공학박사 학위논문

Cooperative Estimation and Control of Large-scale Process Networks

대규모 공정 네트워크의 협동 추정 및 제어

2017년 2월

서울대학교 대학원

화학생물공학부

이 신 제

Cooperative Estimation and Control of Large-scale Process Networks

지도교수 이종민

이 논문을 공학박사 학위논문으로 제출함

2016년 12월

서울대학교 대학원

화학생물공학부

이신제

이신제의 공학박사 학위논문을 인준함

2016년 12월

위원장 ______________________ (인)

부위원장 ______________________ (인)

위원 ______________________ (인)

위원 ______________________ (인)

위원 ______________________ (인)

Abstract

Cooperative Estimation and Control of Large-scale Process Networks

Shin Je Lee
School of Chemical and Biological Engineering
The Graduate School
Seoul National University

State estimation and control of large-scale process network systems are considered as difficult problems because they consist of numerous subsystems and interactions between subsystems make the entire network dynamics complicated. Chemical processes and pipe networks are representative large-scale networks. In this thesis, we propose a novel cooperative estimation and control algorithms of large-scale process networks. In water pipe networks, a fault such as pipe leak or burst often happens and it is difficult to detect and diagnose. For fault detection and location of water pipe networks, state estimation can be an effective tool. However, a mathematical model describing dynamics of leak in water pipe networks does not exist. Before we develop a mathematical model of water pipe network, we propose a novel methodology to detect and locate leak in water pipe networks. Conventional detection methods include a cumulative sum (CUSUM) and a wavelet transform (WT). However, the CUSUM has a problem of slow response and the WT is sensitive to signal transitions. We integrate two algorithms to effectively detect sudden pressure changes of water pipe networks. The developed leak detection and location system is validated with real field data obtained from artificial leaks by opening hydrant valves in small-scale and medium-scale pipe networks and natural leak occurred in large-scale pipe network. The developed algorithm is model-free approach to detection and location of leak in water pipe networks.

We propose consensus algorithm based mathematical model of leak dynamics. Modeling the flow dynamics of leaks in water pipe networks is an extremely difficult problem due to the complex entangled network structure and hydraulic phenomenon. We propose a fundamental model for negative pressure wave dynamics of leaks in water pipe networks based on a consensus algorithm and water hammer theory. The resulting model is a simple and linearly interconnected model in the network even though the dynamics of water pipe networks has a considerable complexity. The model is then validated using experimental data obtained from a real water pipe network. A comparative study demonstrates that the proposed model can describe the real system with high qualitative and quantitative accuracy and that it can be used to develop a model-based leak detection and location algorithm based on the state estimation approach.

Using the developed model, we develop a fault detection and location algorithm based on state estimation in water pipe networks. The detection algorithm is based on cooperative $\mathscr{H}_\infty$-estimation for large-scale interconnected linear systems. To show applicability of the proposed model, we apply distributed and cooperative estimation with $\mathscr{H}_\infty$-performance to the developed model. The estimation result demonstrates the consensus algorithm based pipe network model can be potentially used for leak detection and location with state estimation method. $\mathscr{H}_\infty$-based design provides guaranteed performance with respect to model and measurement disturbances. Also, we propose cooperative Kalman filter of large-scale network systems. Basic concepts are based on cooperative $\mathscr{H}_\infty$-estimation used for detection and location. The proposed cooperative Kalman filter can show fully decentralized or fully distributed state estimation performance depending on parameter selection. It is demonstrated using large-scale chemical process network.

We finally propose a cooperative model predictive control of large-scale process networks based on the same concepts and ideas used to develop cooperative state estimation. Important properties of stability, optimality, local controllability, and scalability are also proved. When the developed cooperative MPC is applied to chemical process network composed of three process units, it shows performance between decentralized and distributed manners. We also show that the proposed cooperative MPC is the same with centralized MPC under certain condition.

Keywords: Water pipe networks, Large-scale process, Network systems, Cooperative Kalman filter, Cooperative model predictive control

Student Number: 2013-30285

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Large-scale integrated network systems are becoming more and more important as the need for higher energy efficiency and productivity increases in chemical process industry. In this thesis, we deal with estimation and control of large-scale systems and related problems. Large-scale network systems include integrated chemical processes, pipe distribution systems, sensor networks, and biological networks and we mainly focus on water pipe networks and chemical process networks. In this chapter, background and motivation of this research are introduced and preliminaries needed to explain main results are presented, followed by contribution and outline of the thesis.

1.1 Background and Motivation

Large-scale process networks are very complex and intricate systems and cast many problems related to control and estimation. Process networks are generally defined as a system where comprise of agent, node, or subsystems having their own dynamics and interacting with each other. Usually these network systems are large-scale and many realistic problems can be considered as network systems.

Process network examples include chemical plant, recycle network, pipe network, sensor network, and biological network. Most systems are large-scale and complex. We mainly deal with water pipe networks and chemical processes with material and energy recycles. Water pipe networks are particularly more complex than other network systems because it is densely structured and composed of many pipelines. Also, chemical plant is required to have more recycle streams for high efficiency and cost reduction. Therefore, these systems are difficult to maintain and manage due to limited sensors and complex dynamics.

For these reasons, major issues arise in large-scale process networks and are listed below.

- It is difficult to detect and diagnose a fault in complex network systems, especially when using only measurement information without a rigorous mathematical model.
- It is difficult to estimate the state variables in large-scale process networks with limited sensors.
- It is difficult to control the dynamic process in large-scale process networks when each subsystems are integrated and interacted.

To address these problems, some basic and important concepts should be defined and contemplated and they are introduced in the following section.

1.2 Preliminaries

Preliminaries and basic concepts are introduced mainly on network theory, consensus algorithm, state estimation and model predictive control for large-scale process networks.

1.2.1 Network topology

A process network can be represented by a directed graph in which the nodes of the graph correspond to process units and junctions. The edges of the graph correspond to connecting streams or pipelines. The mathematical model of the water pipe network is based on a directed, unweighted graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ that describes the network topology between individual nodes. $\mathcal{N}$ is the set of nodes, including virtual nodes on the pipeline or junctions,

$$\mathcal{N} = \mathcal{N}_{\text{node}} \cup \mathcal{N}_{\text{junction}} = \{n_1, \cdots, n_N\} \tag{1.1}$$

where $n_k \in \mathcal{N}$ $(k = 1, \cdots, N)$ represents the k-th node. $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges and represents pipe segments between nodes or pumps,

$$\mathcal{A} = \mathcal{A}_{\text{pipe seg}} \cup \mathcal{A}_{\text{pump}} = \{(n_k, n_j) | k = 1, \cdots, N; j \in \mathcal{N}_k\} \tag{1.2}$$

where $\mathcal{N}_k = \{j : (n_j, n_k) \in \mathcal{A}\}$ is the set of nodes that the node k receives information from and called the neighborhood of the node k. Eq. (1.2) models the information or physical flow, i.e., the k-th node is coupled to the j-th node if and only if $(n_j, n_k) \in \mathcal{A}$. The flow in an arc (n_j, n_k) is defined to be positive when it is directed from j to k; otherwise, it is negative or possibly zero. The flow is always nonnegative in arcs where only one direction is possible.

1.2.2 Consensus algorithm

When multiple nodes agree on the value of a variable of interest, they are said to have reached *consensus*. To achieve consensus, there must be a shared variable of interest, called the information state, as well as appropriate algorithmic methods for negotiating to reach consensus on the value of that variable, called *consensus algorithms*. Continuous-time and discrete-time consensus algorithms are presented as follows.

1.2.2.1 Continuous-time consensus algorithm [1]

A dynamic graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ is a graph in which the set of edges $\mathcal{E}(t)$ and the adjacency matrix $\mathcal{A}(t)$ are time-varying. Clearly, the set of neighbors $\mathcal{N}_i(t)$ of every agent in a dynamic graph is a time-varying set as well. It is shown that the linear system

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(t) - x_i(t)) \tag{1.3}$$

is a distributed consensus algorithm, i.e., guarantees convergence to a collective decision via local interagent interactions. Assuming that the graph is undirected ($a_{ij} = a_{ji}$ for all i, j), it follows that the sum of the state of all nodes is an invariant quantity, or $\sum_i \hat{x}_i = 0$. In particular, applying this condition twice at times $t = 0$ and $t = \infty$ gives the following result

$$\alpha = \frac{1}{n} \sum_i x_i(0) \tag{1.4}$$

In other words, if a consensus is asymptotically reached, then necessarily the collective decision is equal to the average of the initial state of all nodes. A consensus algorithm with this specific invariance property is called an average-consensus algorithm and has broad applications in distributed computing on networks (e.g., sensor fusion in sensor networks). The dynamics of system Eq. 1.3 can be expressed in a compact form as

$$\dot{x} = -Lx \tag{1.5}$$

where L is known as the *graph Laplacian* of G. The graph Laplacian is defined as

$$L = D - A \tag{1.6}$$

where $D = \mathrm{diag}(d_1, \cdots, d_n)$ is the degree matrix of G with elements $d_i = \sum_{ji} a_{ij}$ and zero off-diagonal elements. By definition, L has a right eigenvector of $\mathbf{1}$ associated with the zero eigenvalue because of the identity $L\mathbf{1} = 0$.

1.2.2.2 Discrete-time consensus algorithm [1]

An iterative form of the consensus algorithm can be stated as follows in discrete-time as follows

$$x_i(k+1) = x_i(k) + \varepsilon \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(k) - x_i(k)) \tag{1.7}$$

The discrete-time collective dynamics of the networks under this algorithm can be written as

$$x(k+1) = Px \tag{1.8}$$

with $P = I - \varepsilon L$ (I is the identity matrix) and $\varepsilon > 0$ is the step size. In general, $P = \exp(-\varepsilon L)$ and the algorithm in Eq. 1.7 is a special case that only uses communication with first-order neighbors. We refer to P as *Perron matrix* of a graph G with parameter ε.

1.2.3 State Estimation for large-scale networks

State estimation and control in large-scale process networks are considered as a very complicated problems and many efforts have been made by researchers in the past decades. Decentralize and distributed methodologies have been adopted in literatures. We classify the state estimation methods as centralized, decentralized, distributed, and cooperative strategies for large-scale systems. Cooperative strategy has been rather recently developed to resolve inherent problems of conventional decentralized and distributed methods.

1.2.3.1 Centralized state estimation

Kalman filter (KF) is used as a representative state estimation method. The continuous-time Kalman filter can be summarized as follows. The continuous-time system dynamics and measurement equations are given as [2]

$$\begin{aligned} \dot{x} &= Ax + Bu + w \\ y &= Cx + v \\ w &\sim (0, Q_c) \\ v &\sim (0, R_c) \end{aligned} \tag{1.9}$$

Note that $w(t)$ and $v(t)$ are continuous-time white noise processes. The continuous-time Kalman filter equations are given as

$$\begin{aligned}
\hat{x}(0) &= E[x(0)] \\
P(0) &= E[\{x(0)-\hat{x}(0)\}\{x(0)-\hat{x}(0)\}^\top] \\
K &= PC^\top R_c^{-1} \\
\dot{\hat{x}} &= A\hat{x}+Bu+K(y-C\hat{x}) \\
\dot{P} &= -PC^\top R_c^{-1}CP+AP+PA^\top+Q_c
\end{aligned} \tag{1.10}$$

The centralized KF has a serious problem in large-scale systems because it significantly increases computational burden when the dimension of the state is very large, especially for calculating Kalman gain, K. To solve the computational problem, decentralized and distributed KF have been studied in the past decades.

1.2.3.2 Decentralized state estimation

In decentralized KF, the hierarchical strategy decomposes a large networks into separate subsystems within which measurement data aggregation, processing, and estimation can be carried out locally. Ahmed *et al.* investigates the problem of designing decentralized robust Kalman filters for sensor networks observing a physical process with parametric uncertainty [3]. Also, convergence propoerties of decentralizd KF is proposed in [4]. A new algorithm for decentralized state estimation is proposed in the form of a multi-agent network based on a synergy between local Kalman filters and a dynamic consensus strategy between the agents in [5].

The decentralized state estimation substantially reduces computational load, however, it does not reflect the process dynamics of other subsystems in a completely decentralized manner. Therefore, partially decentralized or distributed state estimation has been developed as described in the subsequent section.

1.2.3.3 Distributed state estimation

For distributed state estimation problem, the inherently asynchronous sensor network is comprised of a large number of sensor nodes with computing and wireless communication capabilities, where the nodes are spatially distributed to form a wireless ad hoc network and every node has its own notion of time. Each individual sensor in a sensor network locally estimates the system state from not only its own measurement but also its neighboring sensors' measurements according to the given topology.

Different from the traditional centralized filtering, an effective distributed estimation algorithm should be capable of handling two additional issues: 1) complicated coupling between the sensor nodes according to a given topology and 2) network-induced phenomena such as randomly varying nonlinearities and missing measurements.

The problem of distributed Kalman filtering for sensor networks is one of the most fundamental distributed estimation problems for scalable sensor fusion. The paper in [6] addresses the DKF problem by reducing it to two separate dynamic consensus problems in terms of weighted measurements and inverse-covariance matrices. A distributed Kalman filter is presented to estimate the state of a sparsely connected and large-scale dynamical system monitored by a network of N sensors [7]. The problem of distributed Kalman filtering and smoothing is studied, where a set of nodes is required to estimate the state of a linear dynamic system from in a collaborative manner. [8] The gossip interactive Kalman filter (GIKF) for distributed Kalman filtering for networked systems and sensor networks is presented [9] and the internal model average consensus estimator is applied to distributed Kalman filtering [10]. A state estimation problem over a large-scale sensor network with uncertain communication channel is addressed in [11]. By using adaptive channel status estimator and robust L_1-norm Kalman filter in design of the processor of the individual sensor node, they are incorporated into the consensus algorithm in order to achieve the robust distributed state estimation.

1.2.3.4 Cooperative state estimation

In recent study, a cooperative state estimation to guarantee $\mathscr{H}_\infty$-performance is developed in [12]. The cooperative estimator compensates conventional decentralized and distributed state estimation in that local detectability is not required and the estimator size does not grow with the system size.

1.2.4 Control for large-scale networks

Control algorithms for large-scale processes have been extensively studied as integrated and networked process systems are important in process industries for high efficiency and product quality. We mainly deal with a model predictive control (MPC). Centralized, decentralized and distributed methods of MPC will be described in the following sections.

1.2.4.1 Centralized control

If a standard MPC is used for large-scale systems in a centralized manner, the most critical problem is that the on-line optimization should be implemented within a sampling time. However, as a system becomes bigger, solving the optimization requires more computational time. Therefore, conventional MPC for large-scale systems has been evolved to solve optimization problems in the sampling instance.

1.2.4.2 Decentralized control

In decentralized control of network systems, the achievement of a global control task is obtained by the cooperation of many controllers, each one computing a subset of control commands individually under a possibly limited exchange of information with the other controllers. Compared to centralized schemes, while decentralized control has the disadvantage of inevitably leading to a loss of performance, it has a twofold technological advantage: (i) no need for a high-performance central processing unit per forming complex global control algorithms that take into account the overall system dynamics, replaced by several simpler units; (ii) all process measurements do not need to be conveyed to a single unit, therefore limiting the exchange of information between spatially distributed components of the process.

A decentralized formulation is presented for model predictive control of systems with coupled constraints. The single large planning optimization is divided into small subproblems, each planning only for the states of a particular subsystem [13]. Robust decentralized model predictive control is impemented for a team of cooperating uninhabited aerial vehicles (UAVs). The problem involves vehicles with independent dynamics but with coupled constraints to capture required cooperative behavior [14]. The design of decentralized receding horizon control (RHC) schemes is studied for decoupled systems where the cost function and constraints couple the dynamical behavior of the systems [15]. Also, a novel decentralized model predictive control (MPC) design approach is poposed in [16] for open-loop asymptotically stable processes whose dynamics are not necessarily decoupled. A decentralized model predictive control method based on a dual decomposition technique is proposed in [17]. A model predictive control problem for a system with multiple subsystems is formulated as a convex optimization problem.

Complex processes are naturally suitable to be controlled in a decentralized framework: centralized control solutions are often unfeasible in dealing with large scale plants and they are computationally prohibitive when the processes are too fast for the existing computational resources. In these cases, the resulting control problem is usually split into many smaller subproblems and the global requirements are guaranteed by means of a proper coordination. A coordination strategy based on a networked decentralized model predictive control is proposed in [18] for improving the global control performances. Also, a decentralized model predictive control scheme is proposed for large-scale dynamical processes subject to input constraints in [19]. The global model of the process is approximated as the decomposition of several (possibly overlapping) smaller models used for local predictions. A hierarchical and decentralised model predictive control (DMPC) strategy for drinking water networks (DWN) is proposed in

[20]. The DWN is partitioned into a set of subnetworks using a partitioning algorithm that makes use of the topology of the network, historic information about the actuator usage and heuristics.

1.2.4.3 Distributed control

Completely centralized control of large, networked systems is impractical. Completely decentralized control of such systems, on the other hand, frequently results in unacceptable control performance. In this article, a distributed MPC framework with guaranteed feasibility and nominal stability properties is described. All iterates generated by the proposed distributed MPC algorithm are feasible and the distributed controller, defined by terminating the algorithm at any intermediate iterate, stabilizes the closed-loop system. These considerations motivate the development of distributed control systems that utilize an array of controllers that carry out their calculations in separate processors yet they communicate to efficiently cooperate in achieving the closed-loop plant objectives. MPC is a natural control framework to deal with the design of coordinated, distributed control systems because of its ability to handle input and state constraints and predict the evolution of a system with time while accounting for the effect of asynchronous and delayed sampling, as well as because it can account for the actions of other actuators in computing the control action of a given set of control actuators in real-time.

Distributed model predictive control is presented in [21] focusing on i) the coordination of the optimization computations using iterative exchange of information and ii) the stability of the closed-loop system when information is exchanged only after each iteration. An efficient distributed model predictive control scheme is presented based on Nash optimality in [22], in which the on-line optimization of the whole system is decomposed into that of several small co-operative agents in distributed structures, thus it can significantly reduce computational complexity in model predictive control of large-scale systems. This article [23] extends existing concepts in linear model predictive control (MPC) to a unified, theoretical framework for distributed MPC with guaranteed nominal stability and performance properties. A distributed output feedback model predictive control framework with guaranteed nominal stability and performance properties is described in [24]. Distributed state estimation strategies are developed for supporting distributed output feedback MPC of large-scale systems, such as power systems. An implementable distributed MPC framework is described with guaranteed nominal stability and performance properties in [25]. The proposed distributed MPC framework consists of three main components (i) distributed estimator (ii) centralized/distributed target calculation (iii) distributed regulator. The problem of distributed control of dynamically coupled nonlinear systems that are subject to decoupled constraints is considered in [26]. Examples of such systems include certain large scale process control systems, chains of coupled oscillators and supply chain management systems. A distributed model predictive control framework is proposed in [27]. The physical plant structure and the plant mathematical model are used to partition the system into self-sufficient estimation and control nodes. A formulation for distributed model predictive control of systems with coupled constraints is proposed in [28]. The approach divides the single large planning optimization into smaller sub-problems, each planning only for the controls of a particular subsystem. A dual-based decomposition method, called here the proximal center method, is presented to solve distributed model predictive control problems for coupled dynamical systems but with decoupled cost and constraints [29]. This work [30] focuses on a class of nonlinear control problems that arise when new control systems which may use networked sensors and/or actuators are added to already operating control loops to improve closed-loop performance. A distributed model predictive control framework, suitable for controlling large-scale networked systems such as power systems, is presented in [31]. The overall system is decomposed into subsystems, each with its own MPC controller. The problem of controlling two linear systems coupled through the inputs is considered in [32] and a novel distributed model predictive control method is proposed based on game theory in which two different agents communicate in order to find a cooperative solution to the centralized control problem. Since hot-rolled strip laminar cooling (HSLC) process is a large-scale, nonlinear system, a distributed model predictive control framework is proposed for computational reason and enhancing the precision and flexibility of control system [33]. The overall system is divided into several interconnected subsystems and each subsystem is controlled by local model predictive control. Theory for distributed model predictive control is developed based on dual decomposition of the convex optimization problem that is solved in each time sample [34]. The process to be controlled is an interconnection of several subsystems, where each subsystem corresponds to a node in a graph. In this work [35], distributed model predictive control of large scale nonlinear process systems is developed in which several distinct sets of manipulated inputs are used to regulate the process. we propose two different distributed model predictive control architectures. This work [36] present an iterative distributed version of Han's parallel method for convex optimization that can be used for distributed model predictive control of industrial processes described by dynamically coupled linear systems. In this paper [37], the control of several subsystems coupled through the inputs by a set of independent agents that are able to communicate is considered. At each sampling time agents make proposals to improve an initial feasible solution on behalf of their local cost function, state and model. In this work [38], a distributed model predictive control scheme is proposed based on a cooperative game in which two different agents communicate in order to find a solution to the problem of controlling two constrained linear systems coupled through the inputs. A new distributed model predictive control method is introduced in [39], which is based on a novel distributed optimization algorithm, relying on a sensitivity-based coordination mechanism. A class of large scale systems, which is naturally divided into many smaller interacting subsystems, are usually controlled by a distributed or decentralized control framework. In this paper [40], a novel distributed model predictive control (MPC) is proposed for improving the performance of entire system. This paper [41] presents a novel distributed predictive control algorithm for linear discrete-time systems. This method enjoys the following properties: (i) state and input constraints can be considered; (ii) under mild assumptions, convergence of the closed loop control system is proved; (iii) it is not necessary for each subsystem to know the dynamical models of the other subsystems; (iv) the transmission of information is limited, in that each subsystem only needs the reference trajectories of the state variables of its neighbors. This work [42] focuses on iterative distributed model predictive control (DMPC) of large-scale nonlinear systems subject to asynchronous, delayed state feedback. In distributed model predictive control, where a centralized optimization problem is solved in distributed fashion using dual decomposition, it is important to keep the number of iterations in the solution algorithm small. In this technical note [43], a stopping condition to such distributed solution algorithms that is based on a novel adaptive constraint tightening approach is presented.

1.2.4.4 Cooperative control

Recent researches on cooperative model predictive control are provided in the following.

A cooperative distributed linear model predictive control strategy applicable to any finite number of subsystems satisfying a stabilizability condition is proposed in[44]. The control strategy has the following features: hard input constraints are satisfied; terminating the iteration of the distributed controllers prior to convergence retains closed-loop stability; in the limit of iterating to convergence, the control feedback is plantwide Pareto optimal and equivalent to the centralized control solution; no coordination layer is employed. A distributed controller is presented that can stabilize nonlinear systems in [45]. A novel nonlinear nonconvex optimizer is proposed that improves the objective function and is feasible at every iterate. A general framework is proposed for distributed model predictive control of discrete-time nonlinear systems with decoupled dynamics but subject to coupled constraints and a common cooperative task [46]. To ensure recursive feasibility and convergence to the desired cooperative goal, the systems optimize a local cost function in a sequential order, whereas only neighbor-to-neighbor communication is allowed. A cooperative distributed linear model predictive control strategy is proposed [47] for tracking changing setpoints, applicable to any finite number of subsystems. The proposed controller is able to drive the whole system to any admissible setpoint in an admissible way, ensuring feasibility under any change of setpoint. A cooperative, distributed form of MPC for linear systems subject to persistent, bounded disturbances is developed in [48]. The distributed control agents make decisions locally and communicate plans with each other. A cooperative distributed stochastic model predictive control algorithm is given for multiple dynamically decoupled subsystems with additive stochastic disturbances and coupled probabilistic constraints, for which states are not measurable [49]. Cooperation between subsystems is promoted by a scheme in which a local subsystem designs hypothetical plans for others in some cooperating set, and considers the weighted costs of these subsystems in its objective.

1.3 Contribution

Major contributions are built on the above concepts and theories and we enumerate major contributions of the thesis in the following as four categories.

- Propose a novel fault detection and diagnosis scheme based on CUSUM and DWT without a system model by using only measurements

- Develop a mathematical model of water pipe network based on consensus algorithm

- Develop a novel cooperative distributed state estimation based on Kalman filter of large-scale process networks

- Develop a novel cooperative distributed model predictive of large-scale process networks

These contributions will be discussed in detail throughout the thesis.

1.4 Outline

First, we propose a novel model-free fault detection and location of water pipe networks using a cumulative (CUSUM) and discrete wave transform (DWT). Then, a consensus algorithm is introduced and used to develop a network model where subsystems interact each other. In particular, if a network has faulty agent which is randomly generated, consensus algorithm should be modified. The modified consensus is applied to water pipe networks to model dynamics of water pipe networks when pipe faults such leak or burst are randomly occurred in the pipeline. Once we have a system model, we can apply well-developed model-based methods such as Kalman filter (KF). Therefore, a cooperative KF is proposed to estimate states and detect the fault in the network. The idea used in developing cooperative KF can be applied to model predictive control (MPC) for solving network system control problems which are more intricate than other systems.

In Chapter 2, model-free fault detection and location method of water pipe networks is presented. We use a well-known cumulative sum (CUSUM) algorithm and a discrete wavelet transform (DWT) to exactly detect and locate the fault. This algorithm does not require a system model and uses only the measurement of the system. However, if we have a mathematical model of the system, we can design a model based systematic estimator to detect and locate the fault.

In Chapter 3, a mathematical model of water pipe networks is presented using consensus algorithm and water hammer theory. The developed model has a form of large-scale interconnected linear systems.

From the satisfactory result of comparison between the proposed model and experimental data, we apply distributed and cooperative estimation algorithm to the developed model for the purpose of fault detection and location in Chapter 4. Even with the model and measurement noises (disturbances), the estimator can robustly find all the states of water pipe network. However, we find out that if a constant disturbance enters the system such as a leak flow in water pipe networks, the estimation error does not asymptotically approach to zero. From this observation, a novel distributed estimation algorithm is developed when there exist a constant disturbance and background noises. The proposed estimation can be used in large-scale process networks such as chemical plant with recycle because the recycle streams are often considered as a constant disturbance to overall network system.

By combining the developed estimation and model predictive control, we propose a distributed and cooperative estimation and control of large-scale process networks in Chapter 5.

Also, the overall summary and concluding remarks are provided in Chapter 6 including future work directions.

Chapter 2

Model-free Approach to Fault Detection and Location of Water Pipe Networks

In this chapter, we propose a model-free approach to fault detection and location of water pipe networks by using cumulative sum (CUSUM) and discrete wavelet transform (DWT). The proposed algorithm is validated with experimental data obtained from three different real pipe networks. When the network system model does not exist, the proposed algorithm can effectively detect and diagnose pipe faults.

2.1 Introduction

Pipeline networks are one of the largest infrastructures of industrial society. In particular, they are of great importance to water distribution systems by efficiently transporting water resources throughout cities, worldwide. When a leak or burst occurs in such a pipe network, it causes an associated loss of water, unnecessary energy usage, and additional treatment cost. The maintenance costs of repair and replacement give rise to a huge financial waste for society. Moreover, it is not only an economic issue but also an environmental and potentially a health and safety issue.

Although the leak indiscriminately refers to chronic long-term leak or sudden pipe burst, they should be distinguished because they are quite different phenomena and the transients for each event are significantly distinct. Not all leaks grow as a burst and many leaks would never be found by existing burst technologies. In this work, we only consider burst occasions showing obvious pressure reduction although some references still use the terms with confusion.

Over the past decades, extensive efforts of governments and academic communities have been dedicated to develop efficient burst detection and location strategies to immediately identify the source after the incident.

Conventional burst detection is mostly conducted by discovering the burst by visual inspection at the surface. To detect the burst more systematically and efficiently, computational methods using a pressure transient of the burst which shows distinct signal transition have been investigated over the last couple of decades [50, 51, 52, 53, 54]. Pressure transients that occur in pressurized pipes propagate back and forth, and carry information about the features of the pipe system. The recognition of these features allows for the identification of hydraulic phenomena, thus providing a potential tool for burst detection [55, 56, 57].

Among transient based techniques, a negative pressure wave based method is one of the most popular approaches. When a burst occurs along the pipeline, it first removes the confining pressure produced locally by the pipe wall, allowing an outward flow from the pipe. A consequence of which is the creation of a low-pressure water hammer wave that propagates from the burst location into the remainder of the system. Taking the pressure before the burst as a reference criterion, the wave generated by such a burst is called a negative pressure wave (NPW) [58]. To apply the NPW based approach, it requires the pipe rupture be quick and abrupt as burst because a slow leak does not generate a distinct pressure reduction signal. Therefore, NPW-based methods are most helpful for detection of pipe bursts.

Two popular methods to detect pressure changes carried by the NPW are a Cumulative sum (CUSUM) and a wavelet transform (WT). Misiunas *et al.* [59] propose the CUSUM based burst detection algorithm. Since the CUSUM gives a robust sum for data change, it can be a potential tool for the detection of signal change with considerable measurement noise, such as water pressure. On the other hand, the CUSUM is relatively slow to respond to large data shifts [60]. This is unfavorable characteristic for burst detection algorithm where an immediate detection is crucial for system accuracy.

The wavelet based methods have been intensively investigated in the detection of signals showing abrupt transition such as a spike or peak because the WT can extract detailed information of the signal through decomposition into several scales. The applications include electric, mechanical, and acoustic systems [61, 62, 63, 64]. The WT technique has been also broadly applied to water pipe systems to detect the burst or leak–some authors did not effectively distinguish leaks from bursts–because such signals show sudden transient changes [65, 66, 67, 68, 69]. Especially, Ferrante *et al.* [65] apply the continuous and discrete WT to detect hydraulic discontinuities in pipe system using different mother wavelets. Ferrante *et al.* [66] present the multi-scale products of the WT for leak-edge detection. Ahadi *et al.* [67] apply the WT to acoustic emission of water pipe to detect the leak and also propose the strategy to find a mother wavelet for the best signal localization. Srirangarajan *et al.* [69] introduce wavelet decomposition into the burst detection of pipe networks not pipelines.

The WT definitely provides excellent means for the detection of sudden signal transitions as presented in the previous works. It is because signal noises are suppressed and singularities of the signal are emphasized as the decomposition level of the WT increases. However, it also indicates there is a high possibility that the insignificant signal transitions could be wrongly emphasized as the significant ones. Water pipe network is particularly such a system since pressure wave signals have considerable amounts of minor signal changes. Therefore, there are high possibilities of false alarms in applying the WT to complex water pipe networks, whereas the CUSUM has less possibilities caused by small signal transitions. This is why we integrate the WT and CUSUM algorithms for effective detection of the burst.

Regarding pipe burst location, most of the previous studies are focused on pipelines [58, 70, 71], not pipe networks which are much more complex systems. It is because the conventional location strategy requires two pressure sensors for finding the burst site and it is not practical to install two sensors every pipeline in the case of complex pipe networks. Thus, applications are mainly concerned with oil or gas pipelines having a long length and for pipe networks such as water distribution systems, a new methodology is required. A few works investigate effective location strategies for the pipe networks using network nodes as possible burst positions, not requiring many pressure sensors [59, 69]. Based on this, the present work proposes a systematic network representation methodology by defining a node matrix to conveniently describe pipe networks and facilitate the location algorithm.

Consequently, techniques of online burst detection and location of water pipe networks are here developed using the NPW signal. Moreover, the developed system is validated with real field data obtained from simulated bursts by opening hydrant valves for simple and complex pipe networks and from real burst event, and conventional algorithms are also tested with the same data for comparison. With the reliable validation results, a software program has been developed based on the proposed algorithm.

The paper is organized as follows. In Section 2, the proposed detection and location algorithm is presented; and in Section 3, simple and complex pipe network test with which the system is validated are presented with detailed discussion. Validation results with real burst event are provided in Section 4, followed by concluding remarks in Section 5.

2.2 Detection Algorithm

The proposed burst detection and location algorithms consist of two parts: detection and location units. Each of the algorithms is described in the following subsections.

2.2.1 Noise filtering of raw pressure data

Raw pressure data contain considerable amounts of noise which should be removed before the detection process. Wavelet denoising is used in this work because other filtering schemes such as SMA can reduce the pressure signal scales used for WT and it is consistent for the WT based detection algorithm presented in the later subsection. Taking into consideration the denoising effect and data distortion, noise of the raw data are suppressed up to level 5.

2.2.2 Cumulative sum for global detection

In pipe networks, the NPW continually reflects and attenuates after its occurrence. However, only the first pressure drop signal is used to detect and locate the event and other reduced waves are not significant. We use a CUSUM chart to detect this pressure drop. The CUSUM chart is a statistical method to detect significant changes in the mean of a data sequence from its random background noise. The advantage of the CUSUM is that it gives a robust sum to the data noise.

In pipe burst detection, the negative CUSUM for a decrease is utilized because the pressure always drops when the burst happens and it is described as follows [72]

$$\begin{aligned} & T_0 = 0 \\ & T_n = \sum_{k=1}^{n} \left(x_k - \mu_0 + \frac{\nu_m}{2} \right) \\ & M_n = \max_{k=1,\cdots,n} T_k \\ & \text{Alarm when } \; M_n - T_n > \lambda \; \text{ and set } \; t_c = t \end{aligned} \tag{2.1}$$

where x_k is a single data value, μ_0 is a mean of a normal data set, ν_m is an *a priori* chosen minimum jump magnitude, and λ is the threshold. Two user parameters, ν_m and λ, affecting the detection performance are calculated by the strategy proposed by Choe [73]. The parameter selection criteria is based on the statistics of the raw signals and from it, ν_m and λ are set as 6σ and $3\sigma_T$, respectively, where σ is the standard deviation of the normal data distribution and σ_T is the standard deviation of T under the normal situation. They should be individually assigned for each sensor since the characteristics of the measured data are different between sensors.

When the decrease detector exceeds the threshold, λ, an alarm is issued and the alarm time is recorded as t_c.

Although the CUSUM is suitable for global detection for large process changes (for example, 3σ shifts) without being interrupted with noise or small transitions, the detection tends to be slow [60] because of its integral property. Slow detection eventually has a negative effect on the location accuracy in the application of pipe burst detection. Therefore, we combine the CUSUM with a multi-level discrete wavelet transform (DWT) so as to improve the location accuracy since the DWT can identify the precise event occurrence time owing to the immediate response to the signal transitions by its derivative property. Although two signal processing algorithms are used, they do not distort the original pressure signal twice because the CUSUM is only to find globally a data segment where the WT should be applied to. Therefore, it is the raw signal, not the integrated signal by the CUSUM that the WT is applied to and thus the actual time resolution is affected only by the WT implementation.

2.2.3 Discrete wavelet transform for local time correction

Given the global detection results of the CUSUM, t_c, the DWT is applied in an interval around t_c. The raw discrete signal can be decomposed into wavelet coefficients which are referred to approximation and detailed coefficients. If the signal is decomposed once, it is called a 1-level DWT and it can be further decomposed into multi-level coefficients.

In WT applications, it is important to choose a mother wavelet that is suitable for the signal of interest since it affects the results of the analysis [62, 67]. Because a wavelet coefficient represents the similarity between the signal and the basis function, it is reasonable to look for the mother wavelet that is similar to the burst transient [62]. For this reason, the Haar wavelet is proved to yield the best results for the burst detection applications. Its stepwise shape which resembles that of the sudden pressure drop of the burst event signal enhances the correlation between the mother wavelet and the pressure signal [65]. With this reason, most of works to detect the burst use the Haar wavelet as the mother wavelet [62, 74] and also in this work the Haar wavelet is used. Other wavelets such as 'db' or 'sym' families were also tested, however, they give an inappropriate signal extraction.

Given the Haar wavelet, the multi-level DWT is applied in the internal of t_c, and a new detection time, t_{w_m}, that exceeds the statistical threshold, λ_w, at the decomposition level m is found.

$$t_{w_m} = n_\lambda \times \frac{2^m}{f} \quad \text{such that} \quad W_\psi[n_\lambda] > \lambda_w \quad \text{for} \quad t_c - d \leq t < t_c + d \tag{2.2}$$

where W_ψ is the detailed coefficient, f is the frequency, and d is the time span. The DWT is performed within the interval $2d$ and d is chosen as 0.3 s in this work. The signal is transformed up to level 5 by considering the noise suppression effect and conservation of the featured signal. Although the featured signal is more prominent with higher levels, the data point interval increases by $T \times 2^m$ at level m

with the sampling instance, T, owing to the down-sampling property of the DWT. Therefore, a back propagation step is needed to elaborate the time interval [69]. The local maxima of the wavelet coefficients between t_{w_m} and the previous time index is defined as the final corrected detection time at level 1.

$$t_{w_1} = n_m \times \frac{2^1}{f} \quad \text{such that} \;\; n_m = \max_n W_\psi[n] \quad \text{for} \;\; t_{w_m} - \frac{2^m}{f} \leq t < t_{w_m} \tag{2.3}$$

Though the time interval, $T \times 2^1$, after the back propagation to level 1, cannot be the same as the sampling instance, T, it would be considered as acceptable if the sampling time of the sensor is small enough to not influence the location accuracy. As many of the newly detected times, t_{w_1}, are found as the sensor numbers and they are transmitted to the location system for burst location.

The DWT detection method apparently gives the improved burst detection results because of the instantaneous signal recognition while the CUSUM slowly responds. On the other hand, using the DWT alone in the burst detection of pipe networks involves risk of high possibilities of false alarms since the water pressure transient continuously varies and leads to frequent signal transitions. Therefore, we propose the CUSUM is implemented with the DWT. Fig. 2.1 illustrates justification of the proposition.

We apply two algorithms to the normal water pressure to compare the possibilities of false alarm. The wavelet coefficients incorrectly appear when no hydraulic event happens while the CUSUM shows a robustness to the data.

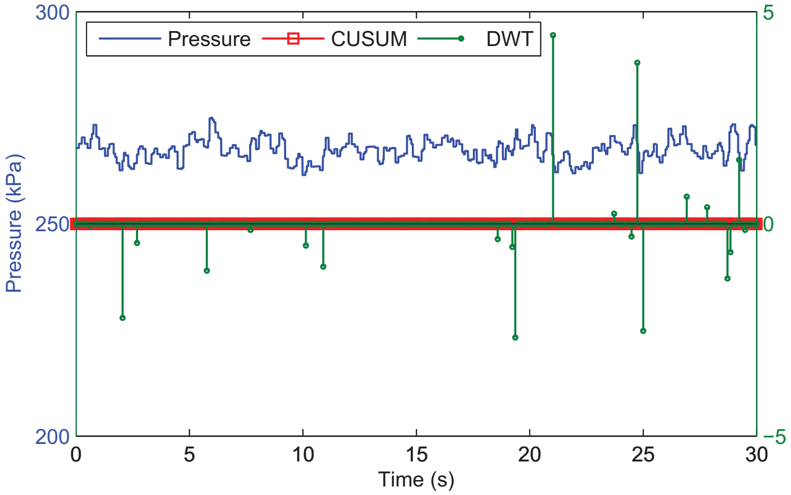


Figure 2.1: CUSUM and DWT detection results of normal pressure

2.3 Location Algorithm

After a fault is detected by the detection algorithm, the fault should be located by location algorithm. In this section, we propose fault location algorithm based on negative pressure wave (NPW) and optimization.

2.3.1 Negative pressure wave

When a burst occurs along the pipeline, fluid loss around the burst point leads to an abrupt pressure drop creating a negative pressure wave (NPW). Based on the NPW, the burst point, x, can be computed using arrival times at pressure sensors, the pipe length, and the pressure wave speed as follows [58, 75]

$$x = \frac{L + (t_1 - t_2)v}{2} \tag{2.4}$$

where L is the pipe length, v is the pressure wave speed, and t_1 and t_2 are the arrival times from the burst point to the sensors on both sides, respectively. Using this method, the burst location can be easily and rather precisely estimated provided time synchronization of the sensors is consistent since the time synchronization determines the time difference, $(t_1 - t_2)$.

The above NPW based burst location has been mostly applied to a long pipeline carrying oil or gas [76, 71]. This is because in such a long pipeline, placing two sensors on upstream and downstream of the pipeline is financially feasible. In a water pipe network, on the other hand, consisting of many pipelines to efficiently supply water, it is impractical to install two sensors every pipeline. Thus, the conventional burst location strategy cannot be applied and a new location strategy is required for water pipe networks. A methodology introducing nodes on the pipes is presented by Misiunas *et al* [59]. These nodes represent the possible burst candidate; and the burst site is chosen as the node with minimum value of an particular objective function. Here, we define the distance between two nodes as a link. If the link between neighboring nodes is long, then the location accuracy is limited to that extent. Thus, we propose a strategy to divide the link into the specific length. For instance, if the link is within 10 m, then the location error is theoretically within 10 m for a correct detection.

2.3.2 Node matrix

We define a node matrix **A** to describe the pipe network based on a graph data structure using the nodes. The node matrix **A**, describing the basic nodes of the network, is defined as

Definition.

$$\mathbf{A}(i,j)=\begin{cases}0 & \text{if } i=j \text{ for } i,j\in[1,N]\\ d_{i,j} & \text{if node } i \text{ and node } j \text{ are linked}\\ \infty & \text{otherwise}\end{cases}\tag{2.5}$$

where N is the node number, and $d_{i,j}$ is the distance between node i and node j and $d_{i,j}=d_{j,i}$. **A** is the $N\times N$ symmetric matrix. If nodes i and j are not linked, $\mathbf{A}(i,j)$ is set as ∞. We propose a new node matrix **A** having every element $\mathbf{A}(i,j)\leq 10$ after the node division. If $\mathbf{A}(m,n)>10$, $\mathbf{A}(m,n)$ is divided into additional nodes M with stepsize D. First, we define an $N\times M$ matrix **B** as

$$\mathbf{B}(i,j)=\begin{cases}D & \text{if } \mathbf{B}(m,1) \text{ or } \mathbf{B}(n,M)\\ \infty & \text{otherwise}\end{cases}\tag{2.6}$$

and **C** is the $M\times M$ symmetric matrix defined as

$$\mathbf{C}(i,j)=\begin{cases}0 & \text{if } i=j \text{ for } i,j\in[1,M]\\ D & \text{if } j=i+1 \text{ for } i\in[1,M-1]\\ \infty & \text{otherwise}\end{cases}\tag{2.7}$$

Then the final new $(N+M) \times (N+M)$ node matrix $\mathbf{A}'$ is

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \tag{2.8}$$

The node division process is repeated until every element of matrix **A** is less than 10 m. We can represent the entire network based on the GIS (Geographical information system) by using the node matrix **A**, and it is used for Dijkstra's minimum distance algorithm explained in the following. After obtaining **A**, we define an objective function to find the burst node.

2.3.3 Objective function

The rationale is that the difference of the detected times in sensors should be the same as the difference of the theoretically calculated times for the waves to propagate from the burst site to each sensor. The calculated time can be obtained from the wave speeds and distances between a node and each sensor. If the burst occurs at node i, the detected time difference and the calculated time difference are ideally the same as follows [59]

$$t_j - t_{k \neq j} = \tau_{i,j} - \tau_{i,k \neq j} \quad \text{for} \;\; 1 \leq i \leq N, \; 1 \leq j,k \leq S \tag{2.9}$$

where, t_j and $\tau_{i,j}$ are the detected time and calculated time, respectively; and N and S are the node number and sensor number, respectively. Based on this idea, the following objective function is formulated as

$$OF_i = \sum_{j=1}^{S-1} \sum_{k=j+1}^{S} [(t_j - t_k) - (\tau_{i,j} - \tau_{i,k})]^2 \quad \text{for} \quad 1 \leq i \leq N \qquad (2.10)$$

The objective function value of node *i* would be theoretically zero if *i* is the burst node. In reality, however, the value hardly becomes zero owing to various uncertainties. Accordingly, its minimal node is referred to as the burst node. With this approach, we can identify the node closest to an actual burst occurrence point. In complex network systems, however, it is more practical to report several candidate nodes as the burst site owing to the various system uncertainties. Therefore, at least three nodes are claimed as possible burst sites including the node with the minimum objective function value.

In finding the path between the burst and the sensor, it is noted that although many routes exist between them, we use the shortest path. It is reasonable since the first waves recorded by the sensors would travel the shortest path among possible routes and most quickly arrive at the sensors. Dijkstra's minimum distance algorithm [77] to find the shortest path from node *i* to all sensors is employed, given the node matrix **A**. Using the computed path and the wave speed, the theoretical travel time from node *i* to sensor *j*, $\tau_{i,j}$, can be found.

2.4 Integrated System

First, water pressure transients are measured at sensors installed on the pipe network, stored in a database, and transmitted to a monitoring unit. Facilities database includes pipe information such as pipe type, length, or diameter and the calculated node matrix **A**. The system deals with the 30-seconds data segment at once as a whole batch update for online implementation. The raw pressure data are denoised using a wavelet denosing technique. If the burst is detected as t_c by the CUSUM based global detection, then the local correction is implemented using the DWT in the interval around t_c. By thresholding the wavelet coefficients, the corrected time, t_{w_1}, is found through back propagation up to level 1. The detection results are transmitted to the NPW based location unit where the burst location is estimated by calculating the objective function. Dijkstra's minimum distance algorithm is applied to find the shortest path between node and sensor using the node matrix **A**. If no change is detected, then the next 30-seconds data window subsequently moves in. The final results are displayed on the user interface in real-time.

2.5 Experiments and Validations

Detection and location algorithm developed in the previous sections is validated with experimental data in this section. We obtained water pressure measurement from three different pipe networks, small-, medium-, and large-scale networks described below.

2.5.1 Small-scale pipe network with artificial faults

The developed burst detection and location system was validated with field data obtained from simple and moderately complex real water distribution systems. The experiment performed at area A in South Korea is rather simple without any household or factory nearby. Hence, the burst experiment could be controlled for sensors to capture the burst incident only.

2.5.1.1 Description of pipe network

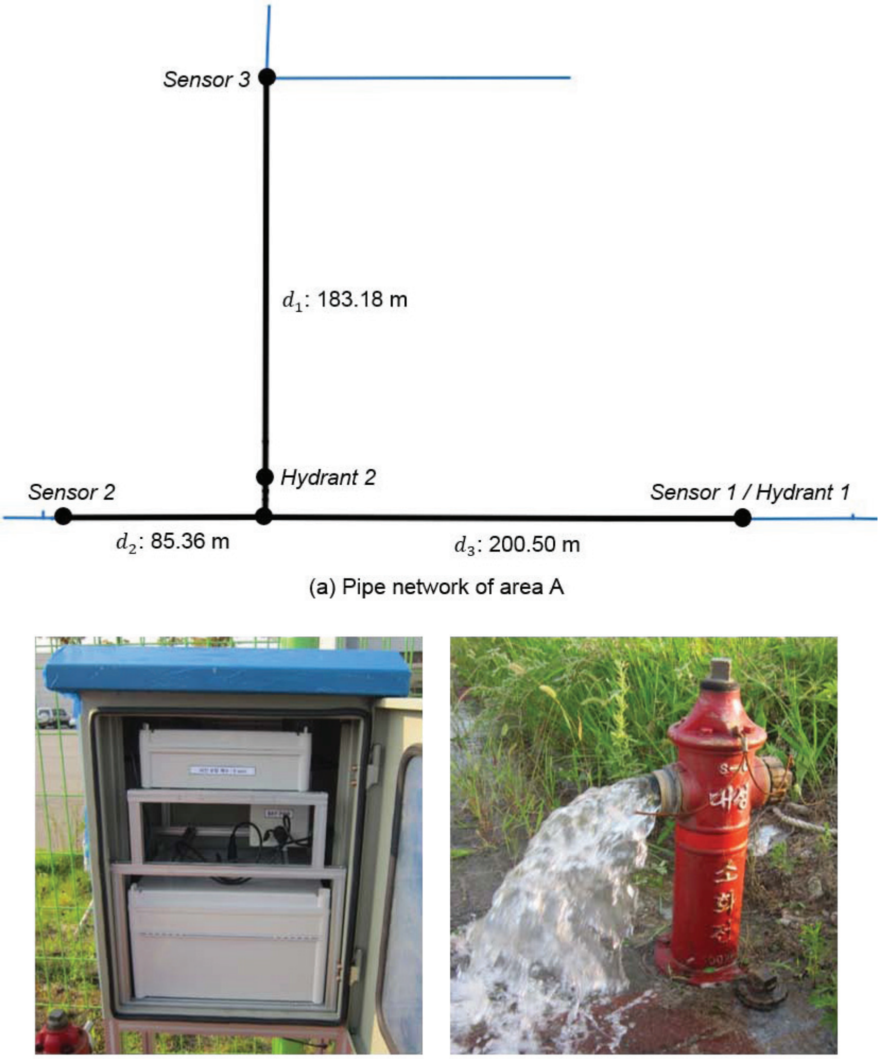
The experimental pipe networks of areas A and B include three pressure sensors and two fire hydrants as shown in Figs. 2.2(a) and 2.5.

Total pipe lengths covered by three sensors in each test area are 451.54 m and 2.577 km, respectively. Fire hydrants were used to generate a burst by opening valves. Fig. 2(b) shows a measurement station and water discharge of the hydrants in the experiments. The normal flow rate continuously varies from 4.17 to 8.33 L/s.

The basic network nodes are firstly selected as sensors and pipe connections. The number of network nodes in area A is four as illustrated in Fig. 2(a). Then, 4 × 4 node matrix **A** is expressed as follows

$$A = \begin{bmatrix} 0 & d_1 & \infty & \infty \\ d_1 & 0 & d_2 & d_3 \\ \infty & d_2 & 0 & \infty \\ \infty & d_3 & \infty & 0 \end{bmatrix}$$

Links over 10 m are divided by the node division strategy. After the node division, the final node matrix **A** has the dimension of 50×50 and the minimum and maximum step sizes are 9.4841 and 9.6410 m, respectively.



(a) Pipe network of area A

(b) Measurement station and water discharge

Figure 2.2: (a) Small-scale pipe network and (b) measurement station and water discharge

2.5.1.2 Measurement strategy and hardware system

Pressure transients are measured by three pressure sensors, named UNIK 5000 (GE). They are connected to a PC equipped with 12 bit A/D and D/A converters. The sampling frequency of the pressure sensor is 250 Hz; that is, the pressure is sampled every 0.004 s (or 4 ms) in this experiment. Pressure change by NPW should be captured in milliseconds because the pressure wave is very fast. Hardware equipment is uniquely designed to deal with the heavy data load; GPS based auto-synchronization technology to synchronize the measurement and transmission times and data compression technique to reduce communication cost. With this, the raw data could be compressed by up to 90%. A dual core high-speed microprocessor has been used to handle the massive data set for online measurement. The same measurement equipment and hardware were employed in two test areas. If frequency of sensor is higher than 250 Hz, the pressure change would be measured more accurately since the pressure wave is very fast. However, a technology does not exist to measure and save such big data in real-time.

2.5.1.3 Data acquisition and hydraulic behavior

Burst experiments were carried out by opening valves of hydrants installed on the pipe networks shown in Figs. 2(a) and 3. During the field tests, there was no other hydrant or pump operation around the area. The valve opening action may affect the pressure transient, however, it would be insignificant as compared with the influence by the sudden water release. Also, hydraulic phenomena at pipe junctions or valves always exist but, it appears as the noise.

Therefore, it is assumed that the experimental pressure transients were caused only by the sudden water discharge such as real burst.

Before we obtain the burst data, a steady state behavior of water pressure in area A was analyzed in advance as shown in Fig. 2.3(a)–(b). Under normal situations, the pressure was maintained at a constant value with background noise. After applying the filtering, the averaged pressures for Sensors 1, 2, and 3 were 250, 267, and 245 kPa, and the standard deviations were 1.86, 2.72, and 2.28, respectively. The statistical data obtained from the steady state analysis were used to calculate the CUSUM parameters, ν_m and λ. Fig. 2.3(c) shows the unsteady state behavior driven by the burst experiment. When the hydrant valve was open, sudden pressure drops were clearly observed and after the pressure dropped, the resulting NPW quickly disappeared with some fluctuations. We obtained a total of thirty burst experimental data, fifteen of which were generated at Hydrant 1 and the rest at Hydrant 2. The water flow was discharged about 3 L/s and the maximum normal flow rate of area A was 9.4 L/s. By a leak/burst criteria proposed by Ferrante *et al.* [78], the simulated burst can be considered as a detectable and huge accident.

2.5.1.4 Validation results

Using the experimental data, the performance of the proposed system was validated. First, Fig. 2.3(d) shows CUSUM results of the burst data from Hydrant 1 opening in test area A. The CUSUM signals appeared when the pressure was abruptly changed owing to the sudden water discharge and the local time correction step was implemented by the DWT. At level 5, the detection times for Sensors 1, 2,

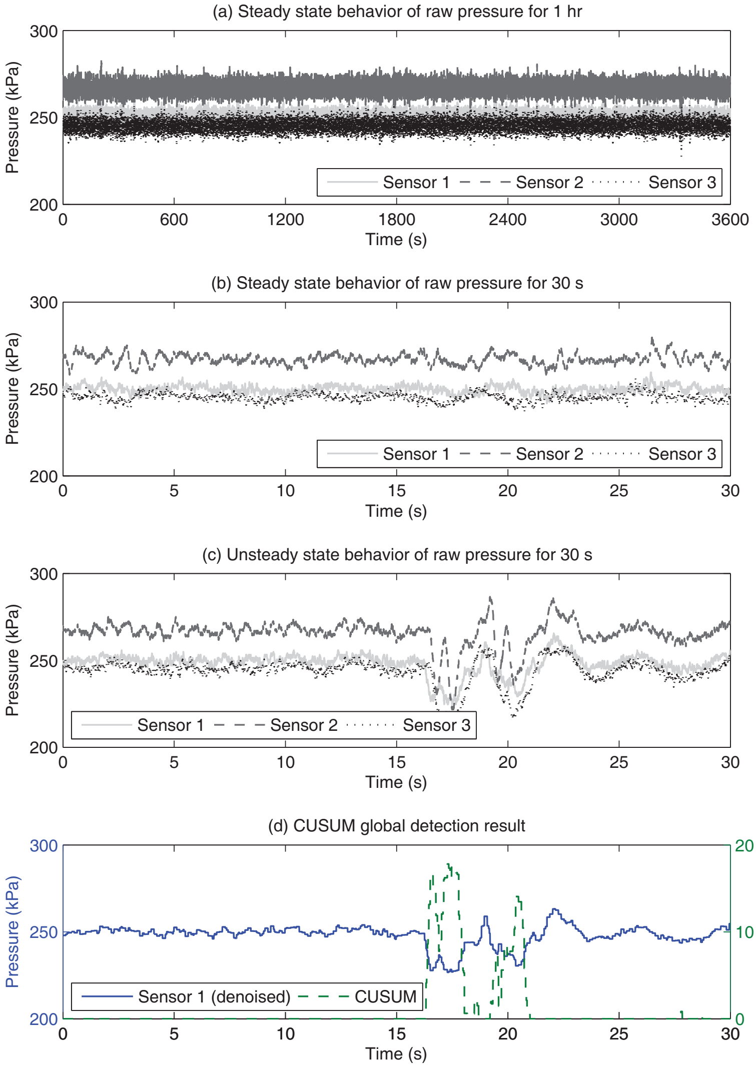


Figure 2.3: Steady state behavior of raw pressure for (a) 1 hr and (b) 30 s, (c) unsteady state behavior for 30 s, and (d) CUSUM global detection result

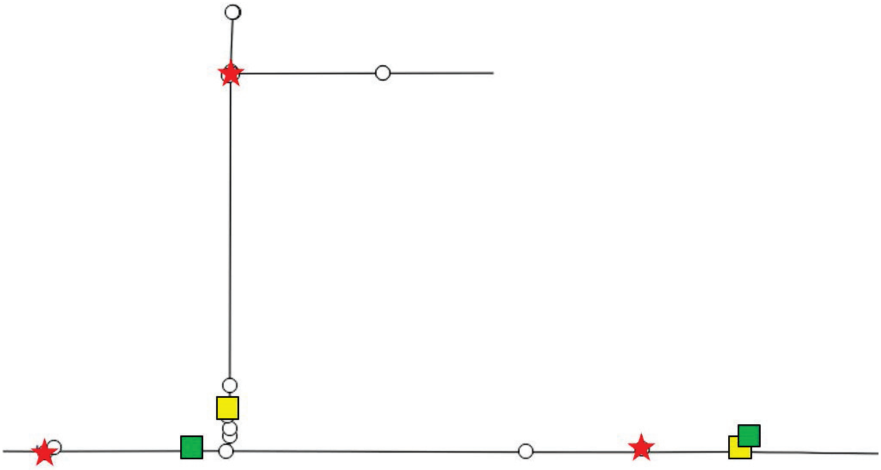


Figure 2.4: Validation result of small-scale network

and 3 were 16.332, 16.572, and 16.640 s; and they were propagated back to level 1 to elaborate the detected time, and adjusted to 16.328, 16.572, and 16.644 s, respectively. The detection results were used in the location unit to find out the burst site.

The burst node was located by minimizing the objective function in Eq. (2.10). A pressure wave speed indicates the speed of transient propagation in the pipeline, and may significantly affect the location accuracy. The wave speed can be measured in the field test by measuring the wave travel time from one sensor to another with information of distances between sensors. The measured wave speeds for 30 simulated waves were averaged to give 1103 m/s. Table 2.1 presents the burst candidates, the minimum node, and location errors between the minimum node and the actual burst site for all 30 events and the proposed algorithm was compared with the CUSUM algorithm. Table 2.1 shows that the integrated system could improve the location

Table 2.1: Burst candidate and minimum nodes and location errors between integrated algorithm and CUSUM only in test area A

Data set	Burst site	Candidate nodes	Minimum node	Location errors (m)	
				CUSUM and DWT	CUSUM
1	1	4, 49, 50	4	0	0
	1	4, 49, 50	4	0	0
	1	4, 49, 50	4	0	10
	2	20, 21, 22	21	2	21
	2	20, 21, 22	21	2	11
	2	18, 19, 20	19	21	2
2	1	4, 49, 50	4	0	0
	1	4, 49, 50	4	0	0
	1	4, 49, 50	4	0	0
	2	20, 21, 22	21	2	21
	2	18, 19, 20	19	21	21
	2	19, 20, 21	20	11	2
3	1	4, 49, 50	4	0	10
	1	4, 49, 50	4 ‘	0	19
	1	4, 49, 50	4	0	0
	2	18, 19, 20	19	21	31
	2	17, 18, 19	18	31	40
	2	18, 19, 20	19	21	31
4	1	4, 49, 50	4	0	0
	1	4, 49, 50	4	0	29
	1	4, 49, 50	4	0	0
	2	20, 21, 22	21	2	21
	2	18, 19, 20	19	21	29
5	1	4, 49, 50	4	0	0
	1	4, 49, 50	4	0	10
	1	4, 49, 50	4	0	0
	2	18, 19, 20	19	21	31
	2	18, 19, 20	19	21	40
	2	17, 18, 19	18	31	40
			Average error (m)	8	15
			Maximum error (m)	31	40

Table 2.2: Comparison with previous researches

Source	Filtering method	Detection method	False alarm	Missed alarm	Average error (m)	Maximum error (m)
			Area A			
This work	Wavelet denoising	Modified CUSUM & DWT	0	0	8	31
Misiunas *et al.*	Exponential filter	CUSUM	2	0	27	57
Srirangarajan *et al.*	Wavelet denoising	Wavelet decomposition	18	0	-	-
			Area B			
This work	Wavelet de-noising	Modified CUSUM & DWT	0	0	22	39
Misiunas *et al.*	Exponential filter	CUSUM	2	0	42	67
Srirangarajan *et al.*	Wavelet de-noising	Wavelet decomposition	7	0	-	-

accuracy, resulting in smaller average and maximum errors of 8 m and 31 m than using the CUSUM only. To the author's knowledge, the average error of less than 10 m in a real field is, at present, one of the outstanding performances reported in the literature.

To illustrate the effectiveness and robustness of the presented method, the validation results were also compared with previous researches [59, 69]. Misiunas *et al.* use exponential filter and CUSUM algorithm as filtering and detection methods, respectively, while Srirangarajan *et al.* use wavelet de-noising and wavelet decomposition. The monitoring performance was evaluated by the number of false alarms and missed alarms using one-hour steady state pressure data without any incident. Table 2.2 presents comparison results of three systems. The wavelet decomposition based monitoring algorithm gives the most frequent false alarms, 18 and 7 times for each network; whereas, two CUSUM based methods show much more robust property to data noise even in the moderately complex network. This is because the wavelet technique can amplify noise signals as significant featured data. Table 2.2 also presents location errors of two CUSUM based systems. In comparing location errors with the method in Misiunas *et al.* [59], the average and maximum location errors of the proposed approach are much smaller since the DWT time correction step was integrated. When only using the CUSUM, the location accuracy decreases owing to slow response of CUSUM algorithm.

2.5.2 Medium-scale pipe network with artificial faults

On the other hand, the complex pipe network located at area B includes hundreds of households and many business districts. Thus, the experiment is uncontrolled since other causes of pressure change exist in addition to the burst generation. For example, water usage is very unpredictable and irregular, which makes burst detection and location more difficult.

2.5.2.1 Description of pipe network

In area B, the initial nodes are chosen as shown in Fig. 3 and the initial node matrix **A** is 26×26 with the minimum and maximum elements of 2.8840 and 224.351 m; after node division, the final node matrix **A** becomes 265×265 with the minimum and maximum elements of 2.8840 and 9.9981 m.

2.5.2.2 Measurement strategy and hardware system

Pressure transients are measured by three pressure sensors, named UNIK 5000 (GE). They are connected to a PC equipped with 12 bit A/D and D/A converters. The sampling frequency of the pressure sensor is 250 Hz; that is, the pressure is sampled every 0.004 s in this experiment. Hardware equipment is uniquely designed to deal with the heavy data load; GPS based auto-synchronization technology to synchronize the measurement and transmission times and data com-

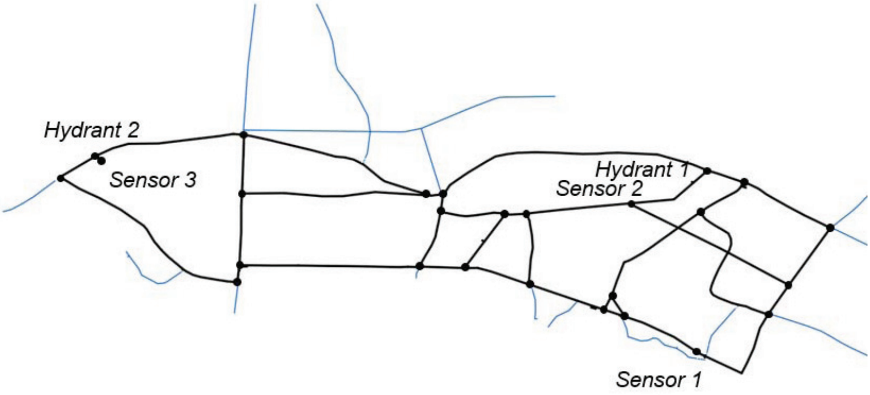


Figure 2.5: Medium-scale pipe network

pression technique to reduce communication cost. With this, the raw data could be compressed by up to 90%. A dual core high-speed microprocessor has been used to handle the massive data set for online measurement. The same measurement equipment and hardware were employed in two test areas.

2.5.2.3 Data acquisition and hydraulic behavior

Fig. 2.6(a)–(b) shows the steady state pressure of medium-scale pipe network. The average pressures for Sensors 1, 2, and 3 after denoised were 304, 424, and 387 kPa, and the standard deviations were 6.96, 3.33, and 4.49, respectively. In the test of area B, raw pressure data are not clean in comparison with the data of area A which is a simple network; and also, some biases exist especially at Sensor

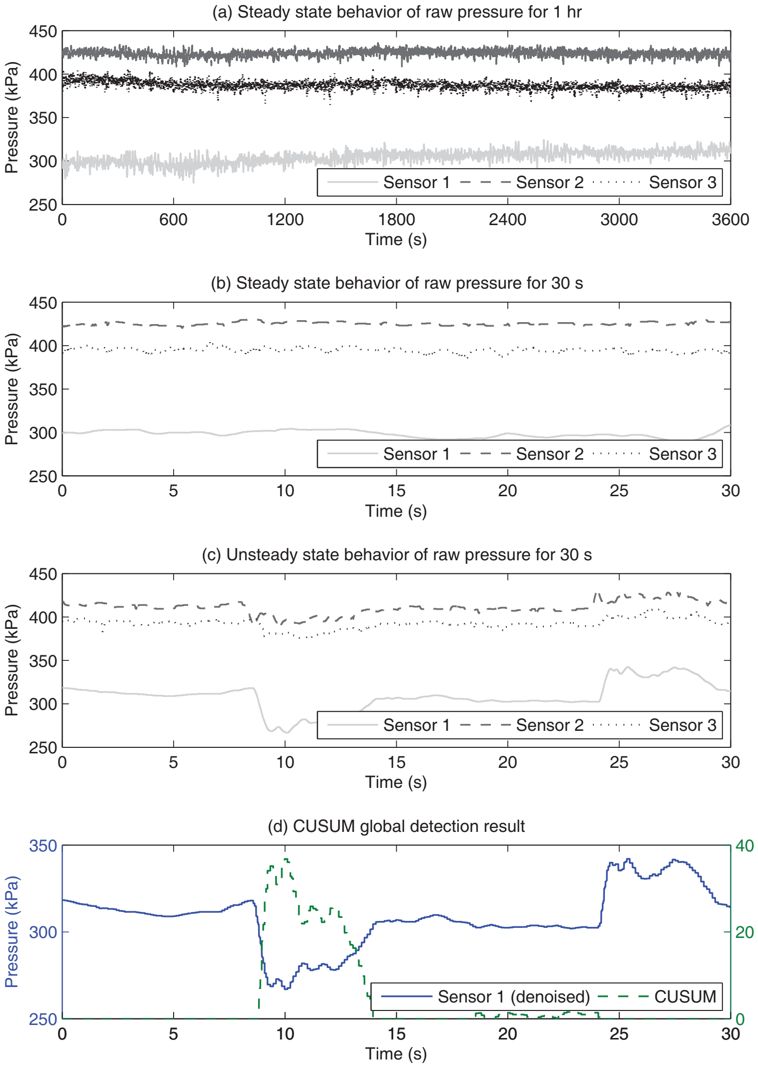


Figure 2.6: Steady state behavior of raw pressure for (a) 1 hr and (b) 30 s, (c) unsteady state behavior for 30 s, and (d) CUSUM global detection result

1. This may be attributable to the erratic water usage and hydraulic phenomena in this area. In this situation, the pressure can hardly be maintained at constant values. Fig. 2.6(c) shows the unsteady state behavior of water pressure driven by the burst generation. Overall, 13 burst experiments were performed, 11 experiments of which were performed at Hydrant 1 and two other experiments were at Hydrant 2. The water discharge flow rate in this test was 4 L/s and the maximum normal flow rate was 8.33 L/s. Also by the criteria, the simulation can be considered as a large burst event.

2.5.2.4 Validation results

Second, the CUSUM based global detection result for Sensor 1 of the first experiment in test area B is shown in Fig. 5(d). As the NPW was more obscure in rather complex network of area B, the CUSUM value was much smaller and the detection of the pressure transition was later. Thus, the detection accuracy was reduced. However, the location error can be considerably improved with the incorporation of the DWT as demonstrated in Table 2.3. The average and maximum errors are 22 m and 39 m, respectively, which are much more improved than using the CUSUM only. The pressure wave speed in area B was measured 1117 m/s by averaging the wave speeds for 13 simulated waves.

2.5.3 Large-scale pipe network with natural faults

The distribution network in large-scale pipe network C contains six sensors and the total length of the pipelines is 7266.44 m with 861 nodes.

Table 2.3: Burst candidate and minimum nodes and location errors between integrated algorithm and CUSUM only in medium-scale pipe network

Data set	Burst site	Candidate nodes	Minimum node	Location errors (m)	
				CUSUM and DWT	CUSUM
1	1	102, 103, 104	102	30	39
	1	102, 103, 104	103	39	39
	1	6, 95, 96	6	11	107
	1	102, 103, 104	103	39	58
	1	102, 103, 104	103	39	39
	1	6, 95, 96	6	11	58
	1	98, 99, 100	98	7	58
	1	102, 103, 104	103	39	39
	1	102, 103, 104	102	30	107
	1	102, 103, 104	103	39	39
	1	98, 99, 100	98	7	58
2	2	1, 2, 38	1	0	0
	2	1, 2, 38	1	0	0
			Average error (m)	22	49
			Maximum error (m)	39	107

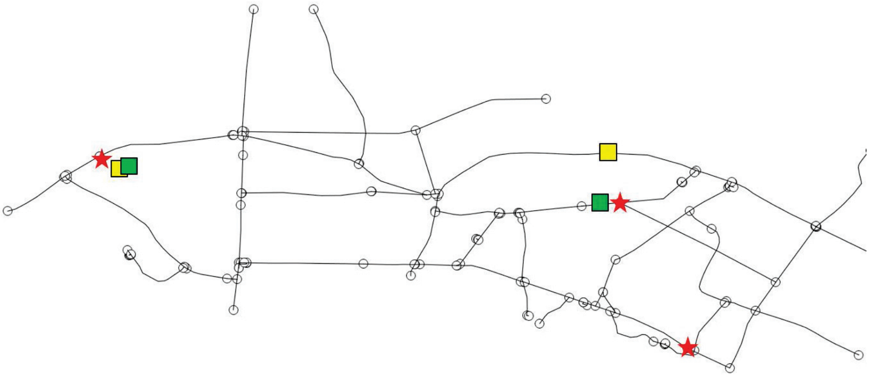


Figure 2.7: Validation result of medium-scale network

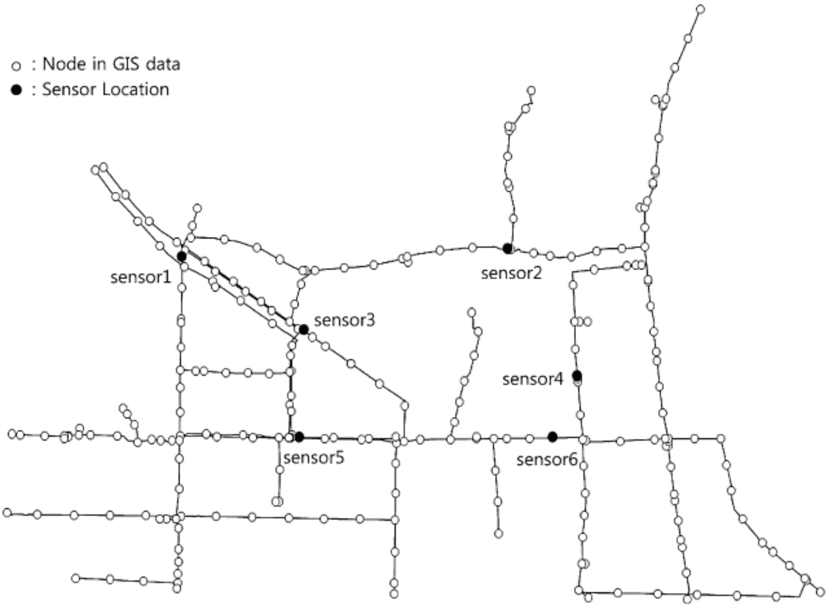


Figure 2.8: Large-scale pipe network

2.5.3.1 Description of pipe network

Large-scale pipe network is described in Fig. The initial nodes are chosen as shown in Fig. 3 and the initial node matrix **A** is 26×26 with the minimum and maximum elements of 2.8840 and 224.351 m; after node division, the final node matrix **A** becomes 265×265 with the minimum and maximum elements of 2.8840 and 9.9981 m. The burst occurring in Yeongwol and the pressure data were measured at 250 Hz. The pipeline network and the location of each sensor are shown in Fig. 2.8.

2.5.3.2 Measurement strategy and hardware system

Pressure transients are measured by three pressure sensors, named UNIK 5000 (GE). They are connected to a PC equipped with 12 bit A/D and D/A converters. The sampling frequency of the pressure sensor is 250 Hz; that is, the pressure is sampled every 0.004 s in this experiment. Hardware equipment is uniquely designed to deal with the heavy data load; GPS based auto-synchronization technology to synchronize the measurement and transmission times and data compression technique to reduce communication cost. With this, the raw data could be compressed by up to 90%. A dual core high-speed microprocessor has been used to handle the massive data set for online measurement. The same measurement equipment and hardware were employed in two test areas.

2.5.3.3 Data acquisition and hydraulic behavior

The raw pressure signal is shown in Fig. 2.9(a)–(b). As shown in Fig. 2.9(a)–(b), a pipe rupture causes a sudden pressure drop of the pipeline and the raw data include considerable amount of noise. Comparing with the experimental data in Figs. 2.3(c) and 2.6(c), it is noted that the sudden valve opening can similarly reproduce the real burst event in that they lead to substantial water loss and the pressure drop. The NPW caused by the burst was not recovered because the burst size was very large to be noticed quickly. Therefore, immediate detection and taking measures were required to stop the serious problem. The pressure transients increased rapidly after 5 min later the event happened and this stemmed from valve closing action when the pipe was finished to be repaired.

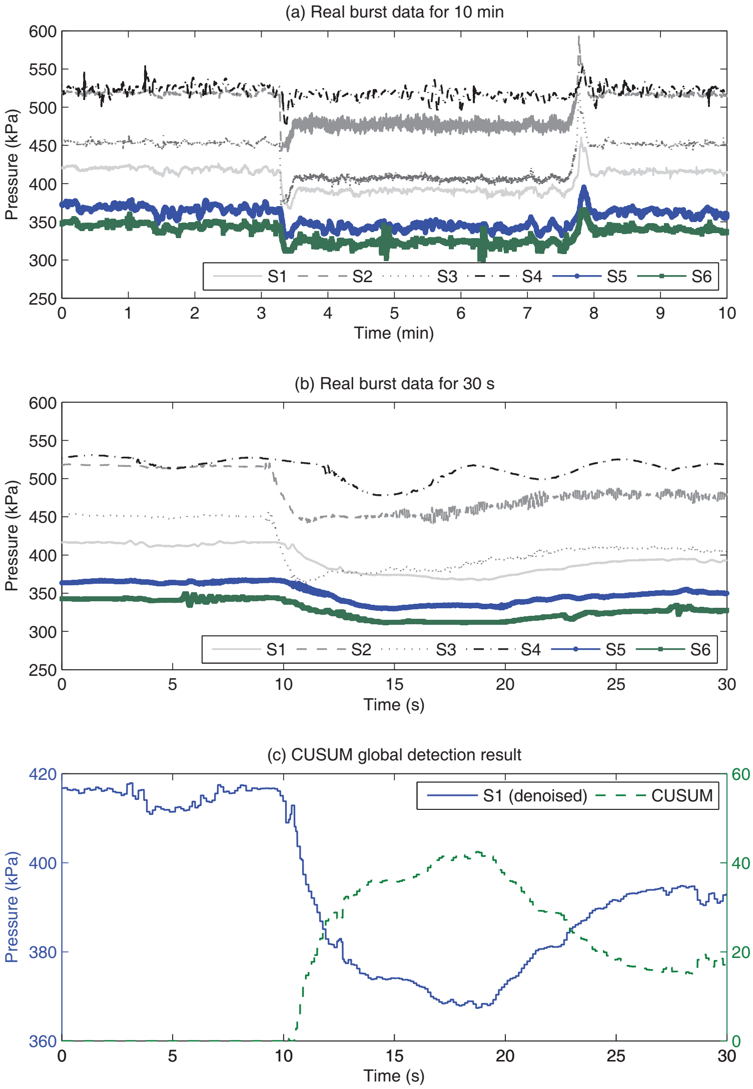


Figure 2.9: Real burst data for (a) 10 min and (b) 30 s and (c) CUSUM global detection result (S1 – S6 refer to Sensor 1 - Sensor 6, respectively.)

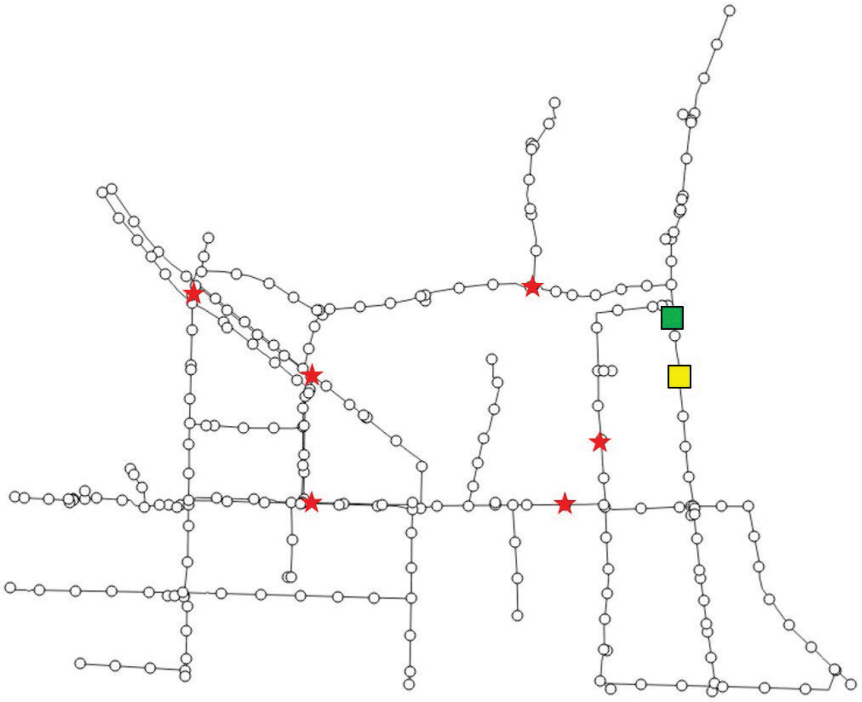


Figure 2.10: Validation result of large-scale pipe network

2.5.3.4 Validation results

The proposed burst detection and location algorithm was applied to the raw burst signal to verify it. Final detection times by the CUSUM and DWT were 10.112, 9.120, 9.216, 11.904, 11.520, and 10.816 s for Sensors 1–6, respectively, and it reported the possible burst site which was different from the actual burst occurrence location as much as 31 m. Two previous works showed worse performances on the real burst, 64 and 549 m. It is noted that the WT based detection system would give the false alarms in finding detection times and the location error would be wrongly estimated.

2.6 Limitations for applicability to complex networks

The ultimate goal of this work is to implement the proposed system online in real complex pipe networks. Until now, however, the algorithm has several limitations to be addressed. First, if the burst event is relatively small or the network is very complex, real pressure transient might not be so clear as available for effective detection because the pressure waves would be reflected, reduced, and attenuated. However, we observed through field test implementations many times that medium-large burst waves showed obvious signal changes regardless of other hydraulic phenomena. We consider it is possible to detect up to 15% of burst ratio in real field even allowing for other pipe actions. Moreover, the robustness of the objective function for burst location can be an issue because it may give several burst nodes or incorrect ones. Three pressure sensors at least should used for robust and reliable implementation and the accuracy would be improved with more sensors. However, if there are just two sensors in the network, several nodes are all claimed to the burst nodes and the actual burst node should be selected by direct inspection. Previous researches for pipe networks have adopted the objective function as the location method based on that it would theoretically give an accurate result if the detection times and the wave speed are precise. Despite the limitations for applicability to complex networks, the work has a significance in that it is validated with practical data of the real field having moderate complexity while the previous works have been mostly applied to simulation data or lab-scale experiments. The study has being steadily conducted by implementing a prototype software program based on the proposed algorithm to the real test bed.

2.7 Conclusions

In this work, we develop an online burst detection and location system of water pipe networks based on the CUSUM and the DWT algorithms, and propose a new node matrix to represent a pipe network with every link less than error bounds. In the monitoring unit, the CUSUM algorithm gives a robust sum to mean changes of data; but at the same time, it gives a slow detection, and thus deteriorates the detection accuracy. While the DWT may not be suitable for global event detection because of the high false alarm rate, the method allows the sudden transition of data to be exactly found. We combine these two techniques to take advantage of their properties, and obtain better location performance than the previous works. In the simple and controlled network, the average and maximum location errors were 8 and 31 m. In rather complex and uncontrolled network, the errors were 22 and 39 m, respectively. The efficacy of the developed algorithm was validated with both cases, and it shows a better result among those applied to real water supply systems up to the present. The pressure data obtained from the real burst accident were also used to verify the proposed system. In addition, a software program with the proposed algorithms has been completely developed and a pilot test is being carried out.

Chapter 3

Consensus Algorithm for Process Networks

In this chapter, we introduce a consensus algorithm to model network systems. In particular, flow dynamics model of water pipe networks is developed based on the consensus algorithm and the proposed model is validated with pressure measurement data obtained from three different pipe networks.

3.1 Introduction

Water pipe networks are one of the largest public infrastructures of industrial society and are used to efficiently transport water resources throughout cities. As a vital municipal system for water distribution, these networks require continuous strict and thorough maintenance. Nevertheless, flow leaks in water pipe networks frequently occur and cause serious problems in terms of safety and cost. Regarding cost, such leaks cause the loss of water, which results in an inevitable monetary loss and unnecessary energy usage because additional pumping energy is required to satisfy the specified carrying capacities. Furthermore, the high maintenance costs attributed to leaks, including rehabilitation and replacement, lead to a huge financial waste for society. Leaks are not only an economic issue but also a safety and potentially a health issue. This is because leaks may introduce contaminants into the pipeline at a relatively low pressure when the incident occurs, thus deteriorating water purity and quality. For these reasons, leaks in water pipe networks must be immediately identified and repaired after the incident to avoid unnecessary loss of resources.

Governments and academic communities have devoted considerable efforts to developing efficient leak detection and location strategies to reduce leak accidents to an economically optimal level in water pipe networks. There are numerous conventional methods for detecting leaks, including transient-based techniques using a negative pressure wave (NPW), inverse-transient analysis (ITA), and frequency domain techniques. Among them, the NPW-based method is one of the most popular approaches [79].

In recent years, model-based estimation techniques for leak detection have been intensively studied, and most of them have developed detection algorithms based on Kalman filter [80]. Emara-Shabaik *et al.* [81] proposed a nonlinear multiple model state estimation scheme using a modified extended Kalman filter to detect and diagnose leaks. This estimation technique, however, requires a dynamic process model of the system, and its estimation and detection performances depend on the corresponding model accuracy.

An appropriate leak dynamics model does not currently exist. The mathematical modeling of leak dynamics in water pipe networks is a difficult problem because it involves extremely large system dimensions and complexity caused by close-meshed networks and unpredictable events. From a practical perspective, this makes it difficult to generate a sufficiently accurate and reliable model in an acceptable error range. Therefore, the development of an appropriate model for leak dynamics in water pipe networks is important for application of the estimation to detect and locate leaks.

In general, water pressure is constantly maintained under normal conditions, but when a leak occurs, the water pressure suddenly decreases. Taking the pressure before the leak as a reference criterion, the wave generated by such a leak is called a negative pressure wave (NPW) [58]. When a leak occurs along the pipeline, it first removes the confining pressure produced locally by the pipe wall, allowing an outward flow from the pipe. Consequently, a low-pressure water hammer wave is generated that propagates from the location of the leak into the remainder of the system. In water pipe networks, the NPW continually reflects and attenuates after its formation by a leak. Therefore, the NPW dynamics should be taken into account since it represents a faulty state and the propagation of the leak signal in water pipe networks. However, this is the most difficult aspect in terms of model development.

Some early works have attempted to model hydraulic phenomena of water pipe networks. Because of the considerable complexity of water pipe networks, early mathematical approaches typically relied on substantially simplified network hydraulics by dropping all nonlinearities, which is often unacceptable in practice [82]. In this work [82], based on the simplification rage identified in the water network model, the model is linearized around a given point and redundant nodes are eliminated with Gauss-Jordan elimination. Then, the remaining nodes are re-linked with pipes according to the structure of the reduced network model. Burgschweiger *et al.* [83] proposed detailed models suitable for nonlinear optimization of daily network operations under reliable demand forecasts. Boussafeur-Lamoudi [84] proposed an automated simplification method of water network models. These works were all focused on simplifying water pipe network models that are mathematically sound but only applicable to small networks. For large networks, however, the models are still very difficult to solve and not accurate for practical application.

In this work, we propose a dynamic model of flow and NPW triggered by a pipe break in water pipe networks using a consensus algorithm and water hammer theory. A *consensus* means to reach an agreement regarding a certain quantity of interest that depends on the state of all agents (or dynamic systems) in networks. A *consensus algorithm* (or protocol) is an interaction rule that specifies the information exchange between an agent and all of its neighbors on the network [1]. Consensus algorithms have recently been extensively studied to describe coordination tasks in various areas of science and engineering, particularly in the context of cooperative control of multiple autonomous vehicles, formation control, decentralized task assignment, and sensor networks [85]. The flow dynamics in water pipe networks shows similar dynamics when discrete positions within the network are modelled as dynamic nodes that interact with their neighbors.

To describe the flow dynamics of a leak, we use water hammer theory and modify it to represent a negative pressure wave since the water hammer equations generally describe positive pressure waves. The mass balance equation of the water hammer is used to represent the relationship between the fluid velocity and water pressure in pipe networks. The significance of the proposed model lies in that the flow dynamics of complex water pipe networks can be represented by a simple and linear model within an acceptable error range.

The developed model is validated using experimental data obtained from field tests. A leak event is simulated by quickly opening a hydrant valve installed in the network. From the validation results, it is realized that the presented model can effectively describe the leak flow dynamics of the real water pipe network.

Using the proposed model and distributed state estimation scheme, we develop a leak detection and location algorithm of water pipe networks. The detection algorithm is based on distributed and cooperative $\mathscr{H}_\infty$-estimation for large-scale interconnected linear systems proposed in Wu *et al* [12].

Cooperative $\mathscr{H}_\infty$-estimation combines the benefits of both decentralized and distributed estimation in that cooperation between the local estimators is used to deal with lack of local detectability and the complexity of local estimators does not grow with the total size of the system in contrast to the existing distributed estimations. An important requirement of decentralized estimation approach is that the local subsystems are detectable from local measurements. In a distributed estimation, on the other hand, multiple estimators cooperate with each other and create an estimate of subsystem without local detectability. Instead, it gives rise to scalability issue where the order of the estimators grows with the size of the network. Therefore, the proposed method is referred to as *Cooperative estimation* which overcomes local detectability and scalability by only reproducing a desired subset of states for a local estimator. It is an $\mathscr{H}_\infty$-based design that provides guaranteed performance with respect to model and measurement disturbances. The methodology is generally applied to large-scale linear systems where subsystems may be physically interconnected and thus also applied to the developed flow dynamics model of water pipe networks.

The remainder of this paper is organized as follows. Preliminaries are provided in Section 2, and the procedure for modeling the leak dynamics of water pipe networks is presented in Section 3 based on a consensus algorithm and water hammer equation. The developed model is validated using experimental data in Section 4 and leak detection and location algorithm by application of distributed estimation to the developed model is presented in Section 5, followed by concluding remarks in Section 6.

3.2 Consensus Algorithm based Process Network Model

In this section, we introduce some basic notations and summarize useful concepts from graph theory to apply to water pipe networks. General consensus algorithm and properties are introduced.

3.2.1 Consensus in networks

We use a consensus algorithm to model the flow dynamics in water pipe networks because the nodes in water pipe networks reach a common value of state (on their own).

When multiple nodes (agents) agree on the value of a variable of interest, they are said to have reached *consensus*. To achieve consensus, there must be a shared variable of interest, called the information state, as well as appropriate algorithmic methods for negotiating to reach consensus on the value of that variable, called *consensus algorithms*. Consensus algorithms are designed to be distributed assuming only neighbor-to-neighbor interactions between nodes and limited information about other parts of the system [85].

If the effects of the interconnections are continuous-time, then the information state update of each node can be modeled using a differential equation. A scalar information state is updated by each node using a first-order differential equation [85]. Suppose that there are n nodes in the network. The most common continuous time consensus algorithm is given by

$$\dot{x}_i(t) = -\sum_{j=1}^{n} a_{ij}(t)[x_i(t) - x_j(t)], \quad i = 1, \cdots, n \tag{3.1}$$

where $a_{ij}(t)$ is the sum over the number of the neighbors of the i-th node and $x_i(t)$ is the information state of the i-th node at time t. Since the pipe network topology is time invariant, the gains a_{ij} are constant. Setting $a_{ij} = 0$ indicates that the node i is not coupled to the node j. A consequence of Eq. (3.1) is that the state $x_i(t)$ of the node i achieves to the same state of its neighbors.

3.3 Application to Water Pipe Networks

In this section, we propose a fundamental model of flow dynamics in water pipe networks when a pipe leak occurs. The proposed model is based on the consensus algorithm and classical water hammer theory, which will be described in the following subsections. Then, the dynamics at the leak point is described based on the concepts of the water hammer effect, and the complete model is summarized in the last subsection.

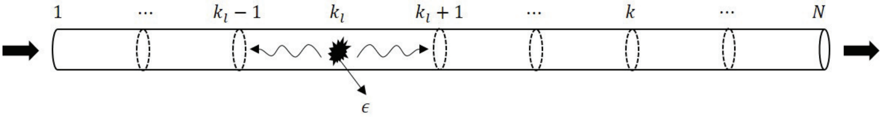


Figure 3.1: Pipeline nodes $k = 1, \cdots, N$ including a leak node k_l with velocity ε

3.3.1 Flow dynamics based on consensus algorithm

From the above consensus algorithm, we propose a flow dynamics model of water pipe networks when a leak occurs. The consensus algorithm can be applied to describe the propagation of a negative pressure wave in water pipe networks because the phenomena of the signal transfer to neighboring nodes are similar in both systems. Therefore, the consensus algorithm can be applied to fluid velocity as follows

$$\frac{dv_k}{dt} = K_k \sum_{j \in \mathcal{N}_k} (v_j - v_k), \quad k \in \mathcal{N} \tag{3.2}$$

where v_k is the fluid velocity at node k and K_k is the consensus gain of node k. The gain K_k can be varied with the network characteristics. Eq. 3.2 represents the steady-state velocity dynamics of the water pipe network.

When a leak occurs in the pipeline, the transient signal propagates in the upstream and downstream, and it will affect the velocities of adjacent nodes. Fig. 3.1 shows an example of a pipeline with a leak.

Eq. 3.2 should be modified to reflect the leak velocity at the nodes adjacent to the leak node, k_l. When the water flows from left to right in the pipeline, Eq. 3.2 is changed as follows: at node k which is on the left side of the leak k_l,

$$\frac{dv_k}{dt} = K_k \sum_{j \in \mathcal{N}_k} (v_j - v_k + \varepsilon), \quad k = k_l - 1 \tag{3.3}$$

and at node k, which is on the right side of the leak k_l,

$$\frac{dv_k}{dt} = K_k \sum_{j \in \mathcal{N}_k} (v_j - v_k - \varepsilon), \quad k = k_l + 1 \tag{3.4}$$

where ε is the leak velocity. When a leak happens, the fluid velocity decreases as much as ε at the leak point. At the $(k_l - 1)$-th node, the reduced velocity, ε, should be added as in Eq. 3.3 because the flow at this point is affected by the leak as much as $+\varepsilon$ and the velocity at the neighbor node j would be reduced by ε. Whereas, at the $(k_l + 1)$-th node, the reduced leak velocity should be subtracted as in Eq. 3.4 because the flow at this point is affected by the leak as much as $-\varepsilon$.

3.3.2 Water hammer theory

The water hammer phenomena have been extensively studied and well established since the early 1900s in this field. With the ever increasing importance of water hammer phenomena, many researchers have developed water hammer theories from first principles, and their combined efforts have resulted in the following classical mass and momentum equations for one-dimensional (1D) water hammer flows [86]

$$\frac{\partial V}{\partial x} + \frac{1}{\rho a^2} \frac{\partial P}{\partial t} = 0 \tag{3.5}$$

$$\frac{\partial V}{\partial t} + \frac{1}{\rho}\frac{\partial P}{\partial x} + \frac{4}{\rho D}\tau_w = 0 \tag{3.6}$$

where V is the cross-sectional average velocity, ρ is the fluid density, a is the acoustic water hammer wave speed, P is the water pressure, D is the pipe diameter, τ_w is the shear stress at the pipe wall, x is the spatial coordinate along the pipeline, and t is the temporal coordinate. The water hammer wave speed can be calculated using the following formula:

$$a = \sqrt{\frac{\frac{K}{\rho}}{1 + \phi\frac{K}{E}\frac{D}{e}}} \tag{3.7}$$

where K is the bulk modulus of elasticity of the fluid, E is Young's modulus of elasticity, e is the pipe wall thickness, and ϕ is a parameter that depends on the pipe anchoring.

From the literature [86], we know that friction in the pipe becomes negligible and τ_w can be safely set to zero in some cases. For example, wall friction is irrelevant as long as the simulation time is significantly smaller than $4L/a$. Then, the classical water hammer model given by Eq. 3.5 and 3.6 becomes

$$\frac{\partial V}{\partial x} + \frac{1}{\rho a^2}\frac{\partial P}{\partial t} = 0 \tag{3.8}$$

$$\frac{\partial V}{\partial t} + \frac{1}{\rho}\frac{\partial P}{\partial x} = 0 \tag{3.9}$$

By using the mass balance of Eq. 3.8, we can obtain the relationship between fluid velocity and water pressure in the pipe network. At node k, Eq. 3.8 is changed into the following form

$$\frac{\partial v_k}{\partial x} + \frac{1}{\rho a^2}\frac{\partial p_k}{\partial t} = 0, \quad k \in \mathcal{N} \tag{3.10}$$

The partial derivative of v_k with respect to x in Eq. 3.10 should be discretized to solve it, and we apply a backward finite difference method (FDM) since the variable does not considerably change in a short length in the pipeline. Using the first-order FDM, Eq. 3.10 can be expressed for pressure as follows

$$\frac{dp_k}{dt} = -L(v_k - v_{k-1}), \quad k \in \mathcal{N} \tag{3.11}$$

where $L = \rho a^2/\Delta x$ and Δx is the length between nodes.

When a leak occurs, Eq. 3.11 should be modified at the nodes adjacent to the leak node k_l,

$$\frac{dp_k}{dt} = -L(v_k - v_{k-1} - \varepsilon), \quad k = k_l \pm 1 \tag{3.12}$$

For the water pressure variable, it does not distinguish the flow direction because the negative pressure wave is considerably faster than the fluid velocity.

3.3.3 Dynamics at leak point

The flow dynamics at the leak point can be described by the water hammer effect explained in the previous section because the propagation of a pressure wave is similar except for the negativeness or positiveness of the pressure wave. Then, the pressure and flow at leak node k_l have the following dynamics

$$\frac{dp_{k_l}}{dt} = -L(v_{k_l} - v_{k_l-1}) \tag{3.13}$$

$$\frac{dv_{k_l}}{dt} = -M(p_{k_l} - p_{k_l-1}) \tag{3.14}$$

where $M = 1/\rho\Delta x$ and p_{k_l} and v_{k_l} are the pressure and velocity at the leak point, respectively.

3.3.4 Complete model

We summarize the complete model of the leak dynamics of water pipe networks in this section. The developed model consists of Eqs. 3.2–3.4 and Eqs. 3.11–3.14 and it is formulated as a state space model as follows

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u + \nu \tag{3.15}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \eta \tag{3.16}$$

where $\mathbf{x} = [p_1, v_1, \cdots, p_{k_l},\ v_{k_l}, \cdots, p_N, v_N]^\intercal \in \mathbb{R}^{2N}$ is the state of the system, $u = \varepsilon$ is the unknown input, $\mathbf{y} \in \mathbb{R}^m$ is the measurement vector, $\mathbf{A} \in \mathbb{R}^{2N\times 2N}$ is the system matrix, $\mathbf{B} \in \mathbb{R}^{2N\times 1}$ is the input matrix, $\mathbf{C} \in \mathbb{R}^{m\times 2N}$ is the measurement matrix, and $\nu \in \mathbb{R}^{2N}$ and $\eta \in \mathbb{R}^m$ are the noise vectors. The measurement $\mathbf{C}$ is defined by sensor number and location in the networks, with 1 for elements of $\mathbf{C}(1,k_1), \cdots, \mathbf{C}(N_s, k_{N_s})$ where N_s is the sensor number and zero for all other entries. The noise should be included in the model to portray the real system since the real water pipe network possesses many uncertainties. The system matrices, **A** and **B**, can be represented as

$$\mathbf{A} = \begin{bmatrix} 0 & L & 0 & -L & 0 & & \cdots & & 0 \\ 0 & -K_1 & 0 & K_1 & 0 & & \cdots & & 0 \\ & & & & \ddots & & & & \\ 0 & \cdots & 0 & L & 0 & -L & 0 & \cdots & 0 \\ 0 & \cdots & M & 0 & -M & 0 & 0 & \cdots & 0 \\ & & & & \ddots & & & & \\ 0 & & \cdots & & & 0 & L & 0 & -L \\ 0 & & \cdots & & & 0 & K_N & 0 & -K_N \end{bmatrix} \tag{3.17}$$

and

$$\mathbf{B}^{\mathsf{T}} = \begin{bmatrix} 0 & \cdots & 0 & K_{k_l-1} & 0 & -K_{k_l+1} & 0 & \cdots & 0 & L & 0 & L & 0 & \cdots & 0 \end{bmatrix}^{\mathsf{T}} \tag{3.18}$$

A and **B** can differ depending on the network topology and the location of the leak.

It is worthwhile to note that the presented model has a form of simple and linear interconnected equations even though the system has highly complex dynamics. This is attributed to the form of consensus algorithm applied to this system and the first-order FDM of water hammer equation. The first-order FDM is appropriate because the fluid velocity does not change much in the short length less than 10 m.

3.3.5 Experiment

In this section, the developed dynamic model is validated using experimental data obtained from a real water pipe network as explained in [87]. First, we introduce geometry information of the water pipe network and present steady and unsteady state water pressures measured from sensors in the test area. Second, the developed dynamic model is compared with the experimental data, and the related residuals and errors are calculated in the following subsection. The developed dynamic model was validated using the field data obtained from a real water pipe network in South Korea. Without any households or factories near the test area, the experiment could be controlled for sensors to capture the leak incident only.

Pipe networks shown in Fig. 2.2, 2.5, and 2.8 are used for validations. Although the pipe network is not as complex as a real-life water pipe network, it is sufficient to serve as a field test for validation since it includes sufficient pipe length and two junctions to characterize the network system. Arcs over 10 m are divided by the node division strategy proposed in [87]. After the node division, the final node matrix $\mathbf{A}'$ has dimensions of 50×50, and the minimum and maximum step sizes of arcs are 9.4841 and 9.6410 m, respectively .

During the field test, there was no other hydrant or pump operating around the area. The valve opening action may affect the pressure transient; however, it would be insignificant compared with the influence of the sudden water release. Additionally, hydraulic phenomena at pipe junctions or valves may always exist; however, such phenomena appear as the background noise of the system. Therefore, it is assumed that the pressure transient changes in the experiment were caused only by the sudden water discharge of the hydrant.

Fig. 3.2(a) and 3.2(b) show the steady state and unsteady state behaviors of water pressure for 30 s driven by the hydrant valve opening, respectively. In Fig. 3.2(a), the water pressure was maintained

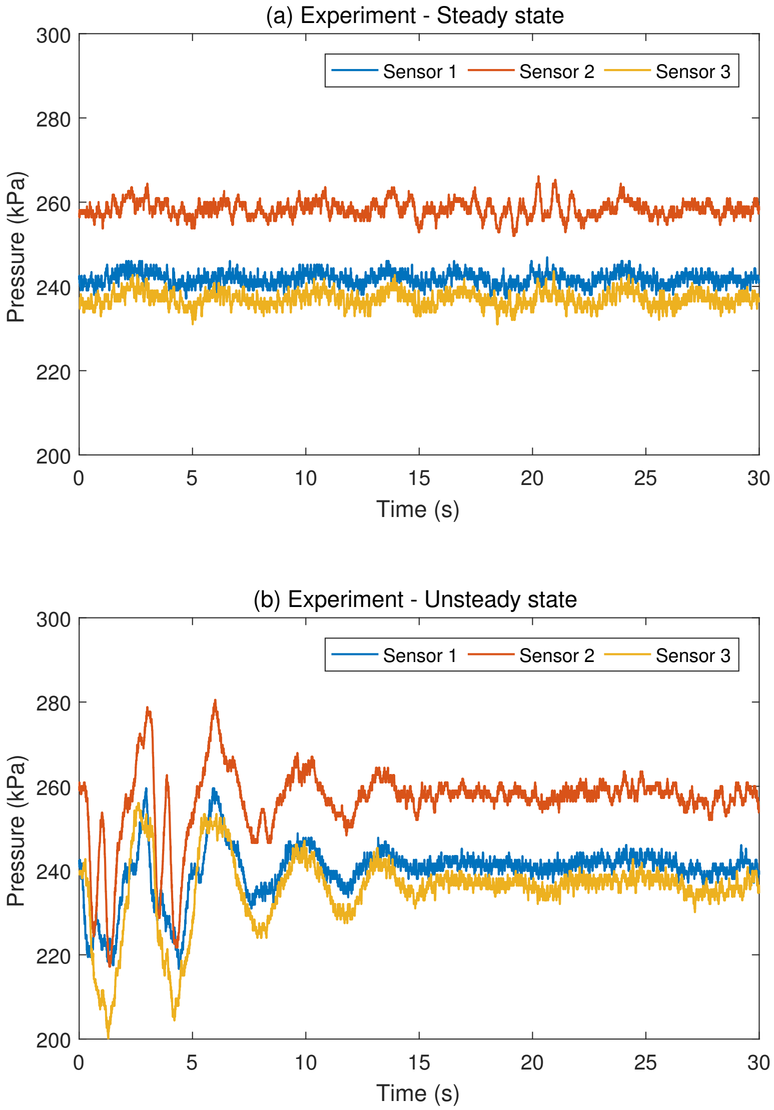


Figure 3.2: Experimental data of water pressure for (a) steady state and (b) unsteady state

at a constant value with background noise under normal conditions without any incidents. The averaged pressures for Sensors 1, 2, and 3 were 250, 267, and 245 kPa, and the variances were 3.46, 7.40, and 5.20, respectively.

The water pressure was maintained at a constant value at normal conditions, and when the hydrant valve was opened, it generated the NPW and sudden pressure drops were clearly observed, as shown in Fig. 3.2(b). After the pressure dropped, the resulting pressure wave quickly disappeared after several fluctuations.

We obtained a total of fifteen experimental data sets, which were generated at Hydrant 1. The water flow was discharged at approximately 3 L/s at Hydrant 1, and the maximum normal flow rate of this network was 9.4 L/s. The normal flow rate continuously varied from 4.17 to 8.33 L/s.

3.3.6 Validation

To simulate a complete model for this test area, we assign network nodes, construct system matrices, A, B, and C, according to the network topology and adjust model parameters. The total node number N of this network is 50, so the system dimension is $\mathbb{R}^{100}$, and the leak velocity, ε, is calculated as 0.38 m/s from cross sectional area of the pipeline. In addition, the system and measurement disturbances, ν and η, are added as white Gaussian noise (WGN) to mimic the effect of many random processes that occur in the network. Then, the simulation results are compared with the experimental data.

The parameter values in the model is presented in Table 3.1.

As in the experimental data shown in Fig. 3.2, the steady state

Table 3.1: Parameter values of the model

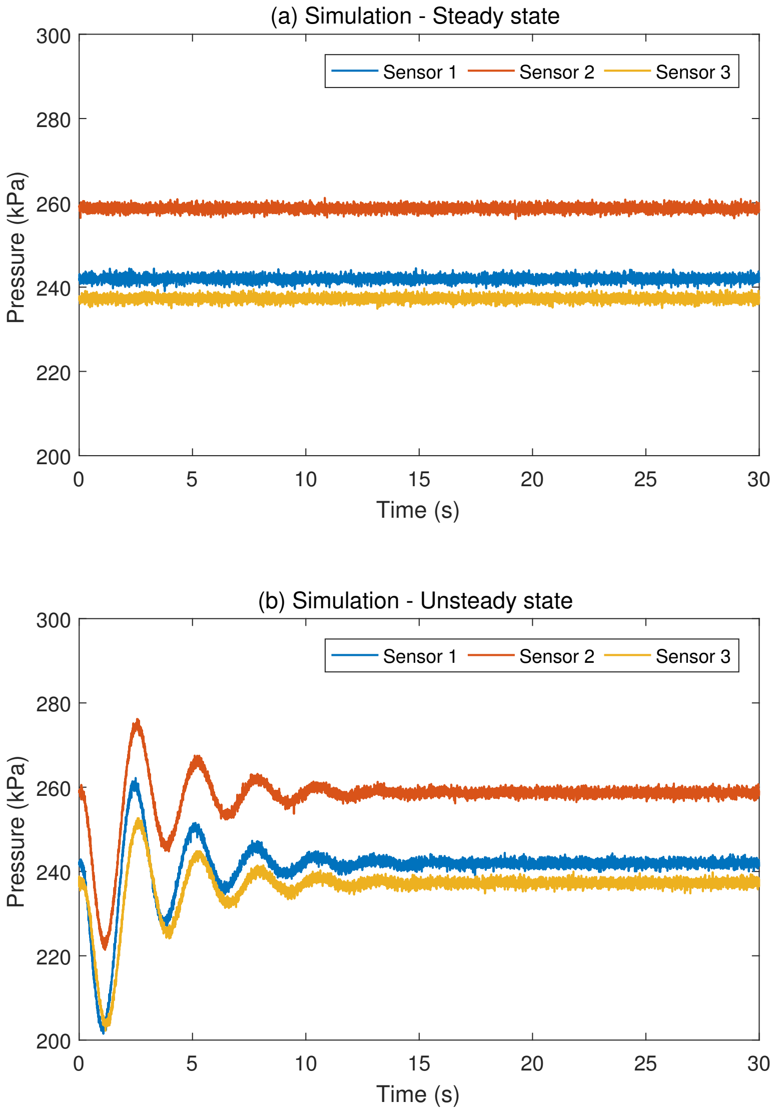
Parameters	Values
K	7
B_m	11
Δx	10
ε	2

and unsteady state behaviors of the water pressure are simulated using the proposed model, as shown in Fig. 3.3. The average pressure values of three sensors obtained from the measurement are employed in the simulation as initial values. In Fig. 3.3(a), the water pressures are constantly maintained with background noise under totally normal state without any leak as in Fig. 3.2(a).

In Fig. 3.3(b) with the leak simulation, it is observed that the pressure suddenly drops, and it is recovered with some fluctuations as in Fig. 3.2(b). However, the proposed model is an ideal case, and there is a difference between realistic and ideal systems.

If a sensor is located near the leak point, the pressure drop would be more obvious than ones at other sensors. To show that the proposed model can describe the time delay effect of the negative pressure wave on sensors, the initial parts of the pressure drop for both the experimental data and simulation results are enlarged as shown in Fig. 3.4. The pressure decreases in the order of Sensor 1, Sensor 2, and Sensor 3 for both the experimental and simulation results, and it corresponds with the network topology in Fig. **??** because Hydrant 1 is close to sensors in the order of Sensor 1, Sensor 2, and Sensor 3. Thus, the difference in accordance with the sensor location can also be represented by the proposed model.

We validated the model with 10 data sets obtained from the ex-



(a) Simulation - Steady state

(b) Simulation - Unsteady state

Figure 3.3: Simulation results of water pressure for (a) steady state and (b) unsteady state

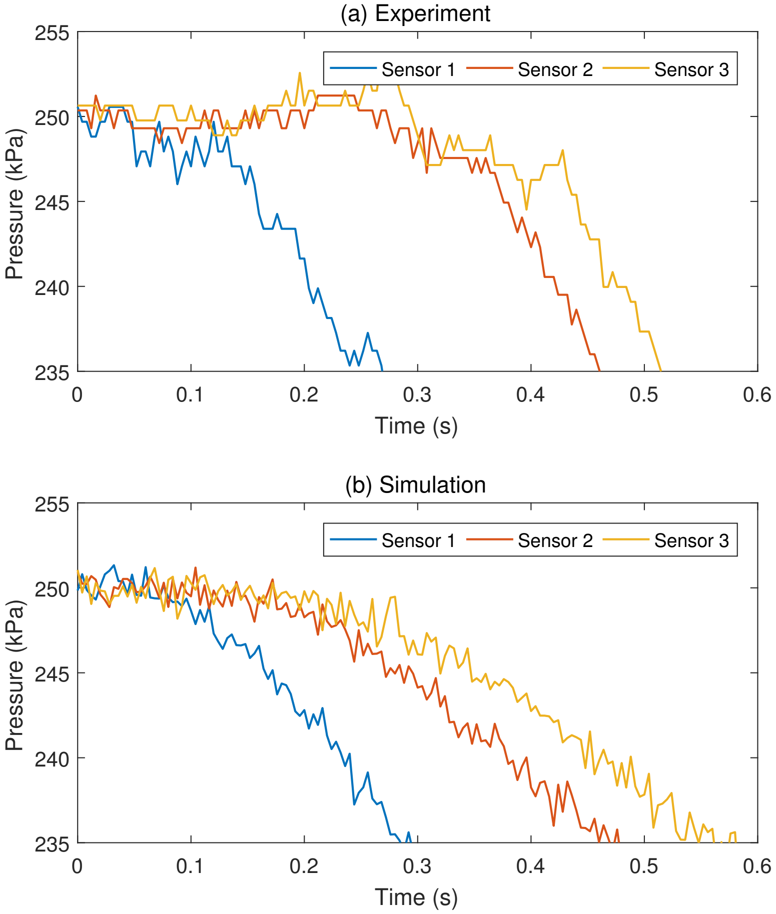


Figure 3.4: Enlarged pressure transient in the occurrence of a leak for (a) experiment and (b) simulation

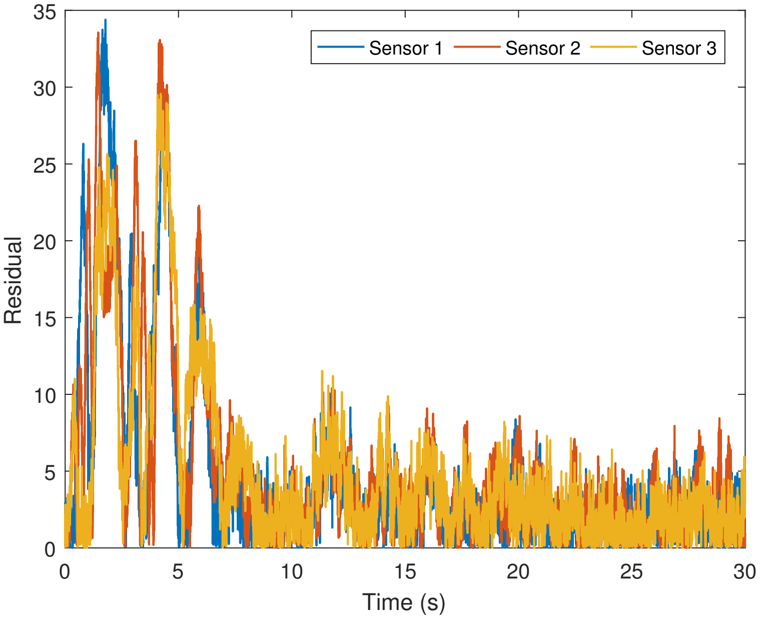


Figure 3.5: Residual of water pressure from experiment and simulation

periments. Fig. 3.5 shows comparative results of the developed model for the selected experimental data set. The residual errors for three sensors are less than 35, and the relative error is calculated to be 16%. As shown in Fig. 3.5, the fluctuation part causes the most severe deviation between the model and experiment. The average and maximum errors for 10 experimental data sets are calculated as 1.62% and 14.0%, respectively. From this result, although the fluctuation part generates a relatively large deviation from the experiment, the developed model shows satisfactory results with high qualitative and quantitative accuracy and it could be used to develop a leak detection and diagnosis algorithm with estimation methods that require the dynamic model of the system. We develop a leak detection and diagnosis algorithm using the proposed model and cooperative state estimation technique in the subsequent section.

The average and maximum errors for 10 experimental data sets are presented in Table 3.2. As shown, the average and maximum errors are within 2 and 20%, respectively. Although the fluctuation part generates a relatively large deviation from the experiment, the developed model shows satisfactory results, and it could be used to develop a leak detection and location algorithm with estimation methods that require the dynamic model of the system because if we have the system model, then the state estimation algorithms could be applied to the model to detect and diagnose faults. We will develop a leak detection and diagnosis algorithm using the proposed model and state estimation technique in a future work.

3.4 Conclusions

The fundamental model of NPW dynamics due to leaks in water pipe networks is developed based on a consensus algorithm and water hammer theory. The leak dynamics model of water pipe networks has hardly been considered in the literature thus far because of the complexity of the water pipe network structure and hydraulics. The developed model has a form of a simple and linear model of an interconnected network system. Then, it is satisfactorily validated using real experimental data obtained from a field test. The average and maximum errors between the model and experiment are within 2 and 20%, respectively, and a large portion of these errors are attributed to fluctuations of the NPW propagation through the network, which is difficult to precisely model. Although the model requires parame-

Table 3.2: Average and maximum errors from residuals of the experiment and simulated data for 10 data sets

Data set	Sensor number	Average error (%)	Maximum error (%)
1	1	1.86	10.4
	2	1.74	11.6
	3	1.51	12.1
2	1	1.76	15.6
	2	1.58	13.1
	3	1.60	12.4
3	1	1.64	15.7
	2	1.58	16.2
	3	1.50	11.2
4	1	1.79	15.3
	2	1.52	12.4
	3	1.56	13.5
5	1	1.88	15.4
	2	1.42	11.9
	3	1.57	14.8
6	1	1.43	13.7
	2	1.56	17.4
	3	1.55	10.8
7	1	1.77	16.9
	2	1.65	12.3
	3	1.23	11.1
8	1	1.35	14.5
	2	1.46	16.2
	3	1.86	17.3
9	1	1.57	13.6
	2	1.46	13.4
	3	1.86	12.7
10	1	1.69	14.3
	2	1.92	15.1
	3	1.77	18.8

ter adjustment to accurately simulate the real water pipe network, it can be used for various applications, such as the development of a leak detection and location algorithm. A leak detection and location algorithm based on state estimation in water pipe networks will be developed in a future work.

Chapter 4

Cooperative State Estimation of Large-scale Process Networks

In this chapter, we propose a cooperative state estimation based on Kalman filter for large-scale process network systems. The basic concepts and ideas are based on cooperative $\mathscr{H}_\infty$ estimation proposed in Wu *et al.* [12]. We develop Kalman filter (also called $\mathscr{H}_2$ estimation) based cooperative state estimation instead of $\mathscr{H}_\infty$ type estimation and provide theorems and proofs of local detectability and scalability which are important properties of the proposed cooperative Kalman filter.

4.1 Introduction

There are numerous methods to detect the leak in water pipe networks including transient based techniques using negative pressure wave (NPW), inverse-transient analysis (ITA), frequency domain techniques. In recent years, model based estimation techniques have been studied intensively for its systematic algorithm. Ye *et al.* [80] proposes detection method using Kalman filter and Emara-Shabaik *et al.* [81] propose a nonlinear multiple model state estimation scheme using a modified extended Kalman filter. The estimation technique is effective, however, it requires a dynamic process model of the system and the estimation performance depends on the model. In the previous work, we propose a dynamic model of leak in water pipe network using consensus algorithm and water hammer equations. Using the previously developed model, we develop a leak detection and location algorithm in water pipe networks. The detection algorithm is based on cooperative $\mathscr{H}_\infty$-estimation for large-scale interconnected linear systems proposed in Wu *et al* [12].

In decentralized estimation, a set of estimators are employed to create estimates of local subsystem states with only limited assistance from each other. An important requirement of this approach is that the local subsystems are detectable from local measurements. On the other hand, in a distributes estimation setup, multiple estimators create an estimate of the system's state, while cooperating with each other. The progress in the area of distributed estimation put forward issues of scalability of estimator networks, i.e., there is an interest in distributed estimation methods where the dimension of the local estimators does not increase with the total size of the system. For instance, this is relevant for multi-agent systems, where the agents are not able to perform a self-measurement, but only receive relative information. Direct applications of the existing distributed estimation algorithms result in the estimators reproducing the entire state of the complete network, and therefore, the order of the estimators grows with the size of the network.

Cooperative $\mathscr{H}_\infty$-estimation combines the benefits of both decentralized and distributed estimation and local estimators only reproduce a desired subset of state variables and their complexity does not grow with the total size of the system, in contrast to the existing methods for distributed estimation mentioned above. Moreover, cooperation between the local estimators will be used to deal with possible lack of local detectability. Therefore, the proposed method is referred to as *Cooperative estimation*. In particular, it is an $\mathscr{H}_\infty$-based design which in addition provides guaranteed performance with respect to model and measurement disturbances. Here, the methodology is applied to general large-scale linear systems where subsystems may be physically interconnected which is also applied to flow dynamics model of water pipe networks.

This chapter is organized as follows. System model under consideration and repartition algorithm are provided in Section 4.2 and the methodology of cooperative estimation design based on Kalman filter is presented in Section 4.3. Finally, two applications are presented in Section 4.4 followed by conclusions in Section 4.5.

4.2 System Model and Repartition

In this section, the system model is introduced in the form of linear interconnected systems and a repartition algorithm that plays a key role in developing cooperative state estimation is also provided.

4.2.1 System model

In chemical engineering systems, most processes consist of continuous processes with discrete measurements. Therefore, for this work, we choose the discrete stochastic system model in nonlinear form

$$x_{k+1} = F(x_k, u_k) + w_k \tag{4.1}$$

$$y_k = h(x_k) + v_k \tag{4.2}$$

The classical theory relies on several assumptions that guarantee convergence of the Kalman filter. Consider the following discrete-time linear dynamical system:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{4.3}$$

$$y_k = Cx_k + v_k \tag{4.4}$$

where $x_k \in \mathbb{R}^n$ is the state vector, $y_k \in \mathbb{R}^m$ the output vector, and $w_k \in \mathbb{R}^p$ are $v_k \in \mathbb{R}^m$ Gaussian random vectors with zero mean and covariance matrices $Q \geq 0$ and $R > 0$, respectively. Assume that the initial state, x_0, is also a Gaussian vector of zero mean and covariance Σ_0. Under the hypothesis of stabilizability of the pair (A, B) and detectability of the pair (A, C), the estimation error covariance of the Kalman filter converges to a unique value from any initial condition [8].

This work considers the design of observers for a class of linear dynamic systems in which system uncertainty can be modeled as an additive unknown disturbance term in the dynamic equation. We consider a large-scale linear time-invariant system, which consists of N interconnected subsystems that are each described by the differential equations

$$\dot{x}_{k+1} = A_k x_k + \sum_{j=1}^{N} A_{kj} x_j + B_k u + w_k \tag{4.5}$$

$$y_k = C_k x_k + \sum_{j=1}^{N} C_{kj} x_j + v_k \tag{4.6}$$

for $k = 1, \cdots, N$, where $x_k \in \mathbb{R}^{n_k}$ is the state variable, $y_k \in \mathbb{R}^{r_k}$ is the output, and w, v are noise inputs of subsystem k. The scalar components of x_k will be denoted $x_{k,i}$. Note that in this system, all subsystems are affected by the common disturbance w. This assumption does not lead to loss of generality since it also captures the case where the subsystems are affected by different disturbances w_k by simply taking w_k into one vector.

The global interconnected system can be written as

$$\dot{x} = Ax + Bw, \quad y = Cx + v \tag{4.7}$$

with

$$A = \begin{bmatrix} A_1 & A_{12} & \cdots & A_{1N} \\ A_{21} & A_2 & & \vdots \\ \vdots & & \ddots & \vdots \\ A_{N1} & \cdots & \cdots & A_N \end{bmatrix} \tag{4.8}$$

$$B^\top = \begin{bmatrix} B_1^\top & \cdots & B_N^\top \end{bmatrix} \tag{4.9}$$

$$C = \begin{bmatrix} C_1 & C_{12} & \cdots & C_{1N} \\ C_{21} & C_2 & & \vdots \\ \vdots & & \ddots & \vdots \\ C_{N1} & \cdots & \cdots & C_N \end{bmatrix} \tag{4.10}$$

by using the stacked state and disturbance vector $x = \left[x_1^\top, \cdots, x_N^\top\right]^\top \in$

$\mathbb{R}^n$ and $v = \left[v_1^\top, \cdots, v_N^\top\right]^\top$.

Assumption 1. *The global plant* (A, C) *is observable.*

Assumption 2. *The global plant* (A, C) *is detectable.*

Assumption 1 is a sufficient condition in the centralized case. Assumptions 1 and 2 are to setup a basic framework under which the state estimation problem under consideration is meaningful.

4.2.2 Repartition of system model

We will re-partition the vector x for designing local estimators. Associated with the collection of outputs (4) for every $k = 1, \cdots, N$, we choose a σ_k-dimensional partial state vector

$$x^{(k)} = \begin{bmatrix} x_{\xi_k(1)} \\ \vdots \\ x_{\xi_k(\sigma_k)} \end{bmatrix} \tag{4.11}$$

where $\xi_k(\cdot)$ is a selection function that determines which scalar components $x_{j,i}$ are included in $x^{(k)}$. It is required that all $x_{j,i}$ which contribute towards y_k are included in $x^{(k)}$. This represents a degree of freedom in the design of the estimators and all elements of the global state vector x may be chosen that are relevant to subsystem k. For instance, $x^{(k)}$ may contain x_k, but it does not have to include all of them, if for subsystem k, some parts of its own state are not important. In particular, it is required that all $x_{j,i}$ which contribute towards y_k are included in $x^{(k)}$. As a result, every output y_k can be equivalently expressed as

$$y_k = C^{(k)} x^{(k)} + v_k \tag{4.12}$$

One possible choice of $x^{(k)}$ is the stacked vector including x_k and all x_j with $C_{kj} \neq 0$. In that case,

$$\dot{x} = Ax + Bw, \quad y = Cx + v \tag{4.13}$$

with

$$A^{(k)} = \begin{bmatrix} A_k & A_{kj_1} & \cdots \\ A_{j_1k} & A_{j_1} & \\ \vdots & & \ddots \end{bmatrix} \tag{4.14}$$

and the rest of the coefficients are defined in a similar fashion.

The selection function ξ_k is a discrete injective map

$$\xi_i : \{1, \cdots, \sigma_k\} \to \mathscr{Y}, \quad \sigma_k \leq n \tag{4.15}$$

where the set $\mathscr{Y} \triangleq \{(k,i) | k = 1, \cdots, N; i = 1, \cdots, n_k\}$ is defined as the combination of all appearing indexes of the subsystem states and their scalar components $x_{k,i}$. For the ease of notation, we refer to the elements of the set $\mathscr{Y}$ as λ, i.e., $\lambda = (k,i) \in \mathscr{Y}$.

The image of ξ_k is denoted as $I^{(k)}$, $I^{(k)} \subset \mathscr{Y}$, and the inverse map ξ_k^{-1} is an enumeration of the elements of $I^{(k)}$,

$$\xi_k^{-1} : I^{(k)} \to \{1, \cdots, \sigma_k\} \tag{4.16}$$

which assigns a position in $x^{(k)}$ to selected components x_λ of the global state vector x. In general, partial state vectors $x^{(k)}$ may overlap, e.g., $x^{(1)}$ and $x^{(2)}$ may contain a common component x_λ.

For all $k = 1, \cdots, N$, the global interconnected system can now be written as

$$\begin{bmatrix} \dot{x}^{(k)} \\ \dot{x}_c^{(k)} \end{bmatrix} = \begin{bmatrix} A^{(k)} & \tilde{A}^{(k)} \\ \tilde{A}_c^{(k)} & A_c^{(k)} \end{bmatrix} \begin{bmatrix} x^{(k)} \\ x_c^{(k)} \end{bmatrix} + \begin{bmatrix} B^{(k)} \\ B_c^{(k)} \end{bmatrix} \begin{bmatrix} u^{(k)} \\ u_c^{(k)} \end{bmatrix} + \begin{bmatrix} w^{(k)} \\ w_c^{(k)} \end{bmatrix} \tag{4.17}$$

$$y^{(k)} = C^{(k)} x^{(k)} + v_k \tag{4.18}$$

by permutation of the states. For every k, the composition of the matrices $A^{(k)}$, $B^{(k)}$, etc., is determined by the composition of the partial state variable $x^{(k)}$, in turn, the latter is determined by the components of the global state x which estimator k seeks to obtain.

4.3 Cooperative State Estimation Based on Kalman Filter

In this section, we propose a cooperative Kalman filter of large-scale system based on the repartitioned model.

4.3.1 Standard Kalman filter

This section reviews unconstrained state estimation via the Kalman filter, along with some important properties of the Kalman filter that is used later in this chapter. The Kalman filter was independently invented in the 1950's by several different researchers and is named after Rudolph Kalman [29]. The problem is to find an estimate $\hat{x}_{k+1}$ of x_{x+1} given the measurements $\{y_0, y_1, \cdots, y_k\}$. We use the symbol Y_k to denote the column vector that contains the measurements $\{y_0, y_1, \cdots, y_k\}$. We assume that the following conditions are satisfied

$$
\begin{aligned}
E\left[x_0\right] &= \bar{x}_0 \\
E\left[w_k\right] &= 0 \\
E\left[e_k\right] &= 0 \\
E\left[\left(x_0-\bar{x}_0\right)\left(x_0-\bar{x}_0\right)^{\intercal}\right] &= \Sigma_0 \\
E\left[w_k w_m^{\intercal}\right] &= Q\delta_{km} \\
E\left[e_k e_m^{\intercal}\right] &= R\delta_{km} \\
E\left[w_k e_m^{\intercal}\right] &= 0 \\
E\left[x_k e_m^{\intercal}\right] &= 0 \\
E\left[x_k w_m^{\intercal}\right] &= 0
\end{aligned}
\tag{4.19}
$$

where $E[\cdot]$ is the expectation operator, $\bar{x}$ is the expected value of x, δ_{km} is the Kronecker delta function ($\delta_{km} = 1$ if $k = m$, 0 otherwise). Q and R are positive semi-definite covariance matrices. The Kalman filter equations are given by

$$P_k^- = FP_{k-1}^+F^{\intercal} + Q \tag{4.20}$$

$$K_k = P_k^- H^{\intercal}\left(HP_k^- H^{\intercal} + R\right)^{-1} \tag{4.21}$$

$$\hat{x}_k^- = F\hat{x}_{k-1}^+ \tag{4.22}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k\left(y_k - H\hat{x}_k{}^-\right) \tag{4.23}$$

$$P_k^+ = \left(I - K_k H\right)P_k^- \tag{4.24}$$

for $k = 1, 2, \cdots$, where I is the identity matrix. $\hat{x}_k^-$ is the *a priori* estimate of the state x_k given measurements up to and including time $k-1$. $\hat{x}_k^+$ is the *a posteriori* estimate of the state x_k given measurements up to and including time k. K_k is the Kalman gain, P_k^- is the covariance of the *a priori* estimation error $x_k - \hat{x}_k^-$, and P_k^+ is the covariance of the *a posteriori* estimation error $x_k - \hat{x}_k^+$. The Kalman filter is initialised with

$$\hat{x}_0^+ = E\left[x_0\right] \tag{4.25}$$

$$P_0^+ = E\left[\left(x_0 - \hat{x}_0^+\right)\left(x_0 - \hat{x}_0^+\right)^{\mathsf{T}}\right] \tag{4.26}$$

where $E[\cdot]$ is the expectation operator.

When the noise sequences $\{w_k\}$ and $\{v_k\}$ are Gaussian, uncorrelated, and white, the Kalman filter is the minimumvariance filter and minimises the trace of the estimation error covariance at each time step. When $\{w_k\}$ and $\{v_k\}$ are non-Gaussian, the Kalman filter is the minimum-variance linear filter, although there might be nonlinear filters that perform better [30].

Note that this is the prediction form of the Kalman filter equations, so x_k is estimated on the basis of measurements up to and including time $k-1$. The filter is initialized with $\hat{x}_0 = \bar{x}_0$, and Σ_0 given above. It can be shown [12] that the Kalman filter has several attractive properties. For instance, if x_0, $\{w_k\}$, and $\{e_k\}$ are jointly Gaussian, the Kalman filter estimate $\hat{x}_{k+1}$ is the conditional mean of x_{k+1} given the measurements Y_k; i.e., $\hat{x}_{k+1} = E[x_{k+1}|Y_k]$. Even if x_0, $\{w_k\}$, and $\{e_k\}$ are not jointly Gaussian, the Kalman filter estimate is the best affine estimator given the measurements Y_k; i.e., of all estimates of x_{k+1} that are of the form $FY_k + g$ (where F is a fixed matrix and g

is a fixed vector), the Kalman filter estimate is the one that minimizes the variance of the estimation error. It can be shown that the Kalman filter estimate (i.e., the minimum variance estimate) can be given by

$$\hat{x}_{k+1} = \bar{\bar{x}}_{k+1} \equiv \bar{x}_{k+1} + \Sigma_{xy}\Sigma_{yy}^{-1}\left(Y_k - \bar{Y}_k\right) \tag{4.27}$$

where $\bar{x}_{k+1}$ is the expected value of x_{k+1}, $\bar{Y}_k$ is the expected value of Y_k, Σ_{xy} is the variance matrix of x_{k+1} and Y_k, Σ_{yy} is the covariance matrix of Y_k, and $\bar{\bar{x}}_{k+1}$ is the conditional mean of x_{k+1} given the measurements Y_k. In addition, we know that the Kalman filter estimate $\bar{x}_{k+1}$ and Y_k are jointly Gaussian, in which case $\bar{x}_{k+1}$ is conditionally Gaussian given Y_k. The conditional probability density function of x_{k+1} given Y_k

$$P(x|Y) = \frac{\exp\left[-(x-\bar{\bar{x}})^{\intercal}\Sigma^{-1}(x-\bar{\bar{x}})/2\right]}{(2\pi)^{n/2}|\Sigma|^{1/2}} \tag{4.28}$$

where n is the dimension of x and

$$\Sigma = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} \tag{4.29}$$

where Σ_{xx} is the covariance matrix of x_k. The Kalman filter estimate is that value of x that maximizes the conditional probability density function $P(x|Y)$, and Σ is the covariance of the Kalman filter estimate.

4.3.2 Cooperative Kalman filter

The problem considered here is to design a local estimator for every subsystem k that creates an estimate for the local partial state variable $x^{(k)}$ using the local measurements y_k described in Eq. 4.17

and 4.18. The vector of local estimates will be denoted

$$\hat{x}^{(k)} = \begin{bmatrix} \hat{x}^{k}_{\xi_k(1)} \\ \vdots \\ \hat{x}^{k}_{\xi_k(\sigma_k)} \end{bmatrix} \in \mathbb{R}^{\sigma_k} \tag{4.30}$$

where $\hat{x}^{(k)}_{\lambda}$ is the estimate for x_λ computed at subsystem k.

The local estimation error vector is defined as

$$\varepsilon^{(k)} = x^{(k)} - \hat{x}^{(k)} = \begin{bmatrix} x^{k}_{\xi_k(1)} - \hat{x}^{k}_{\xi_k(1)} \\ \vdots \\ x^{k}_{\xi_k(\sigma_k)} - \hat{x}^{k}_{\xi_k(\sigma_k)} \end{bmatrix} \in \mathbb{R}^{\sigma_k} \tag{4.31}$$

The cooperative estimator determines a collection of estimates $\hat{x}^{(k)}(t),\ k = 1, \cdots, N$, such that the following two properties are satisfied simultaneously.

1. In the absence of model and measurement disturbances (i.e., when $\nu = 0$, $\eta = 0$), the estimation errors decay so that $\varepsilon^{(k)} \to 0$ exponentially for all $k = 1, \cdots, N$.

2. The estimators satisfy Kalman filter performance.

$$\min_{L^{(k)},K^{(k)}} P^{(k)} = \min_{L^{(k)},K^{(k)}} E[\varepsilon^{(k)}\varepsilon^{(k)T}] \tag{4.32}$$

This will be achieved by allowing certain agents to communicate with each other.

4.3.2.1 Communication requirements

There are two factors, which influence the required communication for the cooperative estimation setup: The first one is detectability of $(A^{(k)}, C^{(k)})$. In the special case of $\tilde{A}^{(k)} = 0$, and $(A^{(k)}, C^{(k)})$ being detectable for all $k = 1, \cdots, N$, no communication is necessary at all, as for every subsystem, an estimator can be designed separately. However, these assumptions may not hold in a general case. In particular, in this paper, we do not require that $(A^{(k)}, C^{(k)})$ are detectable for all $k = 1, \cdots, N$, which is a major difference compared to existing methods in literature, In fact, even all $(A^{(k)}, C^{(k)})$ may be undetectable.

The second factor which influences the required communication is sparsity of $\tilde{A}^{(k)}$. Ideally, when the partial state $x^{(k)}$ is decoupled from the rest of the system, i.e., $\tilde{A}^{(k)} = 0$, a standard $\mathscr{H}_\infty$ filter can be employed to carry out the estimation of $x^{(k)}$ from y_k. However, if $x^{(k)}$ includes a state x_λ, which is connected to a state $x_{\lambda*}$ that is not a component of $x^{(k)}$, then the problem becomes more challenging.

In order to define the required communication channels, we use an assignment function

$$\zeta : \mathscr{Y} \to \{0, 1, \cdots, N\} \tag{4.33}$$

with the property that $\lambda \in I^{(\zeta(\lambda))}$ if $\zeta(\lambda) \neq 0$. Moreover, $\zeta(\lambda) \neq 0$ only if $\lambda \notin I^{(k)}$ for all $k = 1, \cdots, N$. The map $\zeta(\cdot)$ assigns responsibilities in estimating the system's states to the subsystems and their local estimators. In general, $\zeta(\lambda)$ is not unique and there is a degree of freedom in selection of the assignment function. However in the case when $x^{(k)}$'s do not overlap, the assignment function $\zeta(\lambda)$ is unique.

With the definition of the assignment function ζ, we can introduce the assumption on the communication graph used in this work.

Assumption 3. *If a component x_λ of $x^{(k)}$ is physically coupled to a state $x_{\lambda*}$, where $\lambda^* \in \mathscr{Y} \setminus I^{(k)}$, then subsystem $j = \zeta(\lambda^*) \neq 0$ can communicate to subsystem k, i.e., $(j,k) \in \mathscr{E}$.*

We denote with $I_c^{(k)}$ the set of all indexes $\lambda^* \in \mathscr{Y} \setminus I^{(k)}$ with the property that for all $x_{\lambda^*} \in I_c^{(k)}$, there exists a component x_λ of $x^{(k)}$, which is coupled to x_{λ^*}. Assumption 2 reflects the point made above, as the more entries $\tilde{A}^{(k)}$ has, the more communication between the subsystems is required. Some remarks on the realization of this assumption are in order:

Lemma 4.1. *For all $\lambda \in \mathscr{Y}$ there exists a $k \in \{1, \cdots, N\}$, such that $\lambda \in I^{(k)}$.*

Proof. Suppose there exists a $\lambda \in \mathscr{Y}$, such that $\lambda \notin I^{(k)}$ for all $k = 1, \cdots, N$. By the definition of the partial states $x^{(k)}$ and the selection function ξ_k, the column of C which corresponds to x_λ is 0. Moreover, by the definition of ζ, we have $\zeta(\lambda) = 0$ and thus, it follows from Assumption 2 that there is no partial state $x^{(k)}$ that is coupled to x_λ. Therefore, x_λ is not observable, which contradicts Assumption 1. □

As noted for the repartition, the vectors $x^{(k)}$ may overlap. Therefore, including a consensus term whenever overlapping estimators can communicate is able to enhance estimation performance of the subsystems and in some cases even facilitates feasibility of the design conditions.

4.3.2.2 Estimator dynamics

The estimator dynamics are now proposed for each subsystem as

$$\begin{aligned}\dot{\hat{x}}^{(k)} &= A^{(k)}\hat{x}^{(k)} + B^{(k)}u^{(k)} + L^{(k)}\left(y_k - C^{(k)}\hat{x}^{(k)}\right) \\ &\quad + \sum_{\lambda\in I_c^{(k)}} \left[\tilde{A}^{(k)}\right]_\lambda \hat{x}_\lambda^{(\zeta(\lambda))} + \sum_{\lambda\in I_c^{(k)}} \left[\tilde{B}^{(k)}\right]_\lambda u_\lambda^{(\zeta(\lambda))} \\ &\quad + K^{(k)} \sum_{\lambda\in\mathscr{N}_k} \left(\sum_{\lambda\in I_c^{(k)}\cap I^{(j)}} e_{\xi^{-1}(\lambda)} \left(\hat{x}_\lambda^{(j)} - \hat{x}_\lambda^{(k)}\right)\right)\end{aligned} \tag{4.34}$$

where initial condition $x_0 = 0$, $\left[\tilde{A}^{(k)}\right]_\lambda$ is the column of $\tilde{A}^{(k)}$ which corresponds to x_λ and the unit vector $e_{\xi^{-1}(\lambda)}$ injects the difference $\hat{x}_\lambda^{(j)} - \hat{x}_\lambda^{(k)}$ to the σ_k-dimensional space.

The problem is to determine estimator gains $L^{(k)}$, $K^{(k)}$ such that properties 1 and 2 hold in the previous section.

In order to solve this problem, we define the extended graph $\tilde{\mathscr{G}}$, which will be used in the analysis of the interconnection structure between the subsystems. Let every subsystem be represented by a cluster of σ_k nodes, where vertex v_{k_λ} represents the estimator state $\hat{x}_\lambda^{(k)}$. The edges of $\tilde{\mathscr{G}}$ are now determined by Algorithm 1 in [12].

The graph generated by Algorithm 1 graphically displays the detailed connection structure of the estimation vectors $\hat{x}^{(k)}$. The outdegree of vertex v_{k_λ} in the extended graph is denoted by $q(k,\lambda)$. This definition will be used to present our main results on the design of the filter gains, which are given in the next section.

The estimator error dynamics at node k is

$$\begin{aligned}\dot{\varepsilon}^{(k)} &= \left(A^{(k)} - L^{(k)}C^{(k)}\right)\varepsilon^{(k)} + \sum_{\lambda\in I_c^{(k)}} \left[\tilde{A}^{(k)}\right]_\lambda \varepsilon_\lambda^{(\zeta(\lambda))} \\ &\quad + K^{(k)} \sum_{j\in\mathcal{N}_k} \left(\sum_{\lambda\in I^{(k)}\cap I^{(j)}} e_{\xi^{-1}(\lambda)} \left(\varepsilon_\lambda^{(j)} - \varepsilon_\lambda^{(k)}\right)\right) - L^{(k)}\eta^{(k)} + B^{(k)}v\end{aligned} \tag{4.35}$$

4.3.2.3 Filter gains

We define the matrices

$$N^{(k)} = \sum_{j\in\mathcal{N}_k} \left(\sum_{\lambda\in I^{(k)}\cap I^{(j)}} e_{\xi^{-1}(\lambda)} e_{\xi^{-1}(\lambda)}^\top \right) \tag{4.36}$$

$$\begin{aligned}Q^{(k)} &= A^{(k)} + A^{(k)\top} - L^{(k)}C^{(k)} - \left(L^{(k)}C^{(k)}\right)^\top - K^{(k)}N^{(k)} - \left(K^{(k)}N^{(k)}\right)^\top \\ &\quad + \alpha + \underbrace{\pi_k \begin{bmatrix} q(k,\xi_k(1)) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & q(k,\xi_k(\sigma_k)) \end{bmatrix}}_{\Pi_k}\end{aligned} \tag{4.37}$$

where $G^{(k)} \in \mathbb{R}^{\sigma_k\times r_k}$ and $F^{(k)} \in \mathbb{R}^{\sigma_k\times\sigma_k}$ are unknown matrices, $P^{(k)} \in \mathbb{R}^{\sigma_k\times\sigma_k}$ is a symmetric positive definite matrix and $P_i^{(k)} \in \mathbb{R}$ is the i-th diagonal element of $P^{(k)}$. π_k and α are positive constants which will later play the role of design parameters. Furthermore, we define $p(\lambda)$ as the diagonal element of $P^{(\zeta(\lambda))}$ which corresponds to x_λ, i.e., the $\xi_{\zeta(\lambda)}^{-1}(\lambda)$'th diagonal element.

Next, for all $k = 1, \cdots, N$, we define the matrices

$$S^{(k)} = \left[\begin{array}{ccc} \left[\tilde{A}^{(k)}\right]_{\lambda_1^{(k)}} & \left[\tilde{A}^{(k)}\right]_{\lambda_2^{(k)}} & \cdots \end{array} \right] \tag{4.38}$$

$$R^{(k)} = \begin{bmatrix} \pi_{\zeta\left(\lambda_1^{(k)}\right)} & 0 & 0 \\ 0 & \pi_{\zeta\left(\lambda_1^{(k)}\right)} & 0 \\ 0 & 0 & \ddots \end{bmatrix} \tag{4.39}$$

for $\{\lambda_1^{(k)}, \lambda_2^{(k)}, \cdots\} = I_c^{(k)}$ and

$$T_j^{(k)} = \left[\begin{array}{ccc} K^{(k)} e_{\xi_k^{-1}(\lambda_{1,j})} & K^{(k)} e_{\xi_k^{-1}(\lambda_{2,j})} & \cdots \end{array} \right] \tag{4.40}$$

$$U_j^{(k)} = \begin{bmatrix} \pi_j & 0 & 0 \\ 0 & \pi_j & 0 \\ 0 & 0 & \ddots \end{bmatrix} \tag{4.41}$$

where $\{\lambda_1^{kj}, \lambda_2^{kj}, \cdots\} = I^{(k)} \cap I^{(j)}$

With these definitions, we are ready to present following theorem.

Theorem 4.1. *Consider a group of interconnected LTI systems with local outputs. The problem admits a solution in the form of estimators Eq. 4.34 with $L^{(k)}$ and $K^{(k)}$ if the matrices $L^{(k)}$ and $K^{(k)}$ for*

$k = 1, \cdots, N$ *are a solution of the following LMIs:*

$$\begin{bmatrix} Q^{(k)} & -L^{(k)} & B^{(k)} & S^{(k)} & T^{(k)}_{j_1,k} & \cdots & T^{(k)}_{j_{\tau_k},k} \\ -L^{(k)\top} & -I & 0 & 0 & 0 & 0 & 0 \\ B^{(k)\top} & 0 & -I & 0 & 0 & 0 & 0 \\ S^{(k)\top} & 0 & 0 & R^{(k)} & 0 & 0 & 0 \\ T^{(k)\top}_{j_1,k} & 0 & 0 & 0 & U^{(k)}_{j_1,k} & 0 & 0 \\ \vdots & 0 & 0 & 0 & 0 & \ddots & 0 \\ T^{(k)\top}_{j_1,k} & 0 & 0 & 0 & 0 & 0 & U^{(k)}_{j_1,k} \end{bmatrix} < 0 \qquad (4.42)$$

with $\{j_{1,k}, j_{2,k}, \cdots, j_{\tau_k,k}\} = \mathscr{N}_k$

Proof. Using the estimator error dynamics at node k, we use a Lyapunov function

$$\begin{aligned} J^{(k)} &= \mathbf{E}\left[\varepsilon^{(k)\top}\varepsilon^{(k)}\right] \\ &= \mathbf{E}\left[\mathrm{Tr}(\varepsilon^{(k)}\varepsilon^{(k)\top})\right] \\ &= \mathrm{Tr}\,P^{(k)} \end{aligned} \qquad (4.43)$$

$$V(\varepsilon) = \sum_{k=1}^{N} \varepsilon^{(k)\top} P^{(k)} \varepsilon^{(k)} \qquad (4.44)$$

$$P^{(k)} = \mathbf{E}\left[\varepsilon^{(k)}\varepsilon^{(k)\top}\right] \qquad (4.45)$$

$$\mathrm{Tr}\,P^{(k)} = \mathbf{E}\left[\varepsilon^{(k)\top}\varepsilon^{(k)}\right] \qquad (4.46)$$

where $V^{(k)}(\varepsilon^{(k)}) = \varepsilon^{(k)\intercal}P^{(k)}\varepsilon^{(k)}$ are the individual components of $V(\varepsilon)$. The Lie derivative of $V^{(k)}(\varepsilon^{(k)})$ is

$$
\begin{aligned}
j^{(k)} &= \mathbf{E}\Bigg[2\varepsilon^{(k)\top}\left(A^{(k)} - L^{(k)}C^{(k)}\right)\varepsilon^{(k)} \\
&\quad +2\varepsilon^{(k)\top}\left(-L^{(k)}\eta^{(k)} + B^{(k)}v^{(k)}\right) \\
&\quad +2\varepsilon^{(k)\top}\sum_{\lambda\in I_c^{(k)}}\left[\tilde{A}^{(k)}\right]_\lambda \varepsilon_\lambda^{(\zeta(\lambda))} \\
&\quad +2\varepsilon^{(k)\top}K^{(k)}\sum_{j\in\mathcal{N}_k}\sum_{\lambda\in I^{(k)}\cap I^{(j)}} e_{\xi^{-1}(\lambda)}\left(\varepsilon_\lambda^{(j)} - \varepsilon_\lambda^{(k)}\right)\Bigg] \\
&= \mathbf{E}\Bigg[2\varepsilon^{(k)\top}\left(A^{(k)} - L^{(k)}C^{(k)} - K^{(k)}N^{(k)}\right)\varepsilon^{(k)} \\
&\quad +2\varepsilon^{(k)\top}\left(-L^{(k)}\eta^{(k)} + B^{(k)}v^{(k)}\right) \\
&\quad +2\varepsilon^{(k)\top}\sum_{\lambda\in I_c^{(k)}}\left[\tilde{A}^{(k)}\right]_\lambda \varepsilon_\lambda^{(\zeta(\lambda))} \\
&\quad +2\varepsilon^{(k)\top}K^{(k)}\sum_{j\in\mathcal{N}_k}\sum_{\lambda\in I^{(k)}\cap I^{(j)}} e_{\xi^{-1}(\lambda)}\varepsilon_\lambda^{(j)}\Bigg]
\end{aligned}
\tag{4.47}
$$

With the filter gains and the LMIs it can be obtained that

$$
\begin{aligned}
j^{(k)} &= \mathbf{E}\Bigg[\varepsilon^{(k)\top}\left(Q^{(k)} - \alpha - \Pi_k\right)\varepsilon^{(k)} + 2\varepsilon^{(k)\top}L^{(k)}\eta^{(k)} + 2\varepsilon^{(k)\top}B^{(k)}v^{(k)} \\
&\quad +2\varepsilon^{(k)\top}\sum_{\lambda\in I_c^{(k)}}\left[\tilde{A}^{(k)}\right]_\lambda \varepsilon_\lambda^{(\zeta(\lambda))} + 2\varepsilon^{(k)\top}K^{(k)}\sum_{j\in\mathcal{N}_k}\sum_{\lambda\in I^{(k)}\cap I^{(j)}} e_{\xi^{-1}(\lambda)}\varepsilon_\lambda^{(j)}\Bigg] \\
&\leq \mathbf{E}\Bigg[\sum_{\lambda\in I_c^{(k)}}\varepsilon_\lambda^{(\zeta(\lambda))\top}\pi_{\zeta(\lambda)}\varepsilon_\lambda^{(\zeta(\lambda))} + \sum_{j\in\mathcal{N}_k}\sum_{\lambda\in I^{(k)}\cap I^{(j)}}\varepsilon_\lambda^{(j)\top}\pi_j\varepsilon_\lambda^{(j)} \\
&\quad +w^{(k)\top}w^{(k)} + v^{(k)\top}v^{(k)} - \alpha\varepsilon^{(k)\top}\varepsilon^{(k)} - \varepsilon^{(k)\top}\Pi_k\varepsilon^{(k)}\Bigg]
\end{aligned}
\tag{4.48}
$$

Summing up the $V^{(k)}$s, it holds for V that

$$\begin{aligned}\dot{J} &\leq \mathbf{E}\Bigg[\sum_{k=1}^{N} \pi_k \sum_{\lambda \in I^{(k)}} q(k,\lambda) \varepsilon_{\lambda}^{(k)\top} \varepsilon_{\lambda}^{(k)} + \sum_{k=1}^{N} w^{(k)\top} w^{(k)} + \sum_{k=1}^{N} v^{(k)\top} v^{(k)} \\ &\quad - \sum_{k=1}^{N} \alpha \varepsilon^{(k)\top} \varepsilon^{(k)} - \sum_{k=1}^{N} \varepsilon^{(k)\top} \Pi_k \varepsilon^{(k)}\Bigg] \\ &\leq \mathbf{E}\left[-\alpha \sum_{k=1}^{N} \varepsilon^{(k)\top} \varepsilon^{(k)} + \sum_{k=1}^{N} w^{(k)\top} w^{(k)} + \sum_{k=1}^{N} v^{(k)\top} v^{(k)}\right]\end{aligned} \tag{4.49}$$

$$0 \leq \dot{J} \leq 0 \rightarrow \dot{J} = 0 \tag{4.50}$$

Integrating both sides of (14) on the interval $[0, T]$, we obtain

$$\begin{aligned}J(T) \quad + \quad & \sum_{k=1}^{N} \int_0^T \varepsilon^{(k)T} W^{(k)} \varepsilon^{(k)} \\ & \leq \sum_{k=1}^{N} \int_0^T \left(\omega^2 ||v^{(k)}||^2 + \gamma^2 ||w^{(k)}||^2\right) dt + \sum_{k=1}^{N} \varepsilon_0^{(k)T} P^{(k)} \varepsilon^{(k)}\end{aligned} \tag{4.51}$$

As $V(\varepsilon(T)) \geq 0$ and with the zero initial conditions of the observer states, it follows that

$$\sum_{k=1}^{N} \int_0^T \varepsilon^{(k)T} W^{(k)} \varepsilon^{(k)} \leq \sum_{k=1}^{N} \int_0^T \left(\omega^2 ||v^{(k)}||^2 + \gamma^2 ||w^{(k)}||^2\right) dt + I_0 \tag{4.52}$$

Letting $T \to \infty$, this satisfies Property (ii) of Problem 1. Furthermore, if $w_k = 0$ and $v_k = 0$ for all $k = 1, \cdots, N$, then it follows from (14) that

$$\dot{V}(\varepsilon) \leq -\alpha V \tag{4.53}$$

which implies that Property (i) of Problem 1 holds. □

Note that the choice of α determines the convergence speed of the estimators, where a larger α enforces faster convergence of the estimates. However, larger values of α typically lead to higher filter gain values.

The resulting cooperative estimators are local and their complexity does not increase with the total size of the network. The salient feature of the resulting cooperative estimators is that these estimators are local and their complexity does not increase with the total size of the network. In this sense, the method presented in this work is scalable and guarantees Kalman-type performance. In contrast, a direct application of the algorithms developed in [8], [9] and [13], [14], to the problem considered here would result in the order of the estimators growing with the size of the network. Some remarks on the solution of the LMIs are below.

Remark 1. As it can be seen form the LMIs, the solution to design problem presented here involves solving coupled LMIs. When the nature of the application allows for these LMIs to be solved offline, this can be done in a centralized manner. The resulting gain matrices $L^{(k)}$, $K^{(k)}$ can then be deployed to the filters and this will ensure that while the estimation algorithm is running, the estimators are fully distributed.

Remark 2. As noted before, the choice of the partial state vectors is not unique. As a special case, the choice $x^{(k)} = x$ for all $k = 1, \cdots, N$ yields local estimators similar to [13], [14].

Remark 3. If there is no sensor fusion, $K^{(k)} = 0$ and $L^{(k)}$ is a standard Kalman filter gain. In case of sensor fusion, $K^{(k)} \neq 0$

4.4 Application I: Water Pipe Networks for Fault Detection and Location

Cooperative Kalman filter estimation proposed in the previous section is applied to the developed model based on consensus algorithm for fault detection and location. The estimation is appropriate for large-scale interconnected system such as the developed water pipe network model. Especially, the fact that dimensions of local estimators do not grow with the network size can be more useful for larger system such as water pipe network. Furthermore, another important point of this estimator is that it does not require local detectability of subsystems and this is particularly suitable for systems with insufficient sensors.

The cooperative estimator is designed to guarantee Kalman filter-performance for a class of linear interconnected systems where system uncertainty can be modeled as an additive unknown disturbance term in the dynamic equation. Therefore, it can give asymptotically stable state estimates in application to the developed water pipe network model that includes disturbance terms.

Fig. 4.1 shows the divided subsystems in the corresponding water pipe network shown in Fig. 2.2 and which scalar components are selected by $\xi_k(\cdot)$ and included in the partial state vector $x^{(k)}$ $(k = 1,2,3)$. The overall pipe networks is divided into three subsystems and they receive measurement information from their corresponding sensors and estimate the partial states $x^{(k)}$. The junction node 22 is overlapped in all three subsystems and its state components $[x_{1,43},\, x_{1,44}]^\mathsf{T}$ are estimated simultaneously in each subsystem as shown in Fig. 4.1.

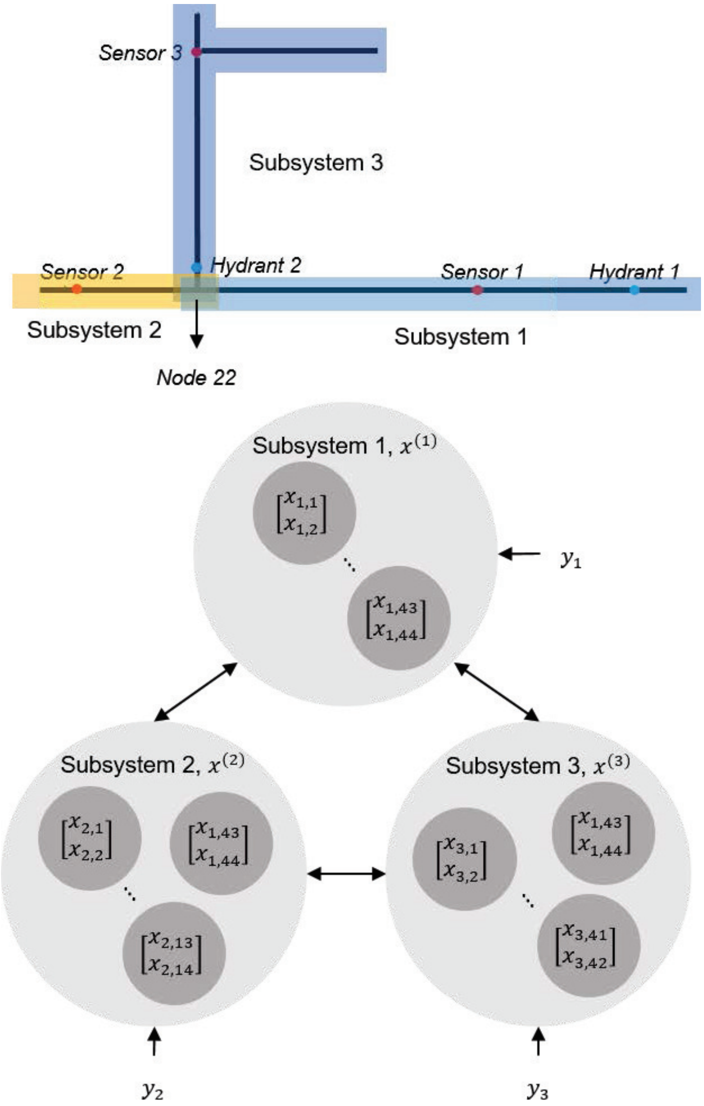

Sensor 3
Subsystem 3
Hydrant 2
Sensor 2
Sensor 1
Hydrant 1
Subsystem 2
Subsystem 1
Node 22
Subsystem 1, $x^{(1)}$
$\begin{bmatrix} x_{1,1} \\ x_{1,2} \end{bmatrix}$
$\begin{bmatrix} x_{1,43} \\ x_{1,44} \end{bmatrix}$
y_1
Subsystem 2, $x^{(2)}$
$\begin{bmatrix} x_{2,1} \\ x_{2,2} \end{bmatrix}$
$\begin{bmatrix} x_{1,43} \\ x_{1,44} \end{bmatrix}$
$\begin{bmatrix} x_{2,13} \\ x_{2,14} \end{bmatrix}$
Subsystem 3, $x^{(3)}$
$\begin{bmatrix} x_{3,1} \\ x_{3,2} \end{bmatrix}$
$\begin{bmatrix} x_{1,43} \\ x_{1,44} \end{bmatrix}$
$\begin{bmatrix} x_{3,41} \\ x_{3,42} \end{bmatrix}$
y_2
y_3

Figure 4.1: Three subsystems of water pipe network and the partial state vectors $x^{(k)}$, $k = 1, 2, 3$ for each subsystem

In this section, we propose Algorithm to detect and locate fault of water pipe networks by using the developed model and cooperative Kalman filter.

Now we can identify all the states at nodes in the network with the cooperative and distributed estimation. If faults occur in the pipe network, a negative pressure wave (NPW) is generated and propagates from the leak point to the entire network. Therefore, the NPW signals would appear in order of distance and this distance difference is also displayed on the pressure sensors as in Fig. 3.4. After all the states are estimated, the leak location is easily identified since the node having the steepest pressure drop should be nearest to the leak point. This method does not require any fault location algorithm. We set up links between nodes are all less than 10 m according to the node division strategy so that the location error range is within 10 m.

Now we propose the following algorithm.

Algorithm. Fault detection and location algorithm

```
Data: $A^{(k)},B^{(k)},C^{(k)}$ for subsystem $x^{(k)},\ k=1,\cdots,N$
Result: Fault location
Compute $L^{(k)}$ and $K^{(k)}$ offline
While Fault detection and location
    Read pressure measurements
    If Fault occurs
        Implement estimator Eq. 4.34 and find $\hat{x}^{(k)},\ k=1,\cdots,N$
        Give fault location
    else
        Give no fault sign
```

The initial conditions of the system and the estimator are set to 250 kPa. Estimation results of the water pipe network under consideration are shown in Fig. 4.5. Fig. 4.5(a) shows all the estimates of pressure at the nodes in the network can be estimated using the cooperative distributed estimation and the proposed model with only three measurement information.

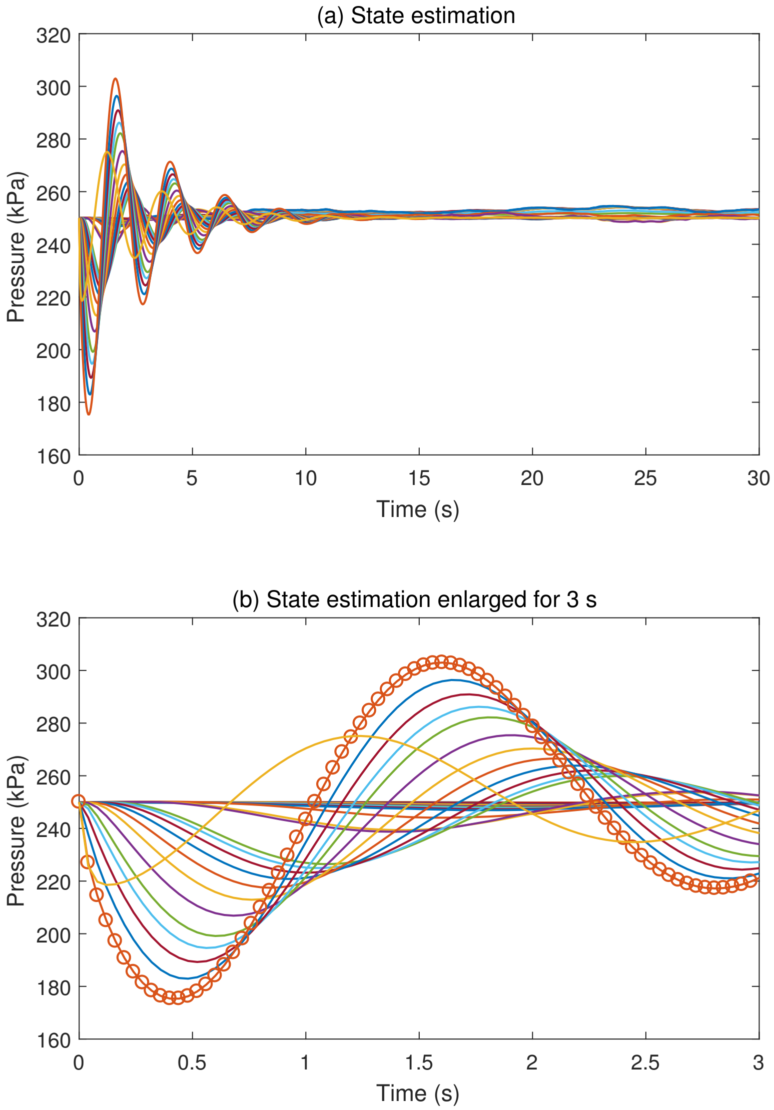
The early part of the pressure drop by the leak is enlarged in Fig. 4.5(b). The line with circle shows the steepest drop of all the estimates and thus it is the nearest to the actual leak location. Indeed, it is node 4 which is right next to the leak point and the location error is 9.55 m. As the pressure wave propagates from the leak to other part of the network, the pressure drop appears in distance order as shown in Fig. 4.5(b).

Fig. 4.3 shows the case when initial conditions of the system and the estimation are different. The initial conditions are set as $x_0 = 250$ and $\hat{x}_0 = 240$ kPa, respectively. Fig. 4.3(a)–(b) indicate the states at nodes of Sensor 1, Sensor 2, and Sensor 3, respectively. Even with the difference between initial states, the estimation errors asymptotically decrease to zero for all subsystems.

We conclude the estimation section with some remarks.

Remark 4. It is found that $(A^{(1)}, C^{(1)})$, $(A^{(2)}, C^{(2)})$, and $(A^{(3)}, C^{(3)})$ defined in Eq. 4.17 and 4.18 are undetectable. We emphasize that the states of the network can be estimated even without guaranteeing detectability of individual subsystems.

Remark 5. In water pipe networks considered here, there are three pressure measurements, subsystems, and local estimators. The complexity of these local estimators does not increase with the total size of the water pipe network unless the size of each subsystem change. In this sense, the estimation method presented here is scalable. However, when the size of the water pipe network increases, more pressure



(a) State estimation

(b) State estimation enlarged for 3 s

Figure 4.2: (a) Estimation results of all states in water pipe network and (b) enlargement of estimation results for 3 s

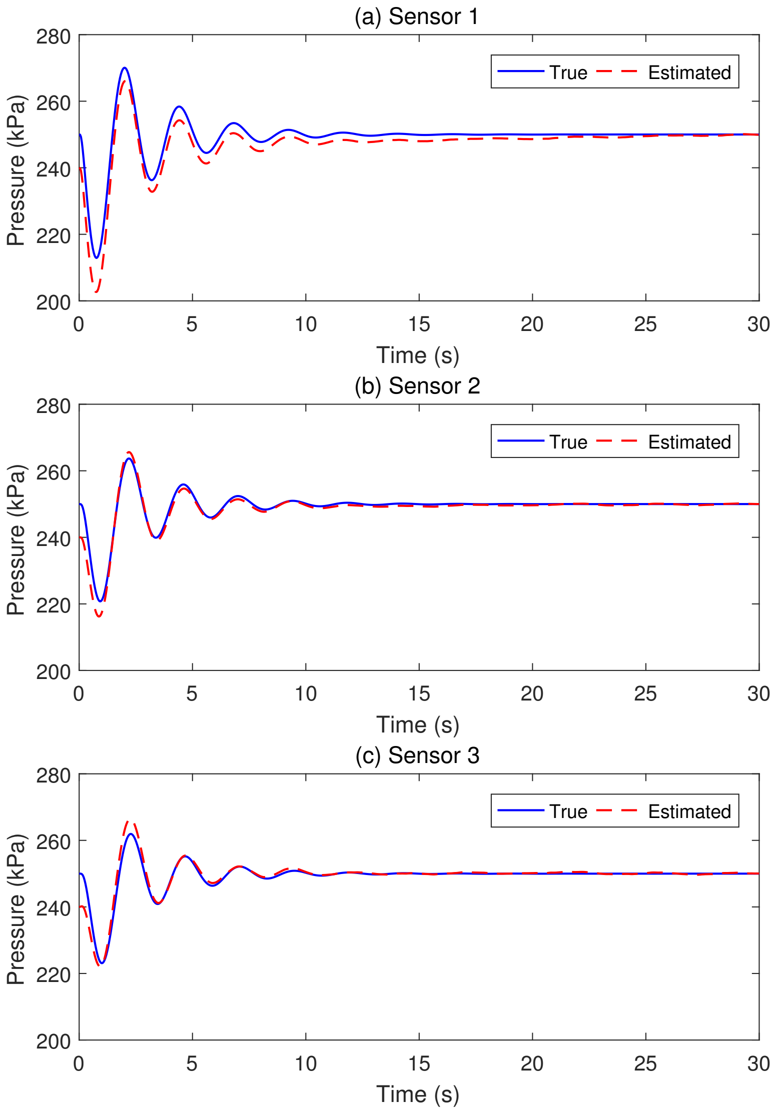


Figure 4.3: Dynamics of real and estimated states for three sensors with different initial conditions

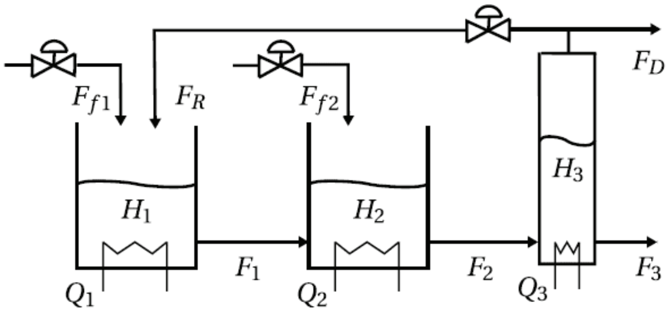


Figure 4.4: Two reactors in series with separator and recycle

sensors are needed to ensure scalability.

4.5 Application II: Chemical Process Networks with Recycles

Consider a plant consisting of two reactors and a separator. A stream of pure reactant A is added to each reactor and converted to the product B by a first-order reaction. The product is lost by a parallel first-order reaction to side product C. The distillate of the separator is split and partially redirected to the first reactor (see Fig. 4.4).

4.5.1 Network model

The model for the plant is

$$\frac{dH_1}{dt} = \frac{1}{\rho A_1}\left(F_{f1} + F_R - F_1\right) \tag{4.54}$$

$$\frac{dx_{A1}}{dt} = \frac{1}{\rho A_1 H_1}\left(F_{f1}x_{A0} + F_R x_{AR} - F_1 x_{A1}\right) - k_{A1}x_{A1} \tag{4.55}$$

$$\frac{dx_{B1}}{dt} = \frac{1}{\rho A_1 H_1} \left(F_R x_{BR} - F_1 x_{B1}\right) + k_{A1} x_{A1} - k_{B1} x_{B1} \tag{4.56}$$

$$\frac{dT_1}{dt} = \frac{1}{\rho A_1 H_1} \left(F_{f1} T_0 + F_R T_R - F_1 T_1\right) - \frac{1}{c_p} \left(k_{A1} x_{A1} \Delta H_A + k_{B1} x_{B1} \Delta H_B\right) + \frac{Q_1}{\rho A_1 c_p H_1} \tag{4.57}$$

$$\frac{dH_2}{dt} = \frac{1}{\rho A_2} \left(F_{f2} + F_1 - F_2\right) \tag{4.58}$$

$$\frac{dx_{A2}}{dt} = \frac{1}{\rho A_2 H_2} \left(F_{f2} x_{A0} + F_1 x_{A1} - F_2 x_{A2}\right) - k_{A2} x_{A2} \tag{4.59}$$

$$\frac{dx_{B2}}{dt} = \frac{1}{\rho A_2 H_2} \left(F_1 x_{B1} - F_2 x_{B2}\right) + k_{A2} x_{A2} - k_{B2} x_{B2} \tag{4.60}$$

$$\frac{dT_2}{dt} = \frac{1}{\rho A_2 H_2} \left(F_{f2} T_0 + F_1 T_1 - F_2 T_2\right) - \frac{1}{c_p} \left(k_{A2} x_{A2} \Delta H_A + k_{B2} x_{B2} \Delta H_B\right) + \frac{Q_2}{\rho A_2 c_p H_2} \tag{4.61}$$

$$\frac{dH_3}{dt} = \frac{1}{\rho A_3} \left(F_2 - F_D - F_R - F_3\right) \tag{4.62}$$

$$\frac{dx_{A3}}{dt} = \frac{1}{\rho A_3 H_3} \left(F_2 x_{A2} - (F_D + F_R) x_{AR} - F_3 x_{A3}\right) \tag{4.63}$$

$$\frac{dx_{B3}}{dt} = \frac{1}{\rho A_3 H_3} \left(F_2 x_{B2} - (F_D + F_R) x_{BR} - F_3 x_{B3}\right) \tag{4.64}$$

$$\frac{dT_3}{dt} = \frac{1}{\rho A_3 H_3}(F_2 T_2 - (F_D + F_R)T_R - F_3 T_3) + \frac{Q_3}{\rho A_3 c_p H_3} \tag{4.65}$$

in which for all $i \in \mathbb{I}_{1:3}$

$$F_i = k_{vi} H_i \tag{4.66}$$

$$k_{Ai} = k_A \exp\left(-\frac{E_A}{RT_i}\right) \tag{4.67}$$

$$k_{Bi} = k_B \exp\left(-\frac{E_B}{RT_i}\right) \tag{4.68}$$

The recycle flow and weight percent satisfy

$$F_D = 0.01 F_R \tag{4.69}$$

$$x_{AR} = \frac{\alpha_A x_{A3}}{\bar{x}_3} \tag{4.70}$$

$$x_{BR} = \frac{\alpha_B x_{B3}}{\bar{x}_3} \tag{4.71}$$

$$\bar{x}_3 = \alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3} \tag{4.72}$$

$$x_{C3} = (1 - x_{A3} - x_{B3}) \tag{4.73}$$

The output and input are denoted, respectively

$$y = [H_1,\ x_{A1},\ x_{B1},\ T_1,\ H_2,\ x_{A2},\ x_{B2},\ T_2,\ H_3,\ x_{A3},\ x_{B3},\ T_3]^T \quad (4.74)$$

$$u = \left[F_{f1},\ Q_1,\ F_{f2},\ Q_2,\ F_R,\ Q_3\right]^T \quad (4.75)$$

We linearize the plant model around the steady state defined by Table 4.1 and derive the following linear discrete-time model with sampling time $\Delta = 0.1$ s

$$x^+ = Ax + Bu \quad (4.76)$$

$$y = x \quad (4.77)$$

In order to control the separator and each reactor independently, we partition the plant into 3 subsystems by defining

$$y_1 = [H_1,\ x_{A1},\ x_{B1},\ T_1]^T, \quad u_1 = \left[F_{f1},\ Q_1\right]^T \quad (4.78)$$

$$y_2 = [H_2,\ x_{A2},\ x_{B2},\ T_2]^T, \quad u_2 = \left[F_{f2},\ Q_2\right]^T \quad (4.79)$$

$$y_3 = [H_3,\ x_{A3},\ x_{B3},\ T_3]^T, \quad u_3 = [F_R,\ Q_3]^T \quad (4.80)$$

Following the distributed model derivation, we form the distributed model for the plant.

Consider the performance of distributed control with the parti-

Table 4.1: Nominal parameters

Parameters	Values	Units	Parameters	Values	Units
H_1	29.8	m	A_1	3	m^2
x_{A1}	0.542	wt%	A_2	3	m^2
x_{B1}	0.393	wt%	A_3	1	m^2
T_1	315	K	ρ	0.15	kg/m^3
H_2	30	m	c_p	25	kJ/kgK
x_{A2}	0.503	wt%	k_{v1}	2.5	kg/ms
x_{B2}	0.421	wt%	k_{v2}	2.5	kg/ms
T_2	315	K	k_{v3}	2.5	kg/ms
H_3	3.27	m	x_{A0}	1	wt%
x_{A3}	0.238	wt%	A_2	3	m^2
x_{B3}	0.570	wt%	A_3	1	m^2
T_3	315	K	ρ	0.15	kg/m^3
F_{f1}	8.33	kg/s	E_A/R	-100	K
Q_1	10	kJ/s	E_B/R	-150	K
F_{f2}	0.5	kg/s	ΔH_A	-40	kJ/kg
Q_2	10	kJ/s	ΔH_B	-50	kJ/kg
F_{f3}	66.2	kg/s	α_A	3.5	
Q_3	10	kJ/s	α_B	1.1	
			α_C	0.5	

tioning defined above. The tuning parameters are

$$\begin{aligned} Q_{y1} &= \text{diag}(1,\ 0,\ 0,\ 0.1) \\ Q_{y2} &= \text{diag}(1,\ 0,\ 0,\ 0.1) \\ Q_{y3} &= \text{diag}(1,\ 0,\ 10^3,\ 0) \\ Q_i &= C_i^T Q_{yi} C_i + 0.001 I, \quad \forall i \in \mathbb{I}_{1:3} \\ R_i &= 0.01 I, \quad \forall i \in \mathbb{I}_{1:3} \end{aligned} \tag{4.81}$$

The input constraints are defined in Table 4.2.

Table 4.2: Input constraints

Parameters	Lower bound	Steady state	Upper bound	Units
F_{f1}	0	8.33	10	kg/s
Q_1	0	10	50	kJ/s
F_{f2}	0	0.5	10	kg/s
Q_2	0	10	50	kJ/s
F_{f3}	0	66.2	75	kg/s
Q_3	0	10	50	kJ/s

4.5.2 Simulation results

We simulate the process network in the output product weight percent x_{B3} at $t = 0.5$ s. Estimation results are shown in Fig. 4.5. In Fig. 4.5, the performance of the distributed estimation strategies are compared to the centralized one. For this example, noncooperative estimation is an improvement over decentralized estimation.

4.6 Conclusions

In this chapter, we develop cooperative Kalman filter for large-scale network systems based on the previous result in [12]. The developed estimation algorithm combines decentralized and distributed manners to compensate conventional problems which are local detectability and scalability in network systems. A leak detection and location algorithm based on the presented state estimation in water pipe networks is subsequently developed. We apply the developed cooperative estimation scheme to the developed model in Chapter 3 since its complexity does not grow with the total size of the network and thus it can be applied to large-scale systems where subsystems may be physically interconnected such as water pipe networks. We

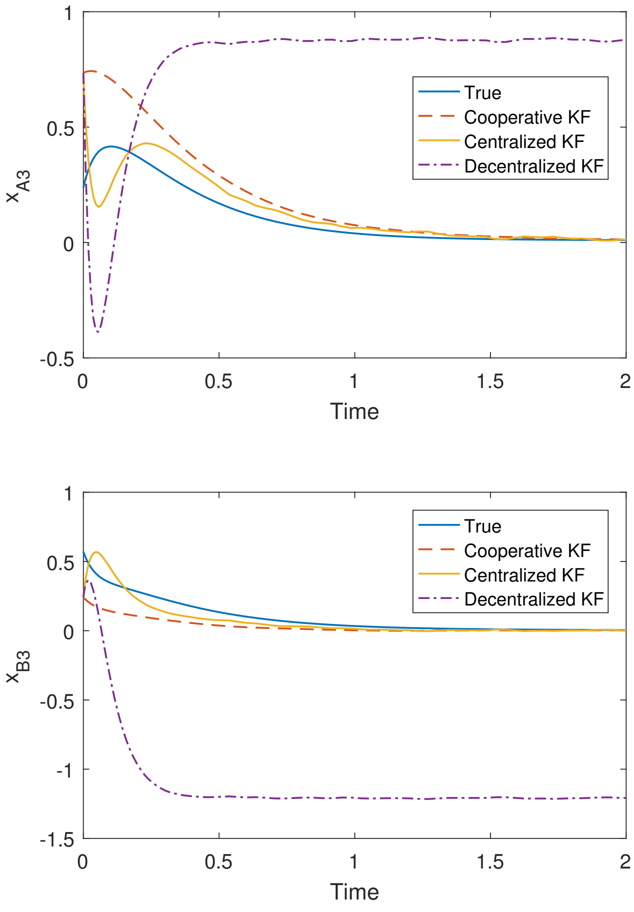


Figure 4.5: Estimation results of cooperative KF

propose an algorithm that can detect and locate the leak based on the cooperative state estimators and from the estimation result, we can find a possible leak location where the corresponding node has the steepest pressure drop and the location error is 9.55 m.

Chapter 5

Cooperative Model Predictive Control of Large-scale Process Networks

Process systems with material and energy recycle are well-known to exhibit complex dynamics and to present significant control challenges, due to the feedback interactions induced by the recycle streams. Chemical plants consist of reaction and separation processes, typically interconnected through material and energy recycle. Recycle can significantly alter the dynamics of a process network, ”slowing down” its overall response, causing high sensitivity to disturbances and giving rise to strongly nonlinear ”overall” dynamics (manifested in the form of multiple steady states, limit cycles etc.). These studies illustrate clearly the challenges caused by the feedback interactions in such networks within a conventional linear control framework. In a broader context, the strong coupling between the control loops in different process units in a chemical plant has been recognized as a major issue that must be addressed in a plant-wide control setting. The motivation arises from (i) the increasing demands for highly integrated chemical plants, tight product quality specifications and tough environmental regulations, and (ii) the inherent complexity and sensitivity of nonlinear controllers designed on the basis of detailed models of entire process networks [88].

5.1 Introduction

Model predictive control (MPC) has been widely adopted in the petrochemical industry for controlling large, multi-variable processes. MPC solves an online optimization to determine inputs, taking into account the current conditions of the plant, any disturbances affecting operation, and imposed safety and physical constraints. Over the last several decades, MPC technology has reached a mature stage. Closed-loop properties are well understood, and nominal stability has been demonstrated for many industrial applications [89]. Chemical plants usually consist of linked unit operations and can be subdivided into a number of subsystems. These subsystems are connected through a network of material, energy, and information streams. Because plants often take advantage of the economic savings available in material recycle and energy integration, the plantwide interactions of the network are difficult to elucidate. Plantwide control has traditionally been implemented in a decentralized fashion, in which each subsystem is controlled independently and network interactions are treated as local subsystem disturbances [90]. It is well known, however, that when the inter-subsystem interactions are strong, decentralized control is unreliable [91].

Centralized control, in which all subsystems are controlled via a single agent, can account for the plantwide interactions. Indeed, increased computational power, faster optimization software, and algorithms designed specifically for large-scale plantwide control have made centralized control more practical [91]. Objections to centralized control are often not computational, however, but organizational. All subsystems rely upon the central agent, making plantwide control difficult to coordinate and maintain. These obstacles deter implementation of centralized control for large-scale plants [44].

As a middle ground between the decentralized and centralized strategies, distributed control preserves the topology and flexibility of decentralized control yet offers a nominal closed-loop stability guarantee. Stability is achieved by two features: the network interactions between subsystems are explicitly modeled and openloop information, usually input trajectories, is exchanged between subsystem controllers. Distributed control strategies differ in the utilization of the open-loop information. In noncooperative distributed control, each subsystem controller anticipates the effect of network interactions only locally [92]. These strategies are described as noncooperative dynamic games, and the plantwide performance converges to the Nash equilibrium. If network interactions are strong, however, noncooperative control can destabilize the plant and performance may be, in these cases, poorer than decentralized control. A more extensive and detailed comparison of cooperative and noncooperative approaches is also provided. Alternatively, cooperative distributed control improves performance by requiring each subsystem to consider the effect of local control actions on all subsystems in the network. Each controller optimizes a plantwide objective function, e.g., the centralized controller objective [44].

5.2 System Model and Repartition

We use the same linear interconnected system model and repartitioned model in Chapter 4 as follows

$$\dot{x}_{k+1} = A_k x_k + \sum_{j=1}^{N} A_{kj} x_j + B_k u + w_k \tag{5.1}$$

$$y_k = C_k x_k + \sum_{j=1}^{N} C_{kj} x_j + v_k \tag{5.2}$$

and the global system can be described as

$$\dot{x} = Ax + Bw, \quad y = Cx + v \tag{5.3}$$

with

$$A = \begin{bmatrix} A_1 & A_{12} & \cdots & A_{1N} \\ A_{21} & A_2 & & \vdots \\ \vdots & & \ddots & \vdots \\ A_{N1} & \cdots & \cdots & A_N \end{bmatrix} \tag{5.4}$$

$$B^{\top} = \begin{bmatrix} B_1^{\top} & \cdots & B_N^{\top} \end{bmatrix} \tag{5.5}$$

$$C = \begin{bmatrix} C_1 & C_{12} & \cdots & C_{1N} \\ C_{21} & C_2 & & \vdots \\ \vdots & & \ddots & \vdots \\ C_{N1} & \cdots & \cdots & C_N \end{bmatrix} \tag{5.6}$$

The resulting repartitioned system is presented as

$$\begin{bmatrix} \dot{x}^{(k)} \\ \dot{x}_c^{(k)} \end{bmatrix} = \begin{bmatrix} A^{(k)} & \tilde{A}^{(k)} \\ \tilde{A}_c^{(k)} & A_c^{(k)} \end{bmatrix} \begin{bmatrix} x^{(k)} \\ x_c^{(k)} \end{bmatrix} + \begin{bmatrix} B^{(k)} \\ B_c^{(k)} \end{bmatrix} w \tag{5.7}$$

$$y_k = C^{(k)} x^{(k)} + v_k \tag{5.8}$$

Also, two assumptions of controllability and stabilizability are used as duality of Assumption 1 and 2.

Assumption 4. *The global plant (A, B) is controllable.*

Assumption 5. *The global plant (A, B) is stabilizable.*

Assumption 4 and 5 are to setup a basic framework under which the control problem under consideration is meaningful.

5.3 Cooperative Model Predictive Control

In this section, cooperative model predictive control (MPC) is developed. The standard MPC is first provided to be used in the cooperative MPC form.

5.3.1 Centralized MPC

We briefly introduce a standard form of MPC for large-scale system. We consider deterministic, nonlinear, continuous-time systems with state x and control u described by [93]:

$$\dot{x} = f(x(t), u(t)), \quad y = h(x), \quad x(0) = x_0 \tag{5.9}$$

subject to input and state constraints of the form:

$$u(t) \in \mathbb{U}, \quad \forall t \geq 0 \tag{5.10}$$

$$x(t) \in \mathbb{X}, \quad \forall t \geq 0 \tag{5.11}$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ denote the vector of states and inputs, respectively. Furthermore, the input constraint set $\mathbb{U}$ is assumed to be compact and $\mathbb{X}$ is connected. For example, $\mathbb{U}$ and $\mathbb{X}$ are often given by box constraints of the form:

$$\mathbb{U} := \{u \in \mathbb{R}^m | u_{\text{min}} \leq u \leq u_{\text{max}}\} \tag{5.12}$$

$$\mathbb{X} := \{x \in \mathbb{R}^n | x_{\text{min}} \leq x \leq x_{\text{max}}\} \tag{5.13}$$

with the constant vectors, u_{min}, u_{max} and x_{min}, x_{max}.

In MPC, the input applied to the system is usually given by the solution of the following finite horizon open-loop optimal control problem which is solved at every sampling instant:

$$\min_{\bar{u}(\cdot)} J\left(x(t), \bar{u}(\cdot)\right) \tag{5.14}$$

subject to

$$\dot{\bar{x}}(\tau) = f\left(\bar{x}(\tau),\ \bar{u}(\tau)\right), \quad \bar{x}(t) = x(t) \tag{5.15}$$

$$\bar{u}(\tau) \in \mathbb{U}, \quad \forall \tau \in [t,\ t + T_c] \tag{5.16}$$

$$\bar{u}(\tau) = \bar{u}(t+T_c), \quad \forall \tau \in [t+T_c,\, t+T_p] \tag{5.17}$$

$$\dot{\bar{x}}(\tau) \in \mathbb{X}, \quad \forall \tau \in [t,\, t+T_p] \tag{5.18}$$

with the cost functional

$$J(x(t), \bar{u}(\cdot)) := \int_t^{t+T_p} F(\bar{x}(\tau), \bar{u}(\tau)) d\tau \tag{5.19}$$

where T_p and T_c are the prediction and the control horizon with $T_c \leq T_p$. The bar denotes internal controller variables and $\bar{x}(\cdot)$ is the solution of Eq. 5.15 driven by the input signal $\bar{u}(\cdot) : [t, t+T_p] \to \mathbb{U}$ under the initial condition $x(t)$. The distinction between the real system variables and the variables in the controller is necessary since even in the nominal case the predicted values will not be the same as the actual closed-loop values. The difference in the predicted and the real values in due to determination of the applied input via a re-optimization over a moving finite horizon T_c at every sampling instant.

The cost functional J is defined in terms of the stage cost F which specifies the performance. The stage cost can for example arise from economical a ecological considerations. Often, a quadratic form for F is used:

$$F(x,u) = (x - x_s)^T Q(x - x_s) + (u - u_s)^T R(u - u_s) \tag{5.20}$$

where x_s and u_s denote a desired reference trajectory that can be constant or time-varying. The deviation from the desired values is weighted by the positive definite matrices Q and R. In the case of a stabilization problem (no tracking), *i.e.*, $x_s = u_x =$ const, one can assume, without loss of generality, that $(x_s, u_x) = (0,0)$ is the steady state to stabilize.

The state measurement enters the system via the initial condition in Eq. 5.15 at the sampling instants, *i.e.*, the system model used to predict the future system behavior is initialized by the actual system state. Since all state information is required for the prediction, the full state must be either measured or estimated. Eq. 5.17 fixes the input beyond the control horizon to $\bar{u}(t+T_c)$.

In the following, optimal solutions of Eq. 5.14 are denoted by $\bar{u}^*(\cdot;x(t)) : [t,\ t+T_p] \to \mathbb{U}$. The open-loop optimal control problem is solved repeatedly at the sampling instants $t_j = j\delta,\ j = 0,1,\cdots$, and the input applied to the system is given by the sequence of optimal solutions of Eq. 5.14:

$$u(t) := \bar{u}^*(t;x(t)) \tag{5.21}$$

where t_j is the closest sampling instant to t with $t_j \leq t$. Thus, the nominal closed-loop system is given by

$$\dot{x}(t) = f(x(t), \bar{u}^*(t;x(t))) \tag{5.22}$$

The optimal cost of Eq. 5.14 as a function of the state is referred to as value function V and is given by

$$V(x) = J(x, \bar{u}^*(\cdot;x)) \tag{5.23}$$

The value function plays a critical role in the stability analysis of

NMPC since it often serves as a Lyapunov function candidate.

5.3.2 Cooperative MPC

The objective function is in the same form as Eq. 5.20 of standard MPC as follows

- Objective:

$$\begin{aligned}\min_{\Delta u} J^{(k)} &= \min_{\Delta u} \sum_{j=1}^{p} \left(r^{(k)}(k+j|k) - y^{(k)}(k+j|k) \right)^{\top} Q^{(k)} \\ &\times \left(r^{(k)}(k+j|k) - y^{(k)}(k+j|k) \right) \\ &+ \sum_{l=1}^{m} \Delta u^{(k)\top}(k+l|k) R \Delta u^{(k)}(k+j|k)\end{aligned} \tag{5.24}$$

by using the repartitioned model for subsystem k and substituting y and r for x and x_s, respectively.

After Δu is calculated from the objective function, $u^{(k)}$ is updated as follows to include overlapped input variables

- Input update:

$$u_{\text{new}}^{(k)} = u^{(k)} + K^{(k)} \sum_{\lambda \in \mathcal{N}_k} \left(\sum_{\lambda \in I_c^{(k)} \cap I^{(j)}} e_{\xi^{-1}(\lambda)} \left(u_\lambda^{(j)} - u_\lambda^{(k)} \right) \right) \tag{5.25}$$

The control gain K should be chosen properly to satisfy certain control performance.

The dynamics at the next sample time is obtained by using $u_{\text{new}}^{(k)}$ of Eq. 5.25. Then, the state should be updated to incorporate the physically interconnected states in other subsystems.

- State update:

$$\dot{x}^{(k)} = A^{(k)}x^{(k)} + B^{(k)}u^{(k)} + \sum_{\lambda \in I_c^{(k)}} \left[\tilde{A}^{(k)}\right]_{\lambda} x_{\lambda}^{\zeta(\lambda)} + \sum_{\lambda \in I_c^{(k)}} \left[\tilde{B}^{(k)}\right]_{\lambda} u_{\lambda}^{\zeta(\lambda)} \tag{5.26}$$

With the proposed cooperative MPC, the decentralized MPC performance can be compensated by updating the state of Eq. 5.26. We demonstrate the performance of cooperative MPC in the following section.

5.4 Application to Chemical Process Networks with Recycles

We use the same chemical process model introduced in Section 4.5 and the control results are shown in Fig. 5.1. As shown in Fig. 5.1, cooperative MPC shows a similar performance of the centralized one. The performance difference is 0.007 at steady state. It is also noticed that decentralized MPC cannot follow the overall process dynamics since it does not involve the dynamics of other subsystems which interact with.

5.5 Conclusions

In this paper we present a novel cooperative distributed controller in which the subsystem controllers optimize the same objective function in parallel without the use of a coordinator. The control algorithm is similar to a centralized MPC controller. The cooperative MPC improves conventional issues in decentralized and distributed MPC in that it does not require local controllability of subsystems and does not grow with the overall system size in the network. These prop-

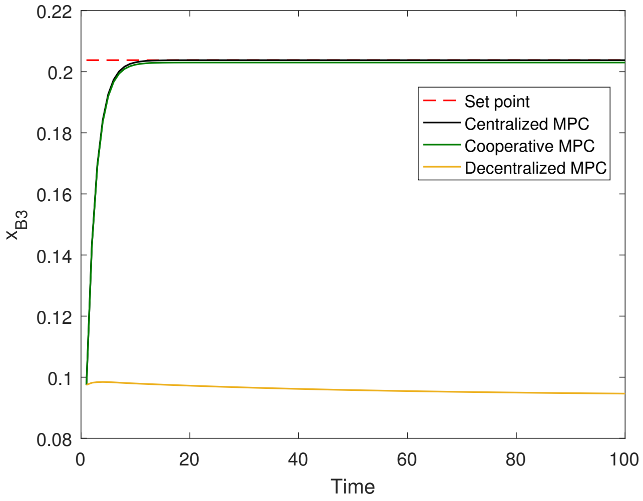


Figure 5.1: Cooperative MPC results

erties lies with the cooperative Kalman filter developed in Chapter 4. Furthermore, the cooperative MPC can improve performance of plants over traditional decentralized control and non-cooperative control, especially for plants with strong open-loop interactions between subsystems. A Process network example is given showing this performance improvement.

Chapter 6

Concluding Remarks

Overall summary and concluding remarks of the thesis are provided below and future works in the subsequence of the thesis also provided.

6.1 Concluding Remarks

In this thesis, we develop an online burst detection and location system of water pipe networks based on the CUSUM and the DWT algorithms, and propose a new node matrix to represent a pipe network with every link less than error bounds. In the monitoring unit, the CUSUM algorithm gives a robust sum to mean changes of data; but at the same time, it gives a slow detection, and thus deteriorates the detection accuracy. While the DWT may not be suitable for global event detection because of the high false alarm rate, the method allows the sudden transition of data to be exactly found. We combine these two techniques to take advantage of their properties, and obtain better location performance than the previous works. In the simple and controlled network, the average and maximum location errors were 8 and 31 m. In rather complex and uncontrolled network, the errors were 22 and 39 m, respectively. The efficacy of the developed algorithm was validated with both cases, and it shows a better result among those applied to real water supply systems up to the present. The pressure data obtained from the real burst accident were also used to verify the proposed system. In addition, a software program with the proposed algorithms has been completely developed and a pilot test is being carried out.

The fundamental model of NPW dynamics due to leaks in water pipe networks is developed based on a consensus algorithm and water hammer theory. The leak dynamics model of water pipe networks has hardly been considered in the literature thus far because of the complexity of the water pipe network structure and hydraulics. The developed model has a form of a simple and linear model of an interconnected network system. Then, it is satisfactorily validated using real experimental data obtained from a field test. The average and maximum errors between the model and experiment are within 2 and 20%, respectively, and a large portion of these errors are attributed to fluctuations of the NPW propagation through the network, which is difficult to precisely model. Although the model requires parameter adjustment to accurately simulate the real water pipe network, it can be used for various applications, such as the development of a leak detection and location algorithm. A leak detection and location algorithm based on state estimation in water pipe networks is subsequently developed. We apply cooperative and distributed estimation scheme to the developed model since its complexity does not grow with the total size of the network and thus it can be applied to large-scale systems where subsystems may be physically interconnected such as water pipe networks. We propose an algorithm that can detect and locate the leak based on the cooperative state estimators and from the estimation result, we can find a possible leak location where the corresponding node has the steepest pressure drop and the location error is 9.55 m.

In this thesis, we presented a Kalman filter-based approach to cooperative state estimation for linear interconnected large-scale systems, such as multi-agent systems. In order to achieve scalability of the estimation setup, we required the local estimators to estimate local states only. We establish an algorithm for interconnecting the local estimators, whereby both physical couplings and detectability issues can be handled. Moreover, design conditions are presented to guarantee Kalman filter-performance with respect to both model and measurement disturbances. The developed estimation algorithm combines decentralized and distributed manners to compensate conventional problems which are local detectability and scalability in network systems. A leak detection and location algorithm based on the presented state estimation in water pipe networks is subsequently developed. We apply the developed cooperative estimation scheme to the developed model in Chapter 3 since its complexity does not grow with the total size of the network and thus it can be applied to large-scale systems where subsystems may be physically interconnected such as water pipe networks. We propose an algorithm that can detect and locate the leak based on the cooperative state estimators and from the estimation result, we can find a possible leak location where the corresponding node has the steepest pressure drop and the location error is 9.55 m.

Finally, we present a novel cooperative distributed controller in which the subsystem controllers optimize the same objective function in parallel without the use of a coordinator. The control algorithm is similar to a centralized MPC controller. The cooperative MPC improves conventional issues in decentralized and distributed MPC in that it does not require local controllability of subsystems and does not grow with the overall system size in the network. These properties lies with the cooperative Kalman filter developed in Chapter 4. Furthermore, the cooperative MPC can improve performance of plants over traditional decentralized control and non-cooperative control, especially for plants with strong open-loop interactions between subsystems. A Process network example is given showing this performance improvement.

6.2 Future Directions

Future works include plant-wide cooperative economic model predictive control (EMPC) based on the same concepts and network framework used to develop cooperative KF and MPC in the thesis. Also, stochastic cooperative MPC and stochastic cooperative EMPC will be considered in the future work.

Bibliography

[1] R. Olfati-Saber, J. A. Fax, R. M. Murray, Consensus and cooperation in networked multi-agent systems, Proceedings of the IEEE 95 (1) (2007) 215–233.

[2] D. Simon, Optimal state estimation: Kalman, H infinity, and nonlinear approaches, John Wiley & Sons, 2006.

[3] A. Ahmad, M. Gani, F. Yang, Decentralized robust kalman filtering for uncertain stochastic systems over heterogeneous sensor networks, Signal Processing 88 (8) (2008) 1919–1928.

[4] M. Kamgarpour, C. Tomlin, Convergence properties of a decentralized kalman filter, in: Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, IEEE, 2008, pp. 3205–3210.

[5] S. S. Stankovic, M. S. Stankovic, D. M. Stipanovic, Consensus based overlapping decentralized estimator, IEEE Transactions on Automatic Control 54 (2) (2009) 410–415.

[6] R. Olfati-Saber, Distributed kalman filter with embedded consensus filters, in: Proceedings of the 44th IEEE Conference on Decision and Control, IEEE, 2005, pp. 8179–8184.

[7] U. A. Khan, J. M. Moura, Distributing the kalman filter for large-scale systems, IEEE Transactions on Signal Processing 56 (10) (2008) 4919–4935.

[8] F. S. Cattivelli, A. H. Sayed, Diffusion strategies for distributed kalman filtering and smoothing, IEEE Transactions on automatic control 55 (9) (2010) 2069–2084.

[9] S. Kar, J. M. Moura, Gossip and distributed kalman filtering: weak consensus under weak detectability, IEEE Transactions on Signal Processing 59 (4) (2011) 1766–1784.

[10] H. Bai, R. A. Freeman, K. M. Lynch, Distributed kalman filtering using the internal model average consensus estimator, in: Proceedings of the 2011 American Control Conference, IEEE, 2011, pp. 1500–1505.

[11] D. Y. Kim, M. Jeon, Robust distributed kalman filter for wireless sensor networks with uncertain communication channels, Mathematical Problems in Engineering 2012.

[12] J. Wu, V. Ugrinovskii, F. Allgöwer, Cooperative ∞-estimation for large-scale interconnected linear systems, in: 2015 American Control Conference (ACC), IEEE, 2015, pp. 2119–2124.

[13] A. Richards, J. How, A decentralized algorithm for robust constrained model predictive control, in: American Control Conference, 2004. Proceedings of the 2004, Vol. 5, IEEE, 2004, pp. 4261–4266.

[14] A. Richards, J. How, Decentralized model predictive control of cooperating uavs, in: Decision and Control, 2004. CDC. 43rd IEEE Conference on, Vol. 4, IEEE, 2004, pp. 4286–4291.

[15] T. Keviczky, F. Borrelli, G. J. Balas, Decentralized receding horizon control for large scale dynamically decoupled systems, Automatica 42 (12) (2006) 2105–2115.

[16] A. Alessio, A. Bemporad, Decentralized model predictive control of constrained linear systems, in: Control Conference (ECC), 2007 European, IEEE, 2007, pp. 2813–2818.

[17] Y. Wakasa, M. Arakawa, K. Tanaka, T. Akashi, Decentralized model predictive control via dual decomposition, in: Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, IEEE, 2008, pp. 381–386.

[18] M. Vaccarini, S. Longhi, M. R. Katebi, Unconstrained networked decentralized model predictive control, Journal of Process Control 19 (2) (2009) 328–339.

[19] A. Alessio, D. Barcelli, A. Bemporad, Decentralized model predictive control of dynamically coupled linear systems, Journal of Process Control 21 (5) (2011) 705–714.

[20] C. Ocampo-Martinez, D. Barcelli, V. Puig, A. Bemporad, Hierarchical and decentralised model predictive control of drinking water networks: Application to barcelona case study, IET control theory & applications 6 (1) (2012) 62–71.

[21] E. Camponogara, D. Jia, B. H. Krogh, S. Talukdar, Distributed model predictive control, IEEE Control Systems 22 (1) (2002) 44–52.

[22] S. Li, Y. Zhang, Q. Zhu, Nash-optimization enhanced distributed model predictive control applied to the shell benchmark problem, Information Sciences 170 (2) (2005) 329–349.

[23] A. N. Venkat, J. B. Rawlings, S. J. Wright, Stability and optimality of distributed model predictive control, in: Proceedings of the 44th IEEE Conference on Decision and Control, IEEE, 2005, pp. 6680–6685.

[24] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, S. J. Wright, Distributed output feedback mpc for power system control, in: Proceedings of the 45th IEEE Conference on Decision and Control, IEEE, 2006, pp. 4038–4045.

[25] A. N. Venkat, J. B. Rawlings, S. J. Wright, Implementable distributed model predictive control with guaranteed performance properties, in: 2006 American Control Conference, IEEE, 2006, pp. 6–pp.

[26] W. B. Dunbar, Distributed receding horizon control of dynamically coupled nonlinear systems, IEEE Transactions on Automatic Control 52 (7) (2007) 1249–1263.

[27] M. Mercangöz, F. J. Doyle, Distributed model predictive control of an experimental four-tank system, Journal of Process Control 17 (3) (2007) 297–308.

[28] A. Richards, J. How, Robust distributed model predictive control, International Journal of control 80 (9) (2007) 1517–1531.

[29] I. Necoara, D. Doan, J. A. Suykens, Application of the proximal center decomposition method to distributed model predictive control, in: Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, IEEE, 2008, pp. 2900–2905.

[30] J. Liu, D. Muñoz de la Peña, P. D. Christofides, Distributed model predictive control of nonlinear process systems, AIChE Journal 55 (5) (2009) 1171–1184.

[31] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, S. J. Wright, Distributed mpc strategies with application to power system automatic generation control, IEEE transactions on control systems technology 16 (6) (2008) 1192–1206.

[32] J. Maestre, D. M. De La Peña, E. F. Camacho, A distributed mpc scheme with low communication requirements, in: 2009 American Control Conference, IEEE, 2009, pp. 2797–2802.

[33] Y. Zheng, S. Li, X. Wang, Distributed model predictive control for plant-wide hot-rolled strip laminar cooling process, Journal of Process Control 19 (9) (2009) 1427–1437.

[34] P. Giselsson, A. Rantzer, Distributed model predictive control with suboptimality and stability guarantees, in: 49th IEEE Conference on Decision and Control (CDC), IEEE, 2010, pp. 7272–7277.

[35] J. Liu, X. Chen, D. Muñoz de la Peña, P. D. Christofides, Sequential and iterative architectures for distributed model predictive control of nonlinear process systems, AIChE Journal 56 (8) (2010) 2137–2149.

[36] M. D. Doan, T. Keviczky, B. De Schutter, An iterative scheme for distributed model predictive control using fenchel's duality, Journal of Process Control 21 (5) (2011) 746–755.

[37] J. Maestre, D. M. De La Pena, E. Camacho, T. Alamo, Distributed model predictive control based on agent negotiation, Journal of Process Control 21 (5) (2011) 685–697.

[38] J. Maestre, D. Muñoz De La Peña, E. Camacho, Distributed model predictive control based on a cooperative game, Optimal Control Applications and Methods 32 (2) (2011) 153–176.

[39] H. Scheu, W. Marquardt, Sensitivity-based coordination in distributed model predictive control, Journal of Process Control 21 (5) (2011) 715–728.

[40] Y. Zheng, S. Li, N. Li, Distributed model predictive control over network information exchange for large-scale systems, Control engineering practice 19 (7) (2011) 757–769.

[41] M. Farina, R. Scattolini, Distributed predictive control: a non-cooperative algorithm with neighbor-to-neighbor communication for linear systems, Automatica 48 (6) (2012) 1088–1096.

[42] J. Liu, X. Chen, D. M. de la Peña, P. D. Christofides, Iterative distributed model predictive control of nonlinear systems: Handling asynchronous, delayed measurements, IEEE Transactions on Automatic Control 57 (2) (2012) 528–534.

[43] P. Giselsson, A. Rantzer, On feasibility, stability and performance in distributed model predictive control, IEEE Transactions on Automatic Control 59 (4) (2014) 1031–1036.

[44] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, G. Pannocchia, Cooperative distributed model predictive control, Systems & Control Letters 59 (8) (2010) 460–469.

[45] B. T. Stewart, S. J. Wright, J. B. Rawlings, Cooperative distributed model predictive control for nonlinear systems, Journal of Process Control 21 (5) (2011) 698–704.

[46] M. A. Müller, M. Reble, F. Allgöwer, Cooperative control of dynamically decoupled systems via distributed model predictive control, International Journal of Robust and Nonlinear Control 22 (12) (2012) 1376–1397.

[47] A. Ferramosca, D. Limón, I. Alvarado, E. F. Camacho, Cooperative distributed mpc for tracking, Automatica 49 (4) (2013) 906–914.

[48] P. Trodden, A. Richards, Cooperative distributed mpc of linear systems with coupled constraints, Automatica 49 (2) (2013) 479–487.

[49] L. Dai, Y. Xia, Y. Gao, B. Kouvaritakis, M. Cannon, Cooperative distributed stochastic mpc for systems with state estimation and coupled probabilistic constraints, Automatica 61 (2015) 89–96.

[50] S. Meniconi, B. Brunone, M. Ferrante, C. Massari, Potential of transient tests to diagnose real supply pipe systems: What can be done with a single extemporary test, Journal of Water Resources Planning and Management 137 (2) (2010) 237–241.

[51] D. Kang, K. Lansey, Novel approach to detecting pipe bursts in water distribution networks, Journal of Water Resources Planning and Management 140 (1) (2012) 121–127.

[52] B. Brunone, M. Ferrante, S. Meniconi, C. Massari, Effectiveness assessment of pipe systems by means of transient test-based techniques, Procedia Environmental Sciences 19 (2013) 814–822.

[53] D. Jung, K. Lansey, Water distribution system burst detection using a nonlinear kalman filter, Journal of Water Resources Planning and Management 10.1061/(ASCE)WR.1943-5452.0000464 (2014) 04014070.

[54] D. Loureiro, C. Amado, A. Martins, D. Vitorino, A. Mamade, S. T. Coelho, Water distribution systems flow monitoring and anomalous event detection: A practical approach, Urban Water Journal (ahead-of-print) (2015) 1–11.

[55] D. Covas, H. Ramos, Case studies of leak detection and location in water pipe systems by inverse transient analysis, Journal of Water Resources Planning and Management 136 (2) (2010) 248–257.

[56] J. Schwaller, J. van Zyl, Modeling the pressure-leakage response of water distribution systems based on individual leak behavior, Journal of Hydraulic Engineering 10.1061/(ASCE)HY.1943-7900.0000984 (2014) 04014089.

[57] Á. Martínez-Codina, L. Cueto-Felgueroso, M. Castillo, L. Garrote, Use of pressure management to reduce the probability of pipe breaks: A bayesian approach, Journal of Water Resources Planning and Management 10.1061/(ASCE)WR.1943-5452.0000519 (2015) 04015010.

[58] C. Ge, G. Wang, H. Ye, Analysis of the smallest detectable leakage flow rate of negative pressure wave-based leak detection systems for liquid pipelines, Computers & Chemical Engineering 32 (8) (2008) 1669–1680.

[59] D. Misiunas, M. Lambert, A. Simpson, G. Olsson, Burst detection and location in water distribution networks, Water Science & Technology: Water Supply 5 (3) (2005) 71–80.

[60] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, F. J. Doyle III, Process dynamics and control, John Wiley & Sons, 2010.

[61] T. T. Liu, A. C. Fraser-Smith, Detection of transients in 1/f noise with the undecimated discrete wavelet transform, Signal Processing, IEEE Transactions on 48 (5) (2000) 1458–1462.

[62] Z. Nenadic, J. W. Burdick, Spike detection using the continuous wavelet transform, Biomedical Engineering, IEEE Transactions on 52 (1) (2005) 74–87.

[63] A. Y. Goharrizi, N. Sepehri, A wavelet-based approach to internal seal damage diagnosis in hydraulic actuators, Industrial Electronics, IEEE Transactions on 57 (5) (2010) 1755–1763.

[64] J. J. Liu, C. Han, Wavelet texture analysis in process industries, Korean Journal of Chemical Engineering 28 (9) (2011) 1814–1823.

[65] M. Ferrante, B. Brunone, S. Meniconi, Wavelets for the analysis of transient pressure signals for leak detection, Journal of Hydraulic Engineering 133 (11) (2007) 1274–1282.

[66] M. Ferrante, S. Meniconi, Leak-edge detection, Journal of Hydraulic Research 47 (2) (2009) 233–241.

[67] M. Ahadi, M. S. Bakhtiar, Leak detection in water-filled plastic pipes through the application of tuned wavelet transforms to acoustic emission signals, Applied Acoustics 71 (7) (2010) 634–639.

[68] S. Meniconi, B. Brunone, M. Ferrante, C. Massari, Small amplitude sharp pressure waves to diagnose pipe systems, Water Resources Management 25 (1) (2011) 79–96.

[69] S. Srirangarajan, M. Allen, A. Preis, M. Iqbal, H. B. Lim, A. J. Whittle, Wavelet-based burst event detection and localization in water distribution systems, Journal of Signal Processing Systems 72 (1) (2013) 1–16.

[70] H. Li, D.-Y. Xiao, X. Zhao, A field-pipeline leakage detection method based on negative pressure wave and improved fast differential algorithm, in: Information and Computing Science, 2009 IEEE International Conference on, IEEE, 2009, pp. 127–130.

[71] C. H. Tian, J. C. Yan, J. Huang, Y. Wang, D.-S. Kim, T. Yi, Negative pressure wave based pipeline leak detection: Challenges and algorithms, in: Service Operations and Logistics, and Informatics, 2012 IEEE International Conference on, IEEE, 2012, pp. 372–376.

[72] M. Basseville, Detecting changes in signals and systems-a survey, Automatica 24 (3) (1988) 309–326.

[73] Y. Choe, Application and implementation of scale-space filtering techniques for qualitative interpretation of real-time process data, Ph.D. thesis, Seoul National University, in Korean (1995).

[74] W. Ming, W. Wei-qiang, Application of wavelet to detect pipeline leak point, in: Intelligent Systems Design and Applications, 2006 IEEE International Conference on, IEEE, 2006, pp. 779–782.

[75] C. Ma, S. Yu, J. Huo, Negative pressure wave-flow testing gas pipeline leak based on wavelet transform, in: Computer, Mechatronics, Control and Electronic Engineering, 2010 IEEE International Conference on, IEEE, 2010, pp. 306–308.

[76] O. Begovich, A. Navarro, E. N. Sanchez, G. Besancon, Comparison of two detection algorithms for pipeline leaks, in: Control Applications, 2007 IEEE International Conference on, IEEE, 2007, pp. 777–782.

[77] E. W. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1 (1) (1959) 269–271.

[78] M. Ferrante, B. Brunone, S. Meniconi, B. W. Karney, C. Massari, Leak size, detectability and test conditions in pressurized pipe systems, Water Resources Management 28 (13) (2014) 4583–4598.

[79] A. F. Colombo, P. Lee, B. W. Karney, A selective literature review of transient-based leak detection methods, Journal of Hydro-environment Research 2 (4) (2009) 212–227.

[80] G. Ye, R. A. Fenner, Kalman filtering of hydraulic measurements for burst detection in water distribution systems, Journal of pipeline systems engineering and practice 2 (1) (2010) 14–22.

[81] H. Emara-Shabaik, Y. Khulief, I. Hussaini, A non-linear multiple-model state estimation scheme for pipeline leak detection and isolation, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 216 (6) (2002) 497–512.

[82] T. Maschler, D. A. Savic, Simplification of water supply network models through linearisation, Cent. Water Syst. Rep. 99 1 (1999) 119.

[83] J. Burgschweiger, B. Gnädig, M. C. Steinbach, Optimization models for operative planning in drinking water networks, Optimization and Engineering 10 (1) (2009) 43–73.

[84] K. Boussafeur-Lamoudi, Automated simplification of water networks models.

[85] W. Ren, R. W. Beard, Overview of consensus algorithms in cooperative control, Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications (2008) 3–22.

[86] M. S. Ghidaoui, M. Zhao, D. A. McInnis, D. H. Axworthy, A review of water hammer theory and practice, Applied Mechanics Reviews 58 (1) (2005) 49–76.

[87] S. J. Lee, G. Lee, J. C. Suh, J. M. Lee, Online burst detection and location of water distribution systems and its practical applications, Journal of Water Resources Planning and Management 142 (1) (2015) 04015033.

[88] A. Kumar, P. Daoutidis, Nonlinear dynamics and control of process systems with recycle, Journal of Process Control 12 (4) (2002) 475–484.

[89] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. Scokaert, Constrained model predictive control: Stability and optimality, Automatica 36 (6) (2000) 789–814.

[90] N. Sandell, P. Varaiya, M. Athans, M. Safonov, Survey of decentralized control methods for large scale systems, IEEE Transactions on automatic Control 23 (2) (1978) 108–128.

[91] H. Cui, E. W. Jacobsen, Performance limitations in decentralized control, Journal of Process Control 12 (4) (2002) 485–494.

[92] R. A. Bartlett, L. T. Biegler, J. Backstrom, V. Gopal, Quadratic programming algorithms for large-scale model predictive control, Journal of Process Control 12 (7) (2002) 775–795.

[93] F. Allgöwer, R. Findeisen, Z. K. Nagy, Nonlinear model predictive control: From theory to application, J. Chin. Inst. Chem. Engrs 35 (3) (2004) 299–315.

초 록

본 연구는 대규모 공정 네트워크의 협동 추정 및 제어에 관한 연구이며 기존 대규모 공정의 추정 및 제어 알고리즘의 단점을 보완한 새로운 알고리즘을 제안한다. 또한 대규모 시스템의 한 가지 예로 주로 대규모 상수관망을 대상으로 하여 모델링 및 추정을 통해 이상 진단 및 감지 알고리즘을 개발한다.

상수관망에서 누수, 파열 등의 이상이 발생할 경우 시스템의 크기 및 복잡성으로 인해 이를 감지 및 진단하는 것이 매우 어렵다. 또한 상수관망의 이상으로 인해 발생한 압력 전파 모델이 존재하지 않기 때문에 시스템 모델 없이 효과적으로 이상감지 및 진단하는 기법을 제안하고자 하였다. 기존에 화학공정에서 이상감지를 위해 많이 쓰이는 통계적 기법인 cumulative sum(CUSUM)과 특이점을 빠르고 정확하게 감지할 수 있는 discrete wavelet transform(DWT)을 통합한 새로운 감지 알고리즘을 제안하였고, 이상감지 결과를 이용하여 대규모 상수관망에서 간단한 최적화 해법으로 이상의 위치를 진단하는 알고리즘을 제안하였다. 실제 상수관망의 이상 실험 데이터를 사용하여 제안된 알고리즘을 검증하였고 진단 오차가 30 m 이내로 기존 기술 대비 이상 진단 오차를 현저히 줄임을 확인하였다.

상수관망의 압력 전파 모델이 존재한다면 상태추정(state estimation)을 이용하여 데이터 기반의 알고리즘에 비해 쉽고 빠르게 이상감지 및 진단을 할 수 있기 때문에 상수관망의 이상으로 인한 압력 전파 모델을 개발하고자 하였다. 모델링을 위해 consensus 알고리즘이라는, 네트워크에서의 노드 간의 상태(state)를 나타내는 알고리즘을 이용하였고 consensus 알고리즘을 상수관망에 맞게 수정하여 복잡한 압력 전파 모델을 선형의 간단한 형태로 나타낼 수

있었다. 이를 실제 실험 데이터와 비교하여 제안한 consensus 알고리즘 기반 모델이 실제 압력 전파 다이나믹스를 15% 이내의 오차로 나타낼 수 있음을 보였다.

다음으로 위에서 개발한 모델을 이용하여 대규모 네트워크 시스템에서의 상태추정 기법을 적용한 새로운 이상 감지 및 진단 알고리즘을 제안하였다. 기존의 칼만필터(Kalman filter) 등의 상태추정 방법은 상수관망과 같은 대규모 시스템에 적용될 경우 시스템의 규모가 매우 크기 때문에 계산량 등의 문제가 발생한다. 이를 해결하기 위해 전체 시스템을 여러 개의 서브시스템으로 나눈 decentralized 추정 방법이 연구가 되었다. 그러나 이는 서브시스템 간 상호작용을 고려하지 않기 때문에 이를 보완하기 위해 distributed 추정이 연구되었지만 이 방식은 전체 시스템의 크기가 커짐에 따라서 서브시스템의 estimator의 크기 또한 커지는, 즉 scalability가 없다는 단점이 있다. 본 연구에서 이러한 기존의 방법들을 보완한 새로운 cooperative estimation 알고리즘을 제안하였다. Cooperative state estimation을 상수관망 뿐만 아니라 대규모 화학공정에도 적용하여 decentralized 그리고 distributed 방식의 단점을 보완하면서 centralized estimation과 유사한 성능을 가짐을 보였다.

마지막으로, cooperative estimation 개발에 사용한 핵심 아이디어를 제어 알고리즘에 똑같이 적용하여 cooperative model predictive control(cooperative MPC)을 제안하였다. Cooperative MPC 또한 대규모 공정 네트워크의 제어에 있어 기존의 decentralized 또는 distributed MPC의 단점을 보완하는 새로운 알고리즘으로, 대규모 화학공정에 적용하여 centralized MPC와 유사한 성능을 보임을 증명하였다. 본 박사 논문에서 제시한 대규모 공정의 추정 및 제어를 위한 cooperative KF 그리고 cooperative MPC를 사용하여 기존의 centralized의 계산량 문제, decentralized의 상호작용 문제, 그리고 distributed의 scalability 문제를 해결한 새로운 추정 및 제어가 가능

하다.

주요어 : 상수관망, 대규모 공정, 네트워크 시스템, 분산 협동 칼만 필터, 분산 협동 모델예측제어
학번 : 2013-30285