t} + (\hat{x}^i_{t|t} - \hat{x}_{t|t})(\hat{x}^i_{t|t} - \hat{x}_{t|t})^\top\right]$$

The IMM algorithm is used as a model-based state estimation algorithm in our study. We aim to show the optimal sensor location based on observability analysis has a value in traffic surveillance. The estimation qualities are compared for different sensor locations, and we attempt to find the correlation between observability and estimation quality.

Two numerical examples are used in the following sections: 1. A 6-link network traffic dynamics is simulated using SUMO. 2. A 22-link network simulated using the stochastic LQM in Eq.5.1.

5.2 A 6-link toy network

Let us first start with the aforementioned 6-link network example shown in Figure 4.1. The traffic is simulated using the simulation software SUMO. The ground truth data is intentionally generated using simulation rather than the LQM to answer the following questions: What if the traffic network dynamics does not exactly follow the LQM, which after all is a simplification of the reality? Is the observability property calculated based on the mode-switching link queue model still meaningful?

In the simulation, each link is set to be 200 meters long. Link 1,3,4,5 are considered as the mainstream corridor and have higher free-flow travel speeds than link 2 and 6 (on- and off-ramps). The fundamental diagrams are fitted both for the mainstream and ramps:

- Link 1,3,4,5: $v_f = 108\text{km/h}, w = 18\text{km/h}, k_j = 135\text{veh/km}$
- Link 2,6: $v_f = 72\text{km/h}, w = 15\text{km/h}, k_j = 140\text{veh/km}$

The outflow at link 5 is intentionally controlled to generate congestion that propagates to link 1, 2, 3, and 4. During the simulation, boundary flows are captured by additional loop detectors at the immediate upstream/downstream of the 6-link network. Real-time occupancy data collected at the midpoint of each link is then converted to density, serving as the ground truth. Note that the ground truth simulated by SUMO is independent of our choice of the macroscopic traffic model. The layout of the sensors in the SUMO network is shown in Figure 5.1.

Figure 5.1: SUMO network

To quantify the observability in Eq.4.4a, we first need to model the traffic dynamics in the LQM. The fundamental diagram and the error covariance are estimated. The fitted LQM and the SUMO simulation are well allign, as shown in Figure 5.2.

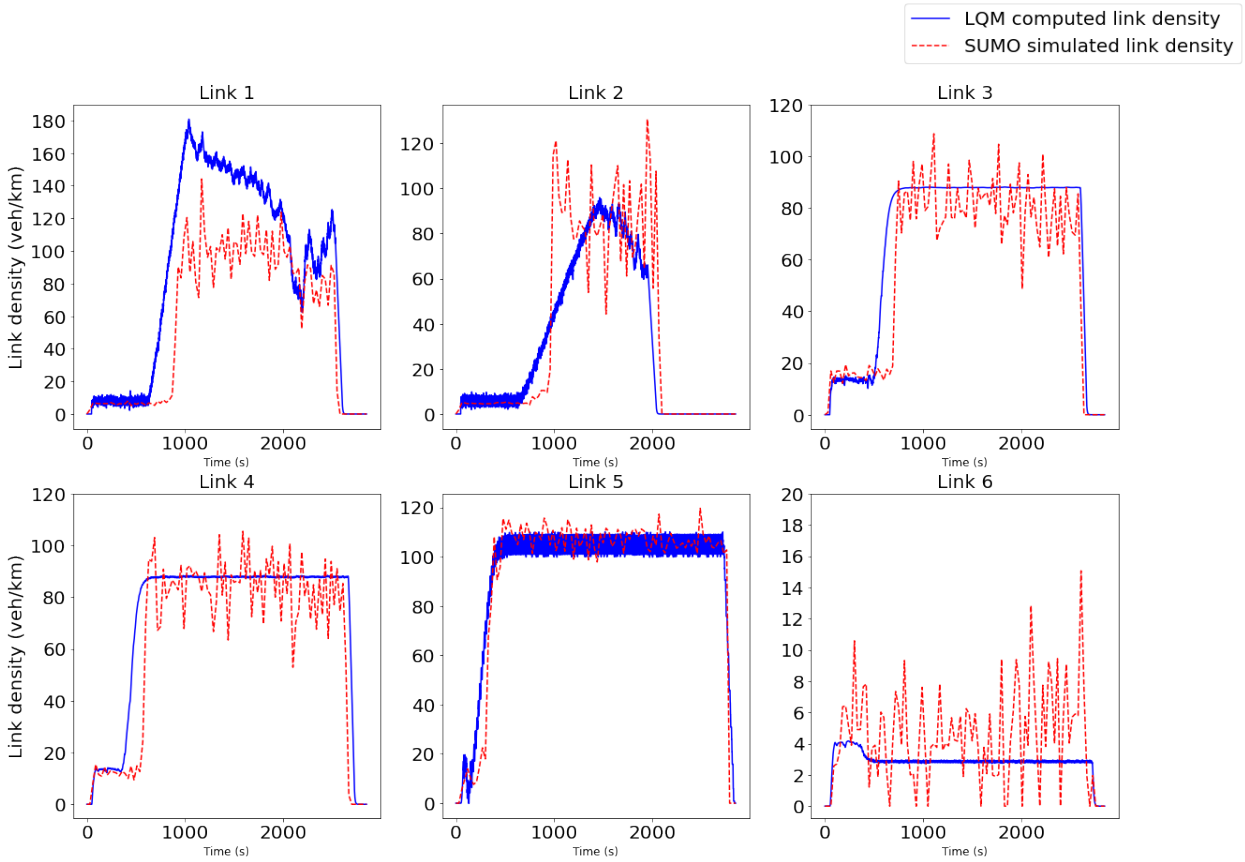


Figure 5.2: The fitted LQM vs the actual simulated link density

The evolution of link densities and the corresponding mode switches in this experiment is illustrated in Figure 5.3. Starting from free-flow condition, the queue spillback generates 7 distinct traffic conditions. The SUMO network and its setup can be found at https://github.com/Xinyue-H/Sensor-Location-Optimization.

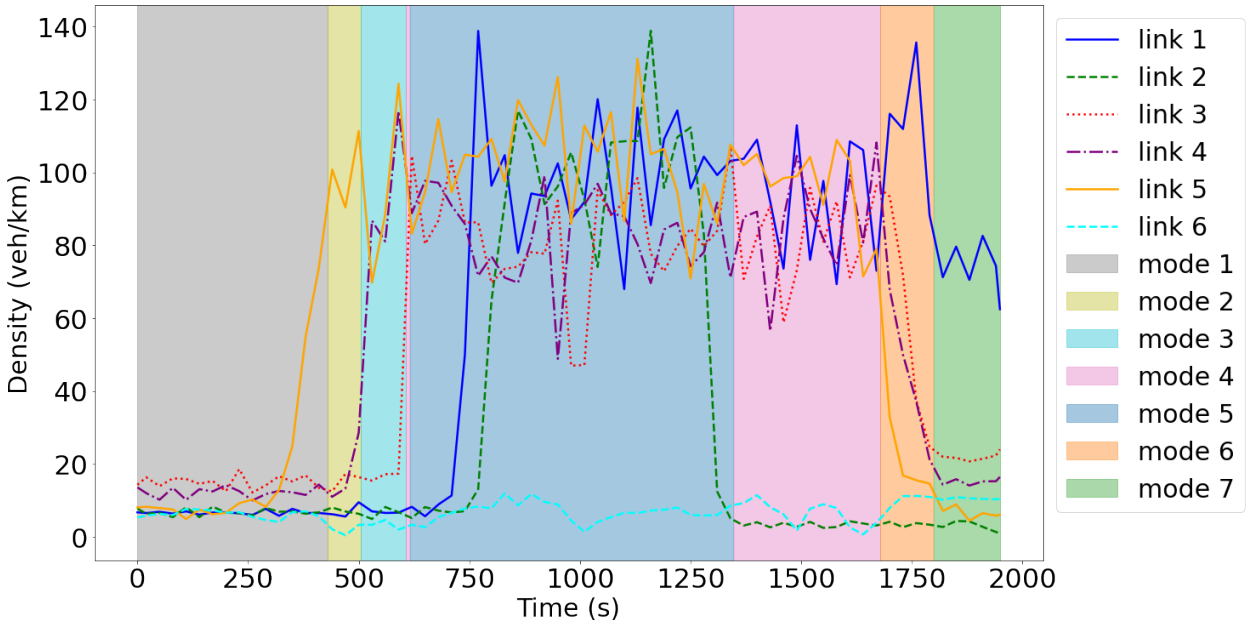


Figure 5.3: Link densities and mode switches in the 6-link network experiment

Let us first look at the results showing how different sensor location strategies could impact the observability outcome. We set the number of sensors to be deployed as 4. By solving the optimization problem in Eq. (4.4), we identify the optimal sensor locations as links 1, 2, 5 and 6, which leads to 5.82 observable links on average. This optimal sensor deployment strategy is compared with all other possible sensor location choices, which lead to an average observability ranging between 4.31 and 5.59. Detailed result is reported in Table 5.1.

Next, we address possible questions that reader might be wondering: the definition and analysis associated with observability seem to be centered around a piecewise linear dynamic traffic model, which might deviate far from the reality. Besides serving our intellectual curiosity in a theoretical sense, does the established observability understanding have any practical meaning? The following experiment shows that increased observability leads to better estimation quality, which indicates that a sensor placement strategy that maximizes observability should benefit downstream traffic state estimation applications.

Table 5.1 compares the degree of observability with the estimation quality, measured by the Mean Absolute Percentage Error (MAPE), under different sensor location strategies. The MAPE

Table 5.1: Different sensor locations comparison

Sensor location	Observability	MAPE (%)
(1, 2, 5, 6)	5.82	17.18
(1, 2, 4, 6)	5.59	38.73
(1, 3, 5, 6)	5.59	26.54
(1, 4, 5, 6)	5.52	33.06
(2, 3, 5, 6)	5.27	344.64
(1, 3, 4, 6)	5.27	58.87
(1, 2, 3, 6)	4.96	91.85
(1, 2, 3, 5)	4.96	394.37
(2, 3, 4, 6)	4.95	382.66
(1, 2, 4, 5)	4.95	199.81
(2, 4, 5, 6)	4.89	25.24
(1, 2, 3, 4)	4.64	548.89
(1, 3, 4, 5)	4.62	547.34
(3, 4, 5, 6)	4.62	288.71
(2, 3, 4, 5)	4.31	1200.27

aggregated across all n links and T time steps is computed as:

$$\text{MAPE} = \frac{1}{nT} \sum_{i=1}^{n} \sum_{t=1}^{T} \frac{x_t^{(i)} - \hat{x}_t^{(i)}}{x_t^{(i)}} \times 100$$

where $x_t^{(i)}$ represents the actual value at time step t for the i-th link time series data, and $\hat{x}_t^{(i)}$ is the corresponding estimated value.

From these results, we hope to observe the correlation between observability and state estimation. This is not trivial because good observability does not necessarily guarantee good estimation quality since estimation also involves the identification of the mode switches – for estimation, the exact mode switching time and the true mode remain unknown and are inferred simultaneously with the state estimation; the only given information are the boundary flows, fundamental diagrams, and the covariance of the Gaussian noise term.

We observe that the highest observability sensor location corresponds to the lowest estimation error. Also, sensor placement with higher observability generally results in better estimation quality. Conversely, instances of poor observability can lead to substantial MAPE, primarily because loss of observability leads to uninformed guesses of the state and a lack of correction. For example,

our IMM algorithm does not enforce the link density to be in the range from zero to jam density. When a state is unobservable under the current system mode, the measurement lacks information about the unobservable state. Consequently, the estimation quality for the unobservable states is very poor. Let us take a closer look at the case of sensor placement (1,2,3,4) as shown in Figure 5.4. During the period when both link 5 and 6 are unobservable, the estimation error consistently grows. When link 5 becomes observable at 430 seconds, the estimation of link 5 density now can be corrected by the measurement and the estimated density aligns closely with the actual value. Subsequently, when link 5 becomes unobservable again, the estimation quality soon deteriorates. While for link 6, since it is not observable throughout the entire period, its estimated state is consistently poor. Thus, although observability and state estimation are two different concepts, we demonstrate that it is advantageous to place sensors in a way that maximizes the average number of observable components. This holds true even when the mode switching time and identification are not provided.

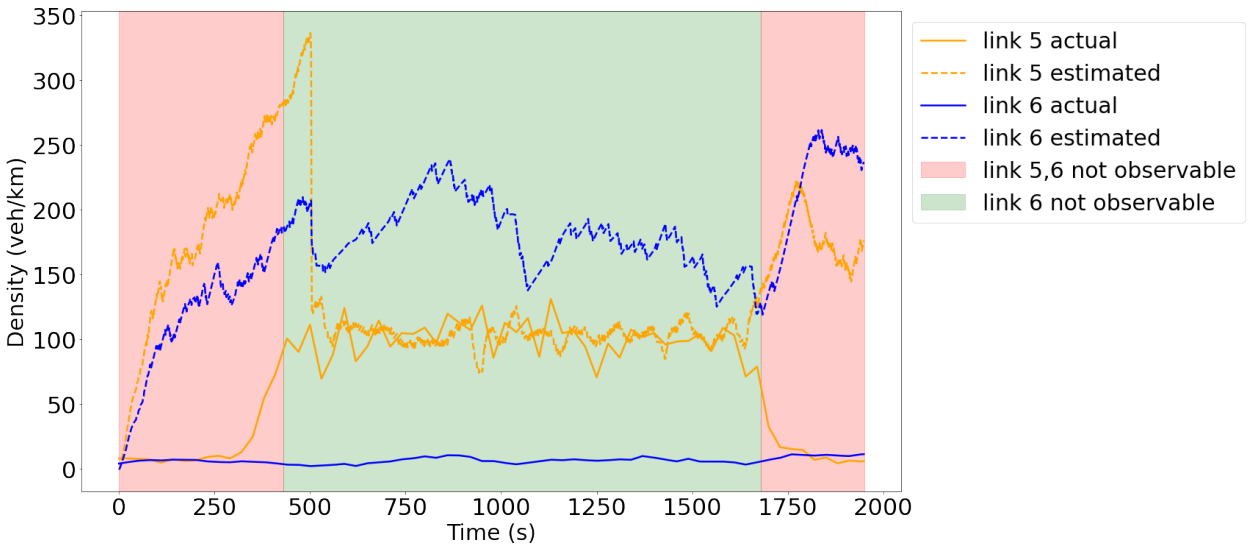


Figure 5.4: Estimation with sensor located on link 1,2,3,4

5.3 A fabricated 22-link network

Now let us show a numerical example using the network used in Rinaldi (2018) (see Figure 5.5). The experiment is set up as follows. The traffic is simulated based on Eqs. 5.1 and Eq. 5.2. We acknowledge that simply adding a Gaussian process noise to the LQM indeed leads to an oversimplified stochastic traffic flow model. For more sophisticated stochastic traffic modeling, interested readers can refer to Jabari and Liu (2013) and Sumalee et al. (2011). Our purpose here is not to develop a stochastic traffic model, but rather to use one to generate traffic ground truth that deviates from the LQM used in our analysis.

5.3.1 Observability and estimation quality

We intentionally controlled boundary flows to generate 585 different congestion modes. Links 10 and 11 are set to be bottlenecks. Modes are weighted based on the duration of their occurrence. Details of the parameter setup of the experiment can be found at https://github.com/Xinyue-H/Sensor-Location-Optimization.

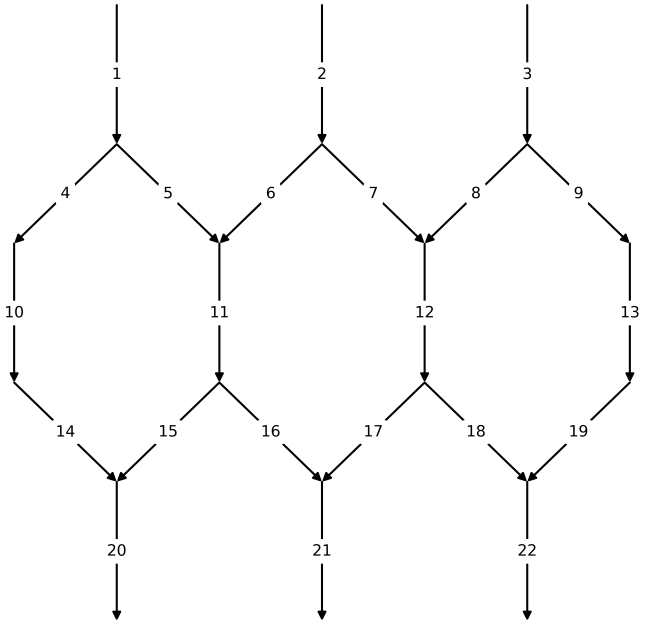


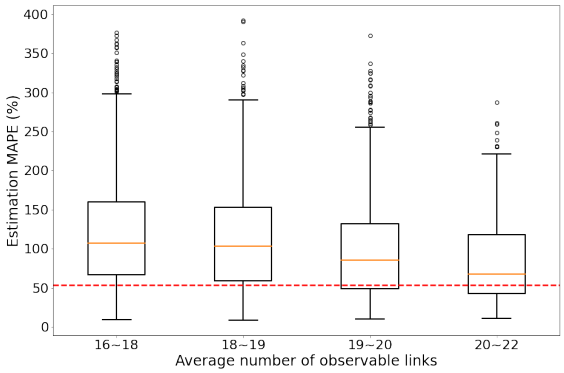
Figure 5.5: Numerical example: a 22-link network

First, let us explore the performance of different sensor location strategies in terms of both observability and estimation quality. The total number of sensors takes a value between 16 and 19. The optimal sensor location solution for each p is obtained by solving the optimization problem in Eq. 4.4. The optimal sensor location solutions and their resulted average observabilities and MAPE are reported in the Table 5.2. To generate different sensor location strategies for comparison purpose, we enumerated all sensor location combinations for p =17, 18, or 19. For $p = 16$, we sampled 6000 possible sensor location combinations, 1500 in each bin. Figure 5.6 reports the estimation errors from different sensor location strategies that are categorized based on their average observability. Each box plot shows the max, 75% quartile, median, 25% quartile, and the min MAPE resulted from various sensor location solutions within the corresponding observability bin[1]. The red dashed line corresponds to the optimal sensor location solution. Using $p = 16$ as an example, our optimal sensor location leads to an average MAPE around 54%.

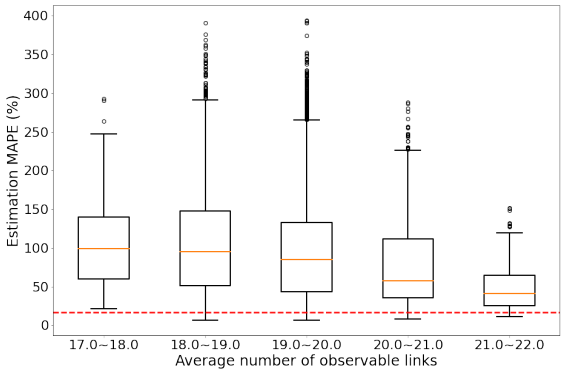
[1]For the case of 16 sensors, since there are very few sensor location strategies resulted in an observability between 16-17 and 21-22, the bins are merged with 17-18 and 20-21, respectively.

Table 5.2: Optimal sensor placement solutions

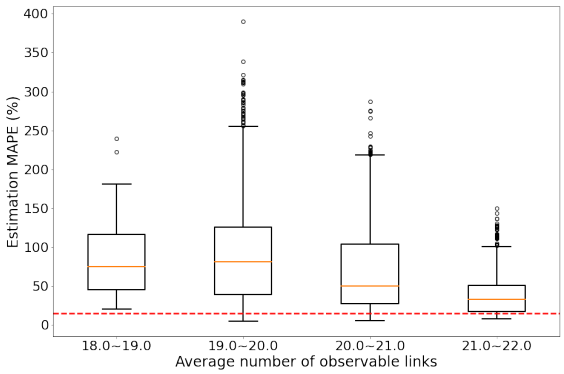
Sensor Num	Sensor location	Observability	MAPE (%)
16	(1, 2, 3, 5, 6, 7, 8, 9, 14, 15, 16, 17, 18, 19, 20, 21)	21.52	53.69
17	(1, 2, 3, 5, 6, 7, 8, 9, 14, 15, 16, 17, 18, 19, 20, 21, 22)	21.74	16.97
18	(1, 2, 3, 5, 6, 7, 8, 9, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22)	21.90	14.86
19	(1, 2, 3, 5, 6, 7, 8, 9, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22)	21.96	11.41



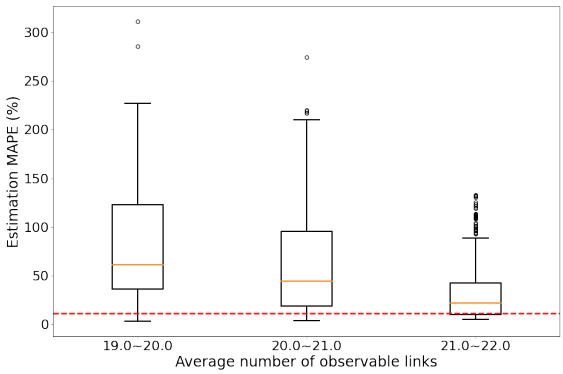
(a) 16 sensors



(b) 17 sensors



(c) 18 sensors



(d) 19 sensors

Figure 5.6: Impact of observability on estimation quality

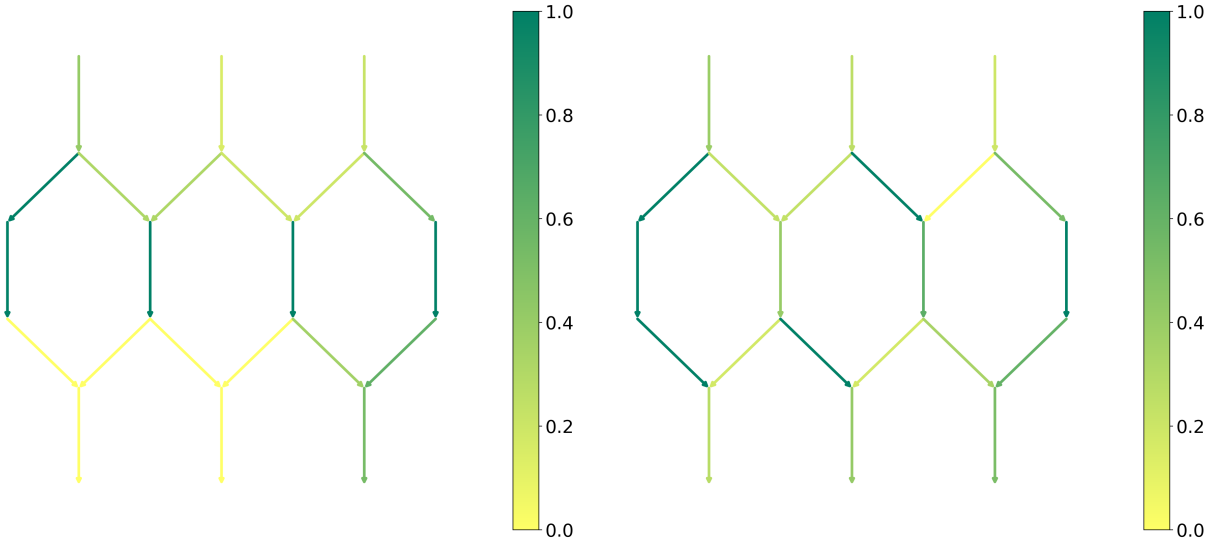
Based on the results in Figure 5.6, we have the following findings:

- The optimal sensor location solutions from our max-observability sensor location model perform consistently well in terms of traffic state estimation measured by MAPE.
- Overall, sensor placement with better observability tends to result in improved estimation quality, indicated by lower MAPE values.
- Sensor placement with high observability is associated with a reduced risk of extreme inac-

curacies in estimation.

These results indicate that maximizing observability across various traffic conditions should be a key consideration in the sensor placement problem due to its demonstrated significance in dynamic traffic state estimation, which is a key for real-time traffic surveillance.

In the experiment above, the number of sensors is relatively large. Readers might wonder whether the concept of obervability still matters in case one has only a small number of sensors to deploy. Now, let us show an example of placing only 4 sensors in the 22-link network. By solving the optimization problem in Eq. 4.4 with $p = 4$ and the 585 modes being incorporated, we identify the optimal sensor locations as links 7, 13, 14, 16, which leads to 11.76 observable links on average. This max-observability sensor location strategy is compared with a naive choice of placing sensors on links 10, 11, 12 and 13 to cover all four paths connecting the three OD pairs. The naive choice leads to 8.89 observable links on average. Figure 5.7 provides more details of the comparison of the two sensor location choices. Darker green indicates better observability. If a link is observable all-time, the corresponding value should be 1. This example shows that even with very few sensors to work with, strategically placing them still makes a significant difference.



(a) Naive solution based on path-covering rule

(b) Optimal sensor location solution

Figure 5.7: Comparison of observability resulted from two sensor location solutions

In Figure 5.7a, the links on the lower left corner (link 14, 15, 16, 17, 20, 21) are poorly observed. This can cause very limited information on those links' dynamics.

5.3.2 Advantage of observability: a comparison with greedy sensor placement strategies

Observability maximization vs greedy sensor placement strategies

Although the previous example shows the our sensor placement based observability maximization leads to better estimation quality, one may wonder how observability based sensor location performs compared with other sensor placement strategies. In this subsection, we proposed 3 greedy sensor location strategies which are potentially good strategies for traffic state estimation:

- **Greedy 1 (maximum variance)**: locate sensors on links where the link densities vary the most
- **Greedy 2 (maximum flow)**: locate sensors on links with maximum traffic flow
- **Greedy 3 (maximum density)**: locate sensors on most congested links

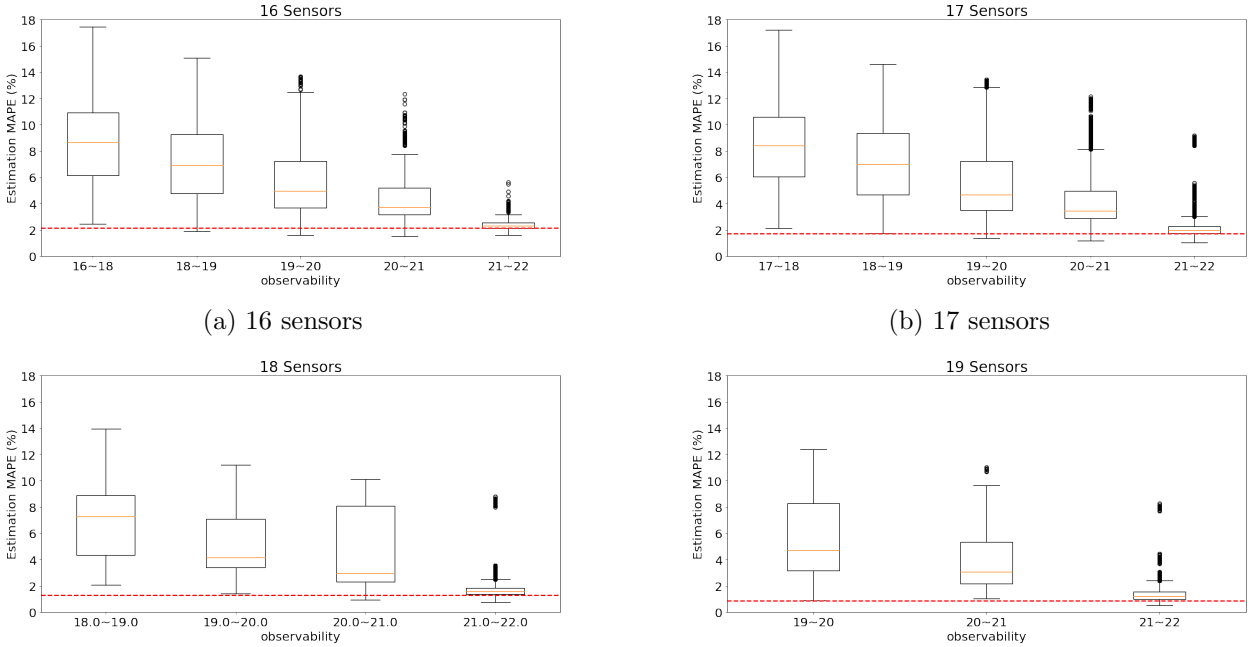
All of these greedy sensor placement strategies maximize information in some sense. In the previous subsection example, the 22 links have 2 different fundamental diagrams. The same parameters cause the problem of symmetry which undermines observability. To better illustrate the effectiveness of observability maximization, we implement a new simulation where structural observability leads to exact observability. Out of the 22 links, we show for different number of sensors ($p = 11, 12, ..., 19$), the observability maximization sensor location strategy outperforms the greedy algorithms, as summarized in Table 5.3:

The lowest MAPE for each sensor number p is highlighted. From this result, the sensor location by observability maximization has the best performance in state estimation compared with the proposed greedy sensor placement strategies. Similar to Figure 5.6, we enumerated all sensor location combinations for p =17, 18, or 19. For p = 16, we sampled 6000 possible sensor

p	Max_var (%)	Max_flow (%)	Max_density (%)	MAPE_maxobs (%)
11	4.70	4.96	5.46	**4.51**
12	4.13	4.57	4.57	**3.82**
13	3.72	4.12	4.12	**3.45**
14	3.44	3.72	3.77	**3.34**
15	3.31	3.37	3.37	**2.90**
16	2.81	2.97	2.97	**2.39**
17	2.47	2.47	2.47	**2.08**
18	2.24	1.88	1.88	**1.70**
19	1.65	1.65	**1.34**	**1.34**

Table 5.3: MAPE in estimation: observability maximization vs greedy sensor placement

location combinations, 1500 in each bin. Figure 5.8 reports the estimation errors from different sensor location strategies that are categorized based on their average observability.



(a) 16 sensors

(b) 17 sensors

(c) 18 sensors

(d) 19 sensors

Figure 5.8: Impact of observability on estimation quality (when structural observability is equivalent to exact observability)

5.3.3 Observability quality: when structural observability is not equivalent to exact observability

Structural observability does not always imply (exact) observability. First, we show an example where structural observability and exact observability may differ. Note that in this example the parameters of some links are intentional altered (to set to be identical) to create symmetry, with the intention to generate a case where structural observability differs from observability. The mode matrix A in this case is:

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -108 & -18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 108 & 0 & 0 & 0 & 0 & 0 & -108 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -108 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 0 & 0 & 36 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 108 & 0 & 0 & 0 & -108 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 54 & 0 & 0 & 0 & -108 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 54 & 0 & 0 & 0 & 0 & -108 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -108 & -18 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 108 & 108 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 108 & 108 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -18
\end{bmatrix}$$

Since exact observability requires knowledge of the exact values of parameters, we will implement the experiment in the "most likely" mode (the one that occurs the most during the simulation). The inference diagram and the necessary sensors are shown in Figure 5.9. For structural observability, our Graphical Approach concludes only the source nodes (link 1, 2, 3, 6, 9, 20, 21) have to be observed, as shown in green. However, for exact observability considering the link weights, extra sensors must be put on link 15 and link 17 (colored in blue). In link pairs (14, 15) and (16, 17), the individual links within a pair cannot be distinguished due to symmetry with only the upstream

being observed. As a result, extra sensors are needed on at least one individual component within a pair.

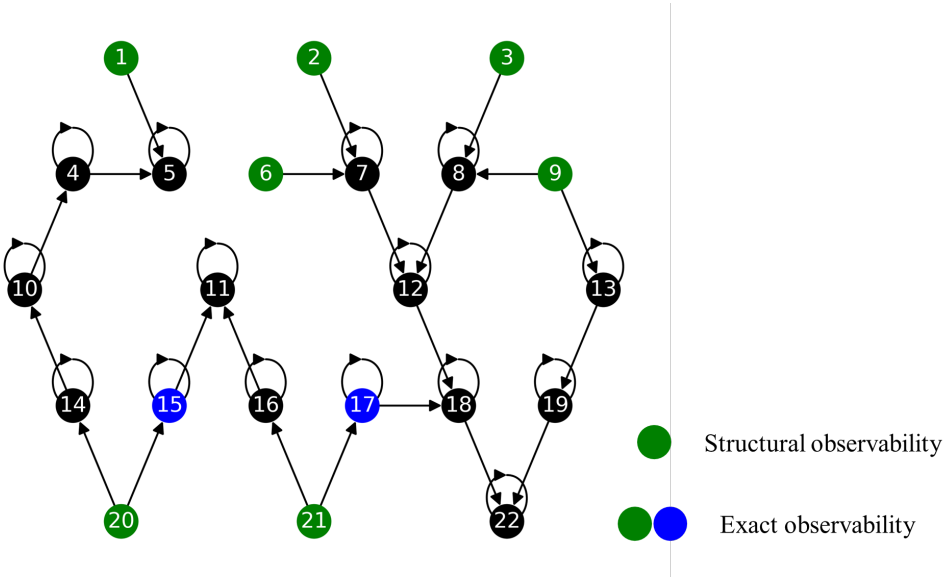


Figure 5.9: Sensor placement for exact observability for one of the modes

With this example, we understand that structural observability does not guarantee exact observability. When exact observability cannot be achieved, the proposed structural observability maximization result might be deteriorated. After all, the so-called observable system may no longer be observable any more. We demonstrate this using 2 more simulation settings on the same 22-link network example.

Case 1: 22 links with only 2 Fundamental diagrams

In this example, the 22 links only have 2 different fundamental diagrams. The same parameters in the fundamental diagrams lead to loss of observability, even if structural observability is achieve. To fully observe the system, extra sensors are needed and the conclusion from the structural observability should be treated with caution.

The comparison between observability maximization and greedy sensor placement strategies are summarized in Table 5.5:

p	Max_var (%)	Max_flow (%)	Max_density (%)	MAPE_maxobs (%)
11	175.57	13.89	30.47	**13.70**
12	205.79	**10.96**	22.93	13.22
13	223.66	**9.32**	11.88	12.12
14	221.37	**7.91**	10.89	20.02
15	138.64	**7.37**	9.16	18.07
16	71.28	**6.07**	9.12	19.34
17	69.99	**5.42**	8.05	16.82
18	6.52	**3.62**	5.81	6.13
19	4.22	**2.38**	4.40	3.82

Table 5.4: Case 1: 22-link network simulated with 2 fundamental diagrams

Case 2: 22 links with 10 Fundamental diagrams

In this example, the 22 links have 10 different fundamental diagrams. Link 14, 15, 16, 17 have the same fundamental diagram, causing a symmetry leading to loss of observability.

The comparison between observability maximization and greedy sensor placement strategies are summarized in Table 5.5:

p	Max_var (%)	Max_flow (%)	Max_density (%)	MAPE_maxobs (%)
11	20.07	**7.14**	16.10	9.01
12	**4.42**	6.70	12.06	8.89
13	**4.19**	5.57	5.62	7.43
14	**3.74**	3.81	4.81	6.85
15	**3.15**	3.33	3.33	4.95
16	**2.62**	2.80	2.95	4.83
17	2.23	2.22	2.54	**2.10**
18	1.77	1.62	2.00	**1.50**
19	1.53	1.42	1.42	**1.13**

Table 5.5: Case 2: 22-link network simulated with 10 fundamental diagrams

From these two additional examples, we have two observations:

- The observability maximization is generally a good sensor location strategy for accurate state estimation, although it might not be the best compared with other good greedy strategies when observability is deteriorated.
- The estimate accuracy is undermined for these two examples. This is due to the lack of

observability when the symmetry in parameters occurs.

With the examples in this chapter, we conclude that observability maximization is good strategy for sensor placement to understand the traffic density dynamics. When observability is deteriorated due to symmetry of parameters, the significance of observability is undermined. It implies that the exact parameterization matters in state estimation. Other metrics of observability such as observability Gramian and condition number of observability matrix $\mathcal{O}$ can be examined.

Chapter 6

Discussion and Conclusion

In this study, we have leveraged the algebraic and graphical properties of traffic network dynamics to design optimal sensor deployment solutions to ensure full observability or maximize observability. We have proposed a systematic Algebraic Approach to identify sensor locations for full observability with a minimal number of sensors. The Algebraic Approach remains applicable regardless of the choice of parameter, and thus is an exact observability method. Additionally, by exploiting special features of dynamic traffic systems, we are able to establish new analytical and computational methods beyond what has been reported in the general complex systems literature.

Our sensor location model takes into account a wide range of traffic conditions, and provides an optimal solution that performs well in an average sense. This design philosophy supports the growing interest in moving away from a deterministic mentality and achieving a system design that could cope with broader future uncertainties. It is true that observability analysis of a dynamical system would require some prior knowledge of the system. In the I80 East case study, we showcased that even with coarse information about the modes based on speed information from Google Maps, we are able to identify good sensor location solutions. In a sense, one can consider the preparation of the information about traffic modes equivalent to the exercise of uncertainty modeling when one is constructing a stochastic optimization model. Consistent with the widely adopted practice in the stochastic programming field, the modes and their associated weights/probabilities that are input

to the sensor location model do not need to cover the entire distribution of the uncertain traffic condition, rather it is meant to provide a good representation of the important traffic scenarios that are considered significant to the system operator.

Despite the progress presented in this dissertation, the understanding of observability in dynamic traffic systems is still in its preliminary stage and demands much further research. For example, to incorporate the various traffic conditions, either simulations or real data can be used to provide such prior knowledge, as we demonstrated in the numerical example and case study. For future research, one may consider incorporating the simultaneous identification or detection of mode changes in a sensor placement problem. Also, the numerical examples included in this dissertation only include varying traffic conditions as a source of uncertainty. There could be other sources in a traffic system that might generate uncertainties. For example, an incident might change the shape of the fundamental diagram of involved links. The stochastic programming based sensor location model indeed can handle a wide range of uncertainty scenarios, but to prepare input characterizing those uncertainties is not a trivial task, which itself requires deep domain expertise. In addition, the sensor optimization model (4.4), which is equivalent to a two-stage stochastic programming model, treats different traffic modes as possible snapshots of uncertain scenarios, despite the dynamic evolution of these modes. If one wishes to model the sequence of the occurrence of various modes caused by dynamic evolution, one may consider a multi-stage stochastic programming framework, where uncertainty evolves over time as a scenario tree. However, such additional modeling flexibility would come at a price of computational difficulty. Another potential direction is to integrate observability and estimation quality in the context of sensor placement problems. Through numerical examples, we have demonstrated a positive correlation between observability and traffic state estimation quality. How to explicitly express estimation quality in the optimal sensor location model, whether through constraints or in the objective function, would be an interesting question to explore. Moreover, the current optimization model aims for an average performance. When estimation quality comes to the picture, there could be many creative ways of expressing a targeted performance, such as in terms of reliability or robustness, depending on one's risk preferences. Finally, our structural observability analyses are based on piecewise linear systems. Even though we

are able to show the effectiveness of incorporating structural observability in dynamical systems that deviate from the piecewise linear model, it would be too naive to assume there could not be more suitable information measures than what we introduced in this work. Exploring other ways of measuring information richness to guide sensor placement strategies would be a worthy effort for future research.

Bibliography

Shaurya Agarwal, Pushkin Kachroo, and Sergio Contreras. A dynamic network modeling-based approach for traffic observability problem. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1168–1178, 2015.

Nikolaos Bekiaris-Liberis, Claudio Roncoli, and Markos Papageorgiou. Highway traffic state estimation per lane in the presence of connected vehicles. *Transportation research part B: methodological*, 106:1–28, 2017.

Lucio Bianco, Giuseppe Confessore, and Pierfrancesco Reverberi. A network based model for traffic sensor location with implications on o/d matrix estimates. *Transportation Science*, 35(1):50–60, 2001.

Claus C Carøe and Rüdiger Schultz. Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1-2):37–45, 1999.

Enrique Castillo, Antonio J Conejo, Rosa E Pruneda, and Cristina Solares. Observability analysis in state estimation: A unified numerical approach. *IEEE Transactions on Power Systems*, 21(2): 877–886, 2006.

Enrique Castillo, Antonio J Conejo, José María Menéndez, and Pilar Jiménez. The observability problem in traffic network models. *Computer-Aided Civil and Infrastructure Engineering*, 23(3): 208–222, 2008a.

Enrique Castillo, José María Menéndez, and Pilar Jiménez. Trip matrix and path flow reconstruc-

tion and estimation based on plate scanning and link observations. *Transportation Research Part B: Methodological*, 42(5):455–481, 2008b.

Enrique Castillo, Inmaculada Gallego, Santos Sánchez-Cambronero, and Ana Rivas. Matrix tools for general observability analysis in traffic networks. *IEEE Transactions on Intelligent Transportation Systems*, 11(4):799–813, 2010.

Enrique Castillo, Maria Nogal, Ana Rivas, and Santos Sánchez-Cambronero. Observability of traffic networks. optimal location of counting and scanning devices. *Transportmetrica B: Transport Dynamics*, 1(1):68–102, 2013.

Enrique Castillo, Zacarías Grande, Aida Calviño, Wai Yuen Szeto, and Hong K Lo. A state-of-the-art review of the sensor location, flow observability, estimation, and prediction problems in traffic networks. *Journal of Sensors*, 2015, 2015.

Ricardo A. Collado, Dávid Papp, and Andrzej Ruszczyński. Scenario decomposition of risk-averse multistage stochastic programming problems. *Annals of Operations Research*, 200(1):147–170, November 2012. ISSN 0254-5330. doi: 10.1007/s10479-011-0935-y. Funding Information: This research was supported by the NSF award CMII-0965689.

Sergio Contreras, Pushkin Kachroo, and Shaurya Agarwal. Observability and sensor placement problem on highway segments: A traffic dynamics-based approach. *IEEE Transactions on Intelligent Transportation Systems*, 17(3):848–858, 2015.

Carlos F Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28 (4):269–287, 1994.

Carlos F Daganzo. The cell transmission model, part ii: network traffic. *Transportation Research Part B: Methodological*, 29(2):79–93, 1995.

Adam Danczyk, Xuan Di, and Henry X Liu. A probabilistic optimization model for allocating freeway sensors. *Transportation Research Part C: Emerging Technologies*, 67:378–398, 2016.

Anett Ehlert, Michael GH Bell, and Sergio Grosso. The optimisation of traffic count locations in road networks. *Transportation Research Part B: Methodological*, 40(6):460–479, 2006.

Xiang Fei, Hani S Mahmassani, and Stacy M Eisenman. Sensor coverage and location for real-time traffic prediction in large-scale networks. *Transportation Research Record*, 2039(1):1–15, 2007.

Xiang Fei, Hani S Mahmassani, and Pamela Murray-Tuite. Vehicular network sensor placement optimization under uncertainty. *Transportation Research Part C: Emerging Technologies*, 29: 14–31, 2013.

Monica Gentili and Pitu B Mirchandani. Locating sensors on traffic networks: Models, challenges and research opportunities. *Transportation research part C: emerging technologies*, 24:227–255, 2012.

Sheng-xue He. A graphical approach to identify sensor locations for link flow inference. *Transportation Research Part B: Methodological*, 51:65–76, 2013.

Shou-Ren Hu, Srinivas Peeta, and Chun-Hsiao Chu. Identification of vehicle sensor locations for link-based network traffic applications. *Transportation Research Part B: Methodological*, 43(8-9): 873–894, 2009.

Saif Eddin Jabari and Henry X Liu. A stochastic model of traffic flow: Gaussian approximation and estimation. *Transportation Research Part B: Methodological*, 47:15–41, 2013.

Wen Long Jin. A link queue model of network traffic flow. *Transportation Science*, 55(2):436–455, 2021. ISSN 15265447. doi: 10.1287/TRSC.2020.1012.

Rudolf E Kalman. On the general theory of control systems. In *Proceedings First International Conference on Automatic Control, Moscow, USSR*, pages 481–492, 1960.

Xiaopeng Li and Yanfeng Ouyang. Reliable sensor deployment for network traffic surveillance. *Transportation research part B: methodological*, 45(1):218–231, 2011.

Michael James Lighthill and Gerald Beresford Whitham. On kinematic waves ii. a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 229(1178):317–345, 1955.

Ching-Tai Lin. Structural controllability. *IEEE Transactions on Automatic Control*, 19(3):201–208, 1974.

Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. Controllability of complex networks. *nature*, 473(7346):167–173, 2011.

Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. Observability of complex systems. *Proceedings of the National Academy of Sciences*, 110(7):2460–2465, 2013.

Shima Sadat Mousavi and Anastasios Kouvelas. Structural observability of traffic density dynamics on a motorway ring road. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.

Laura Muñoz, Xiaotian Sun, Roberto Horowitz, and Luis Alvarez. Traffic density estimation with the cell transmission model. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 5, pages 3750–3755. IEEE, 2003.

ManWo Ng. Synergistic sensor location for link flow inference without path enumeration: A node-based approach. *Transportation Research Part B: Methodological*, 46(6):781–788, 2012.

Sebastian A Nugroho, Suyash C Vishnoi, Ahmad F Taha, Christian G Claudel, and Taposh Banerjee. Where should traffic sensors be placed on highways? *IEEE Transactions on Intelligent Transportation Systems*, 2021.

Manoj Panda, Dong Ngoduy, and Hai L Vu. Multiple model stochastic filtering for traffic density estimation on urban arterials. *Transportation research part B: methodological*, 126:280–306, 2019.

Paul I Richards. Shock waves on the highway. *Operations research*, 4(1):42–51, 1956.

Marco Rinaldi. Controllability of transportation networks. *Transportation Research Part B: Methodological*, 118:381–406, 2018.

R Tyrrell Rockafellar and Roger J-B Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research*, 16(1):119–147, 1991.

Majid Rostami-Shahrbabaki, Ali Akbar Safavi, Markos Papageorgiou, Peyman Setoodeh, and Ioannis Papamichail. State estimation in urban traffic networks: A two-layer approach. *Transportation Research Part C: Emerging Technologies*, 115:102616, 2020.

Andrzej Ruszczyński and Alexander Shapiro. Stochastic programming models. *Handbooks in operations research and management science*, 10:1–64, 2003.

Mostafa Salari, Lina Kattan, William HK Lam, HP Lo, and Mohammad Ansari Esfeh. Optimization of traffic sensor location for complete link flow observability in traffic network considering sensor failure. *Transportation Research Part B: Methodological*, 121:216–251, 2019.

Toru Seo, Alexandre M Bayen, Takahiko Kusakabe, and Yasuo Asakura. Traffic state estimation on highway: A comprehensive survey. *Annual reviews in control*, 43:128–151, 2017.

Agachai Sumalee, RX Zhong, TL Pan, and WY Szeto. Stochastic cell transmission model (sctm): A stochastic dynamic traffic model for traffic state surveillance and assignment. *Transportation Research Part B: Methodological*, 45(3):507–533, 2011.

Jérôme Thai and Alexandre M Bayen. State estimation for polyhedral hybrid systems and applications to the godunov scheme for highway traffic estimation. *IEEE Transactions on Automatic Control*, 60(2):311–326, 2014.

Francesco Viti, Marco Rinaldi, Francesco Corman, and Chris MJ Tampère. Assessing partial observability in network sensor location problems. *Transportation research part B: methodological*, 70:65–89, 2014.

Ren Wang and Daniel B Work. Interactive multiple model ensemble kalman filter for traffic estimation and incident detection. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 804–809. IEEE, 2014.

Hai Yang and Jing Zhou. Optimal traffic counting locations for origin–destination matrix estimation. *Transportation Research Part B: Methodological*, 32(2):109–126, 1998.

Isaak Yperman. The link transmission model for dynamic network loading. *Katholieke Universiteit Leuven*, 481:482, 2007.

Zhengzhong Yuan, Chen Zhao, Zengru Di, Wen-Xu Wang, and Ying-Cheng Lai. Exact controllability of complex networks. *Nature communications*, 4(1):1–9, 2013.

Liguo Zhang and Xuerong Mao. Vehicle density estimation of freeway traffic with unknown boundary demand–supply: an interacting multiple model approach. *IET Control Theory & Applications*, 9(13):1989–1995, 2015.

Xuesong Zhou and George F List. An information-theoretic sensor location model for traffic origin-destination demand estimation applications. *Transportation Science*, 44(2):254–273, 2010.

Appendix A

As we mentioned in the Algebraic Approach, the procedure may induce redundant sensors. This is due to the fact that **Step 2** is not aware of the sensor placement result corresponding to different eigenvalues. The numerical example corresponding to Figure 5.9 in Section 5.3 has the following system matrix A:

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -108 & -18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 108 & 0 & 0 & 0 & 0 & 0 & -108 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -108 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 0 & 0 & 36 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 108 & 0 & 0 & 0 & -108 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 54 & 0 & 0 & 0 & -108 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 54 & 0 & 0 & 0 & 0 & -108 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -108 & -18 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 0 & 9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 & 9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 108 & 108 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 108 & 108 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -18
\end{bmatrix}$$

If we implement **Step 1-3**, we will get extra sensors needed (link 7, 13) as shown in Figure A.1. These two sensors are identified to be redundant.

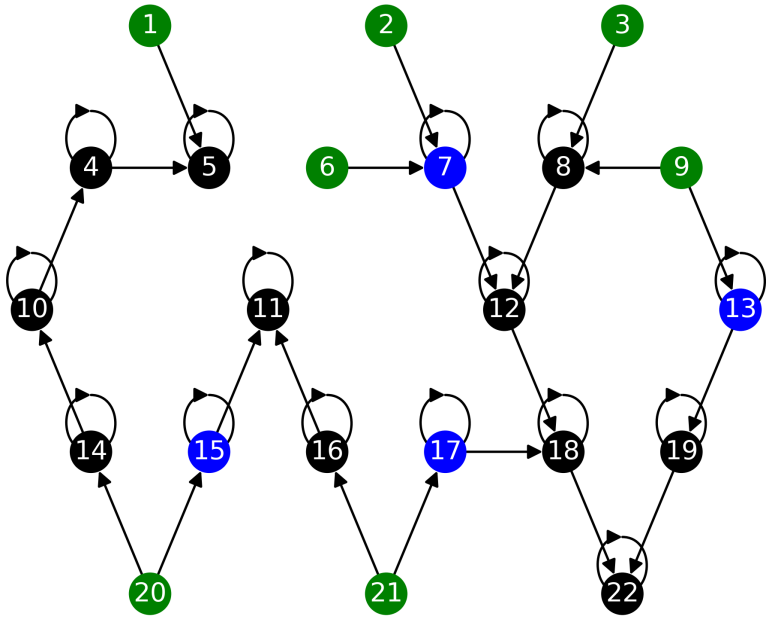


Figure A.1: Without Step 4: redundant sensors

Step 4 detects and removes the redundant sensors by implementing the following steps:

- The initial solution require to observe link 1, 2, 3, 6, 7, 9, 13, 15, 17, 20, 21.

- The corresponding observation matrix C is:

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}$$

- For each sensor, check if not using that sensor still leads to full observability: remove the corresponding row in C, use the reduced matrix C^* for the observability test (Kalman's rank test). If the system is observable, then the sensor is redundant.