ANALYSIS AND DESIGN OF MULTI-AGENT SYSTEMS UNDER COMMUNICATION AND PRIVACY CONSTRAINTS

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

Vaibhav Katewa

Vijay Gupta, Director

Graduate Program in Electrical Engineering
Notre Dame, Indiana
December 2016

ANALYSIS AND DESIGN OF MULTI-AGENT SYSTEMS UNDER COMMUNICATION AND PRIVACY CONSTRAINTS

Abstract

by

Vaibhav Katewa

This dissertation presents techniques for design and analysis of multi-agent distributed systems with control oriented objectives. We study two problems: one related to networked estimation in Networked Control Systems and the other related to privacy in Cyber-physical systems.

In the first problem, we focus on congestion control in a communication network that is supporting remote estimation of multiple processes. A stochastic rate control protocol is developed using the network utility maximization framework. This decentralized protocol avoids congestion by regulating the transmission probabilities of the sources. The presence of estimation costs poses new challenges; however, for low congestion levels, the form of rate controller resembles that of the standard TCP rate controller. Stability of the protocol is analyzed in the presence of fixed network delays.

In the second problem, we address the issue of privacy of agents in a multi-agent LTI system which is monitored by a control center via the measurements sent to it by the agents. We show that such architecture is prone to privacy breaches in which an intruder can gain access to agents' sensitive parameters that govern their dynamics. To prevent this, we employ the differential privacy framework and develop a noise adding privacy mechanism in which the agents add synthetic noise while sending their

measurements to the control center. We design the privacy noise by characterizing the sensitivity of the system. We substantiate our framework by studying two concrete examples of second-order consensus and LQR control. Our numerical results show that in an asymptotic regime of low privacy and high SNR, the privacy noise results in marginal performance degradation at the control center, when compared to the error suffered by the intruder in identifying the sensitive parameters.

We study another related privacy problem for a scenario where multiple agents cooperatively solve a quadratic optimization problem. To maintain privacy of their states over time, agents implement a noise-adding mechanism according to the classic differential privacy framework. We characterize how the noise due to the privacy mechanism degrades the performance of the multi-agent system. Interestingly, we show that depending on the desired level of privacy (and thus noise), the system performance is optimized by reducing the level of cooperation among the agents. The notion of cooperation level models the trust of an agent towards the information received from neighboring agents. For the prototypical examples of consensus and centroidal Voronoi tessellations, we are able to characterize the optimum cooperation level that maximizes the system performance while ensuring a desired privacy level. Our results suggest that for the class of problems we study, and in fact for a broad class of multi-agent systems, it is always beneficial for the agents to reduce their cooperation level when the privacy level increases.

CONTENTS

FIGURES

ACKNOWLEDGMENTS

Foremost, I would like to express my gratitude to my advisor, Dr. Vijay Gupta, for his guidance, encouragement and constant support. I benefited a lot from our discussions and his supervision helped me stay on track throughout my research. I would also like to thank Dr. Fabio Pasqualetti and Dr. Aranya Chakrabortty for their valuable ideas and inputs for problems related to cyber-physical privacy. I also thank my research group members for numerous insightful discussions that helped me getting my question answered. A large amount of credit goes to my friends who made my stay enjoyable at Notre Dame. Finally, I dedicate this dissertation to my parents and my wife Shweta. Without their love and constant support, this dissertation would not have been possible.

SYMBOLS

$\mathbb{R}^k$	k-dimensional Euclidean space
$X = [x_{ij}]$	matrix X with $(i,j)^{th}$ entry as x_{ij}
$\|.\|_p$	p-norm of a vector or induced p-norm of a matrix, 2-norm if the subscript is absent
$\|X\|_F$	Frobenius norm of matrix X
$X > 0 (X \geq 0)$	positive (semi-)definite matrix X
$tr(.)$	trace of a matrix
$\lambda(X)$	eigenvalue of square matrix X
$\nu_\lambda(X), \tilde{\nu}_\lambda(X)$	right and left eigenvectors corresponding to eigenvalue λ
$\lambda_{min}(X), \lambda_{max}(X)$	minimum and maximum eigenvalues of Hermitian matrix X
$\rho(X)$	spectral radius of square matrix X
$\kappa(X)$	condition number of matrix $\left(\triangleq ||X||_2 ||X^{-1}||_2\right)$
$diag(X)$	diagonal matrix consisting of diagonal entries of X
$diag\{x\}_{x\in\mathcal{X}}$	diagonal matrix consisting of elements of set $\mathcal{X}$
$diag(X_1, \cdots, X_k)$	block diagonal matrix consisting of blocks $X_1, \cdots, X_k$
$\{x(k)\}_{k\geq 0}/x[0:\infty]$	trajectory/infinite sequence $x(0), x(1), x(2) \cdots$
$\{x(k)\}_{k=0}^{T}/x[0:T]$	truncated trajectory/sequence $x(0), x(1), x(2) \cdots, x(T)$
$\|\mathcal{X}\|$	cardinality of set $\mathcal{X}$
$(x)^+$	$\max\{x, 0\}$
$Re(.), Im(.)$	real and imaginary parts of a complex number
$\otimes$	Kronecker product
$\delta(t), \delta(k)$	Dirac and Kronecker delta functions

I_N	$N \times N$ identity matrix
$\mathbf{1}_N$	all one vector $[1, 1, \cdots, 1]^T \in \mathbb{R}^N$
$\mathbf{0}_N$	all zero vector $[0, 0, \cdots, 1]^T \in \mathbb{R}^N$
$\mathbf{0}_{N\times M}$	all zero matrix $\in \mathbb{R}^{N\times M}$
$Lap(0, b)^N$	N-dimensional Laplace distribution with $i.i.d.$ components, each with probability density function $f(x) = \frac{1}{2b} e^{\frac{-\lvert x\rvert}{b}}$
$\mathbf{N}(0, \Sigma)$	Gaussian distribution with mean 0 and covariance Σ
$\mathcal{Q}(x)$	$\mathcal{Q}-$ function $\left(\triangleq \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{u^2}{2}} du\right)$
$\mathbb{E}$	expectation of a random variable
$\mathbb{P}$	probability of a random variable

CHAPTER 1

INTRODUCTION

The architecture of control systems has evolved significantly over time. Traditionally, all the components of a dynamical system were co-located and shared information seamlessly. However, with the advance in communication, computation and processing technology, systems with distributed components are being proposed widely. The distributed components interact among each other by exchanging information over a communication link/network to achieve a particular objective. Such systems are typically characterized as Networked Control Systems(NCS) or Cyber-physical Systems(CPS), and often consist of multiple agents that interact with each other either implicitly or explicitly. They are becoming increasingly important because the distributed nature of such systems allows implementation of flexible architectures, renders them more autonomy and scalability, and reduces the system complexity and installation costs. As a result, they are being proposed for diverse application including transportation management systems, power grid, large scale monitoring of a geographical area, automated vehicle systems, process industry and so on[1].

The *networked* or *cyber* features of these systems also introduce limitations associated with them. The links between the components have imperfections such as packet drops, data rate limits, delay etc. The cyber component of the system renders it susceptible to potential security attacks and privacy breaches. Moreover, there can be limits on the computational capabilities and complexity of the systems. Thus, there is a need to analyze the effect of these constraints and imperfections on the functionality of the distributed systems and design systems that explicitly take them

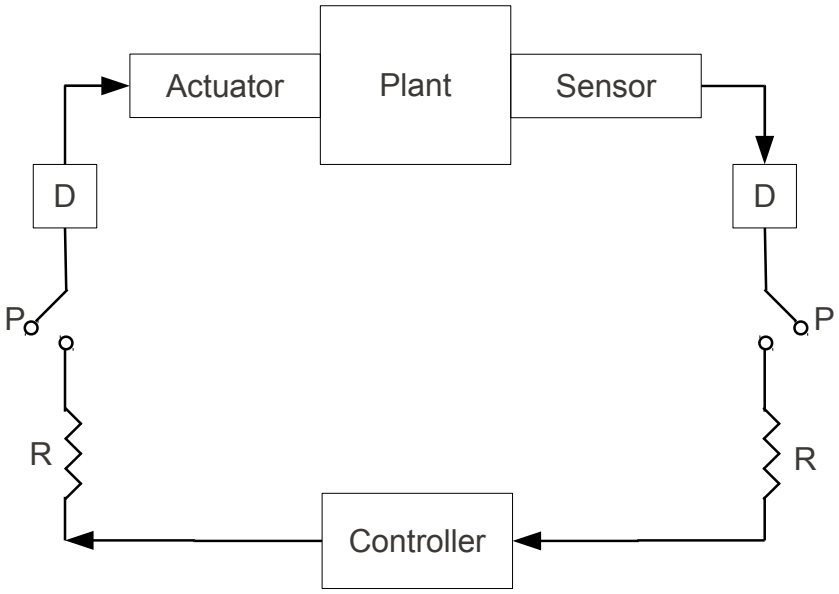


Figure 1.1. A networked control system containing a single feedback loop. D represents delay, P represents packet dropout and R represents rate constraints on respective links.

into account. Early works in this direction analyzed the effects of communication links on simple configurations such as a single loop NCS shown in figure 1.1. There is a vast amount of literature analyzing the effect of packet drops, delay, rate constraints etc. on estimation and control of the NCS [2]. The study of these systems has led to important foundational results in the field.

The NCS have evolved significantly beyond this single loop configuration. A generalized distributed system may have all its components such as plants, sensors, controllers and actuators located at different places. Moreover, there can be interactions between any of these components. A more generalized setup in shown in figure 1.2. The type of interactions between the components may include coupling among multiple plants, multiple sensors observing a single plant, multiple sensors sharing a communication link, selective controllers sharing information among each other etc. Moreover, there can be intruders and attackers present anywhere in the system ar-

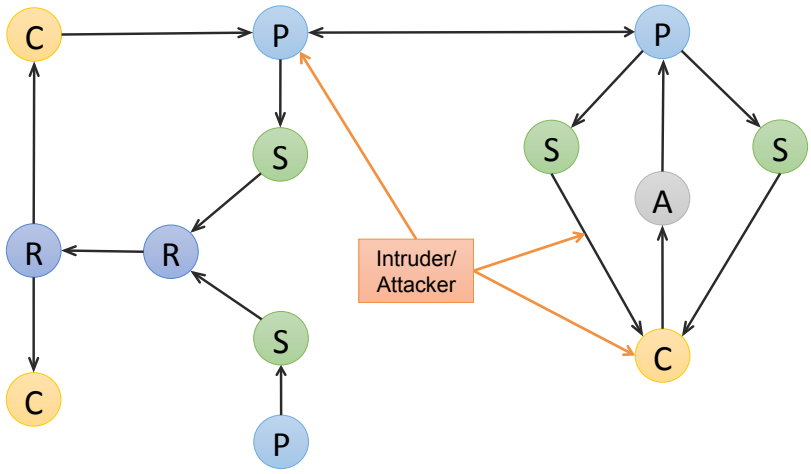


Figure 1.2. A general networked control system containing plants(P), sensors(S), controllers(C), actuators(A) and relays(R). Each link may include rate constraints, packet dropouts and delays. Also, an intruder can snoop on messages between S and C, or hack into P or C.

chitecture, which may try to snoop upon some private parameters or actively attack and disrupt the system.

In this dissertation, we study two problems in this generalized setup that address the design and analysis of multi-agent NCS and CPS in presence of data rate limitations and privacy requirements, respectively.

1.1 Rate Control for Networked Estimation

In chapter 2, we focus on a scenario in which control oriented applications operate over a rate constrained communication network. Due to rate limits on the links, the network may become congested if the applications exchange information at a high data rate, thereby leading to packet drops and information loss. To prevent this, we develop a rate control protocol that regulates the data transmission rates of these applications. This problem lies at an intersection of the fields of control systems and communication networks, which have largely evolved independently. Traditional

rate control protocols such as Transmission Control Protocol (TCP) [3], which is widely used in internet, were not developed for these control oriented applications and therefore, cannot be used in NCS. We develop a systematic procedure to obtain the rate control protocol suitable for such applications.

We consider a distributed estimation setup, wherein a remote estimator tries to estimate the state of a local plant. The plant transmits its state to the remote estimator over a rate limited communication network. Moreover, several plant-estimator pairs share the same communication network. As explained earlier, congestion in the network leads to packet drops and the estimators do not receive perfect information from the plants, thus degrading estimation performance. We propose a probabilistic transmission strategy through which the plants can vary their transmission rates. We characterize the effect of transmission rates and packet drops on the estimation error. Then, using the network utility maximization (NUM) [4] framework we formulate a network optimization problem for finding optimal plant transmission rates that minimize the overall system estimation error. The rate control protocol is then obtained as a distributed solution of this optimization problem.

1.2 Privacy and Cooperation in Multi-agent Cyber-physical Systems

Architectures of modern cyber-physical systems (CPS) include tight coupling between its physical dynamical components, computational and processing units, and communication network. As a result of this integration, the performance of the physical components depends on the correct functioning of the other components. These physical components are interconnected through local communication networks, which in turn may be connected to a global network like internet. This architecture of the CPS is inherently susceptible to security attacks and privacy breaches since there are several access points though which potential intruders may gain unauthorized access into the CPS. Moreover, the intruder may be physically present in the

system. For example, an intruder may be able to monitor the interactions between system components or an component/agent itself can go rouge and try to disrupt the system functionality. Therefore, security and privacy mechanisms need to be an integral part in the design of upcoming cyber-physical systems.

In this dissertation, we focus on the privacy aspect of distributed multi-agent dynamical systems. These agents communicate and coordinate among themselves to achieve a particular objective. Examples include consensus [5], formation control [6], and distributed optimization [7, 8]. Each agent has some associated sensitive parameters that it wishes to keep private. For example, in a consensus network, each (node) agent has edges with associated edge weights. These edge weights represent sensitive information, for example, load values in the case of power networks. In another scenario, the states of the node can represent sensitive information like the position and velocity of the agent. As mentioned above, such systems are susceptible to privacy breaches in which an intruder gains information about these sensitive parameters. In this dissertation, we design privacy mechanisms to prevent such breaches and analyze their effect on the system performance.

In chapter 3, we consider a distributed linear multi-agent dynamical system which is monitored by a control center using measurements from the agents. Each agent has some sensitive parameters associated with its dynamics. In this setup, a potential intruder can try to infer these sensitive parameters by snooping upon the measurements sent by the agents, or it may hack into the control system itself. To prevent this, we design the privacy mechanism using the notion of Differential Privacy (DP) [9, 10]. Under this notion, the agents mask their actual measurements by adding random noise before sending them to the control center. The noise ensures that if the sensitive parameters change within a specified limit, the resulting measurements are *statistically not very different*. Thus, an intruder is unable to determine the actual parameters using the noisy measurements.

In chapter 4, we study another privacy related problem to answer the following question: in the presence of a noisy differential privacy mechanism, is it always beneficial for the agents to cooperate? We consider a setup in which multiple agents cooperatively minimize a quadratic cost as a function of their states by exchanging state information among each other. We design a noise adding DP mechanism to keep their states private. Next, we introduce a method by which the agents can vary their cooperation level. By studying the combined effect of the privacy mechanism and the cooperation level on the system performance, we argue that generally, it is always beneficial for the agents to reduce cooperation if they want to achieve a higher level of privacy.

1.3 Contributions of the Dissertation

This dissertation studies problems related to analysis and design of distributed protocols for NCS and privacy mechanisms for multi-agent systems, respectively. The main contributions with regard to these problems are summarized below.

1. Rate control for Networked Estimation

   - We consider a distributed estimation setup wherein multiple plant(source)-estimator(destination) pairs are connected through a common rate constrained communication network. We propose a probabilistic transmission strategy by which a source can vary its data transmission rate, and analyze its effect on the estimation performance.
   - We use the network utility maximization (NUM) framework to obtain a scalable rate control protocol that optimally allocates the transmission rates to the sources such that the overall estimation performance of the system is maximized. Our method provides a systematic way of designing protocols that are specifically catered towards estimation oriented applications.

- We develop the rate control protocol in primal form and show that under low network congestion, it resembles the structure of the standard TCP protocol. Therefore, our protocol can co-exist with other rate control protocols. This allows estimation oriented applications to be integrated easily into existing communication networks like the internet. We also analyze the stability of our protocol in presence of time invariant delays.

2. Dynamical Privacy in Multi-agent LTI Systems

   - We present a noise adding DP mechanism for protecting the privacy of the parameters related to the dynamics of the agents in a continuous-time linear time invariant system.
   - We obtain an analytical upper bound to the sensitivity of the system which provides a sufficient condition to design the privacy noise.
   - Our framework has a wide range of applications and we present two of these: second-order consensus problem and a LQR control problem in which the DP mechanism protects the topology of the consensus network and the state cost matrix used in the quadratic cost, respectively.
   - Using numerical simulations for the second-order consensus problem, we show that for asymptotically low privacy levels and asymptotically high SNR values, the privacy mechanism has marginal effect on the eigenvalue estimation performed by the control center when compared to the parameter identification error incurred by the intruder. Thus, in this asymptotic regime, the proposed mechanism provides privacy in the system with only marginal performance degradation.

3. Privacy vs Cooperation in Multi-agent Systems

   - We consider a general class of multi-agent systems arising from the solution

of quadratic optimization problems via distributed computation. We propose a noise-adding privacy mechanism for the agents to solve the optimization problem while maintaining privacy of their states over time. We analytically quantify the effect of the privacy noise on the system performance.

- We present a novel method to introduce the notion of *cooperation level* in cooperative multi-agent systems, as a weighting factor by which the agents weigh the system cost vs. their individual costs. We show that, due to the privacy noise, the performance of the distributed system may improve when reducing the cooperation level among the agents. In fact, we simultaneously characterize the effect of cooperation and privacy levels on the system performance, and show that a fundamental tradeoff exists between the two in multi-agent systems.
- We illustrate our results through the problems of consensus and one-dimensional Voronoi tessellation. In both cases, we show (by simulation) that an optimal cooperation level exists to maximize the system performance for a desired privacy level. Our results mathematically support the intuition that the optimal cooperation level should decrease if the privacy level increases.

CHAPTER 2

RATE CONTROL PROTOCOL FOR DISTRIBUTED ESTIMATION

2.1 Background

The architecture and protocols in a communication network should ideally depend on the objectives of the end users. Traditionally, such networks were used with the sole goal of reliable data transfer. More recently, such networks have been proposed to be used in control and estimation applications in the Networked Control Systems (see, e.g., the special issue [11]). In such applications, the performance metric is a complicated function of delay, throughput, and reliability; hence, traditional network protocols may not be suitable. For both the cases when the communication network is designed specifically for estimation or control, and when the communication network is shared with data unrelated to such applications, it is of interest to design network protocols that optimize the performance relevant to these applications.

However, most of the research in Networked Control Systems so far has focused on analyzing and designing a single networked control system in isolation. While this has led to important foundational results, it has ignored the new problems that may arise when multiple such systems operate over a common communication network. As an example, networked communication may give rise to congestion or MAC delays. Such effects will impact the performance of every networked control system and in fact, will couple their performance even though the systems may not be dynamically coupled. It is, thus, of interest to study the impact of communication network protocols on the performance of multiple control systems sharing a common network, and further, design network protocols more suitable for estimation and control ([12], [13]).

In this dissertation, we focus on a rate control protocol suitable for an estimation oriented cost function. We consider multiple systems, each of which consists of an estimator that remotely estimates the state of an associated process. A sensor collocated with each process transmits information over a shared communication network to the estimator. The network has capacity constraints for every link. Such a capacity constrained network may result in congestion when the network load increases. Congestion results in packet losses and delays, which adversely affect the estimation performance. We show that traditional rate control protocols such as TCP may not be suitable for optimizing estimation performance, and propose a new distributed rate control protocol that can co-exist with existing rate control protocols.

The problem of congestion control has been well studied for communication networks (see, e.g., [14]). TCP ([3]) is the most widely used congestion control protocol in the Internet. While originally an engineering heuristic, TCP has now been reverse engineered to show that it is a distributed solution that optimizes a particular utility function ([15]). The chief tool in this regard is the Network Utility Maximization (NUM) framework ([16]) which transforms the end objective to an optimization problem with constraints. The communication protocols are the distributed solutions to these optimization problems ([17]).

The primary aim of traditional TCP is reliable transfer of data, even at the expense of delays. For estimation and control, it may be more useful to have a lower reliability, but a higher throughput. Moreover, not all processes need to transmit data at the same rate to achieve the same estimation error covariance. Thus, issues such as fairness relevant to traditional TCP may not be applicable. In fact, using TCP for estimation purposes may result in instability of the estimation error covariance. Because of these reasons, designing an estimation oriented rate control protocol is not simply a matter of substituting the estimation error covariance as a cost function instead of the throughput. Our proposed protocol, while sharing the

formal structure of TCP protocols, considers these issues directly. The proposed protocol is implemented at the transport layer of the standard OSI layer stack, and thus, preserves the layered structure of the network.

To ensure that the proposed protocol can co-exist with the standard TCP, we use a cost minimization framework that is analogous to the standard NUM framework. The total cost that the rate control protocol aims to minimize includes both an estimation performance cost and a congestion cost. The work closest to ours is that of [18] which presents a bandwidth allocation scheme by using a dual form of NUM problem. However, our solution is in the primal form and is similar to the structure of the standard TCP protocol. Moreover, we present a stochastic transmission scheme as opposed to the deterministic transmission scheme in [18].

We also come up with conditions on network delay and system parameters for which the protocol remains stable. The delays can be time varying in realistic networks. However, we analyze the stability of the system with fixed delays for tractability. Although it is a special case, fixed delay analysis is important and has a rich history for standard TCP ([17, 19–21]).

2.2 Problem Formulation

2.2.1 Network and Process Setting

Consider the problem set up shown in Figure 2.1. Let all the sources form the source set $\mathcal{S}$. With every source $s \in \mathcal{S}$, associate a unique destination d and denote the destination set by $\mathcal{D}$. Let every source be connected to its corresponding destination through a shared capacity constrained network $\mathcal{N}$. We model the network as a graph, wherein the end-nodes are the sources and the destinations, the intermediate nodes are routers that forward packets and the edges correspond to the communication channels in the network. Let $\mathcal{L}$ be the set of links in the network and $L(s)$ be the set

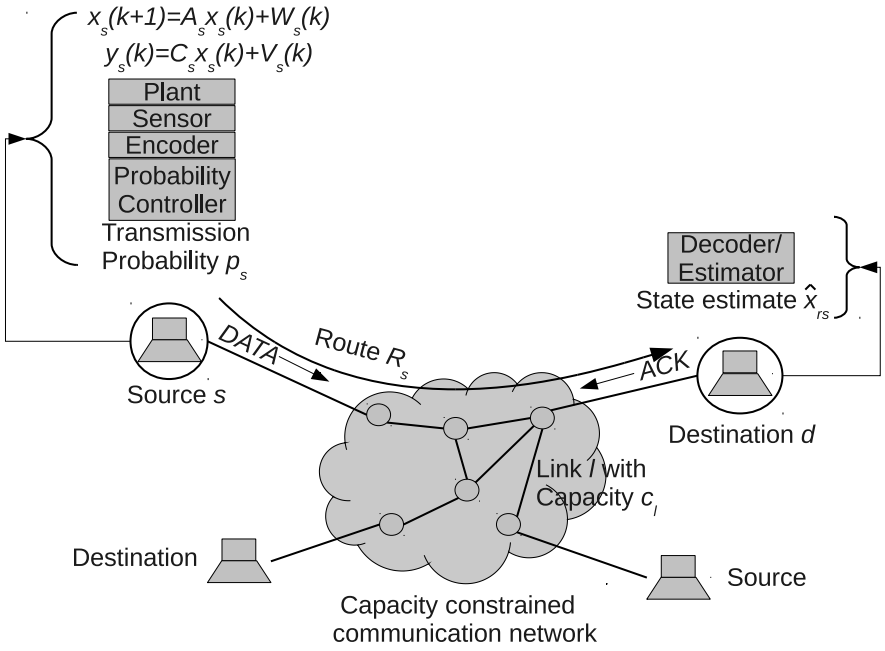


Figure 2.1. System architecture in which multiple processes are remotely estimated across a shared communication network.

of links that are used by source s to communicate with its corresponding destination d. Further, denote the route between source s and destination d by R_s. Each link $l \in \mathcal{L}$ has a limited capacity c_l in terms of "packets per time slot" on an average. Any individual link may be shared by one or more sources.

Each source s comprises of a process P_s, a sensor SR_s, an encoder ENC_s, and a rate controller PC_s. The process P_s evolves according to the discrete-time linear model

$$P_s : \qquad x_s(k+1) = A_s x_s(k) + W_s(k), \qquad k \geq 0 \tag{2.1}$$

where $x_s(k) \in \mathbb{R}^{n_s}$ is the process state and $W_s(k)$ is the process noise. The initial condition $x_s(0)$ and the white process noise $W_s(k)$ are assumed to be Gaussian with zero mean and variance $X_s > 0$ and $Q_s > 0$, respectively. The output of the process P_s is sensed by the sensor SR_s which generates noisy measurements according to the

relation

$$SR_s : \quad y_s(k) = C_s x_s(k) + V_s(k), \qquad k \geq 0 \tag{2.2}$$

where $y_s(k) \in \mathbb{R}^{m_s}$ is the process output, $V_s(k)$ is the measurement noise that is assumed to be white, Gaussian with zero mean and variance $\Sigma_s > 0$. The initial state and the noises $\{x_s(0), W_s(k), V_s(k)\}$ are assumed to be mutually independent $\forall\ s \in \mathcal{S}$ and $\forall\ k$. Further, these random variables are assumed to be mutually independent among all sources. Finally, we assume that each pair (A_s, C_s) is observable.

2.2.2 Encoding and Decoding Scheme

The encoder ENC_s uses the noisy measurements to generate transmission data and sends it to its corresponding destination using constant size packets. The packet size is assumed to be large enough to represent a real number with negligible quantization error. The data from ENC_s is received at the corresponding destination possibly with a stochastic delay τ_{sd} which models the transmission delay. Each destination comprises of a decoder DEC_d, that uses the received data to generate a state estimate that is optimal in the minimum mean squared error (MMSE) sense. We ignore any queuing delays in the network and assume the existence of a time stamp for every transmitted packet. When a destination receives a packet, it sends back an acknowledgment (ACK) to the corresponding source. We assume that $ACKs$ are never lost in the network.

We employ the encoder and decoder scheme described in [22]. At source s, denote the local estimate of state $x_s(k)$ given the measurements $\{y_s(j)\}_{j=1}^{k}$ by $\hat{x}_s(k)$. Further, denote the remote state estimate, produced by DEC_d at the corresponding destination d, by $\hat{x}_{rs}(k)$. The encoder and the decoder are given by

- ENC_s:
  - At each time slot k, calculate $\hat{x}_s(k)$ using (say) a Kalman Filter.

– Transmit $\hat{x}_s(k)$ along with the time stamp k.

- DEC_d:
  - If $k = 0$, set the stored time stamp $t_d = -1$.
  - If DEC_d receives a packet in time slot k, extract the time stamp $k^{'}$ from the packet.
    1. If $k^{'} \leq t_d$, ignore the old packet and set $\hat{x}_{rs}(k) = A_s\hat{x}_{rs}(k-1)$.
    2. If $k^{'} > t_d$, set $\hat{x}_{rs}(k) = A_s^{k-k^{'}}\hat{x}_s(k^{'})$ and set $t_d = k^{'}$.
  - If DEC_d does not receive a packet in time slot k, set $\hat{x}_{rs}(k) = A_s\hat{x}_{rs}(k-1)$.

As discussed in [22], this encoder-decoder structure is optimal amongst all causal structures.

2.2.3 Communication Scheme

We consider a stop-and-wait type communication protocol. In any time slot k, the source s transmits the local estimate $\hat{x}_s(k)$ to the corresponding destination d. The transmission is stochastic with the transmission probability $p_s(k)$ at time slot k. The transmission events at different time slots are assumed to be independent. The transmission probability $p_s(k)$ can be viewed as the effective transmit rate of the source s in terms of "packets per time slot" on average. Hence, the rate controller PC_s is implemented as a probability controller, which controls the source rate. We use the term 'rate' and 'transmission probability' interchangeably.

As the total rate on a link approaches the link capacity, congestion in the link increases, which may result in packets being dropped by the routers in the network. Let the packet drop probability on a link $l \in \mathcal{L}$ at time slot k be denoted by $d_l(k)$. The drop probability on a link depends on both the link capacity and the total rate on the link. As the total rate approaches link capacity, the queue in the corresponding

router becomes full. In such a situation, all the packets are dropped by the router with a probability approaching 1. To avoid such instances, the routers use queue management protocols such as Random Early Detection (RED) protocol ([23]). In RED, the routers increase the drop probability as the queue size increases. The packet drops serve as a feedback mechanism to rate control protocols such as TCP, which reduces the source rate in response to congestion. In standard RED protocol, the link drop probability is a pre-specified increasing and convex function of the total rate (assuming, say, a M/M/1 queue model).

Let $d_s(k)$ be the probability that a packet is dropped by the network on route R_s at time slot k. The packet drop events on route R_s at different time slots are assumed to be independent. Further, the packet drop and packet transmission events on route R_s are assumed to be independent for every time slot. Using the standard assumption (see e.g., [24]) that the drop events on various links are independent, the drop probability $d_s(k)$ on route R_s as observed by the destination d can be expressed as

$$d_s(k) = 1 - \prod_{l \in L(s)} (1 - d_l(k - \tau_{ld})), \tag{2.3}$$

where τ_{ld} denotes the forward delay between link l and destination d. Thus, $d_s(k)$ depends on the rates of the sources that share the links with source s. This introduces a coupling to the problem. Note that d_s may not be a convex function of the source rates.

Remark 2.2.1. ***(Stochastic rate control)*** The stochastic transmission scheme that we propose controls the source rate by varying the transmission probability. This is in contrast to deterministic schemes, wherein the sources send the information at deterministic instants and rate control is achieved by varying the time interval between the transmissions. A stochastic transmission scheme is a natural choice since a congested network drops packets stochastically. Therefore, the information

reception process is inherently probabilistic. We superimpose an additional stochastic transmission process on the stochastic network, still resulting in a stochastic reception process. □

Remark 2.2.2. ***(Instantaneous behavior)*** Due to the stochastic rate control, there may be instants when many sources may not transmit resulting in instantaneous network underutilization or, many sources may transmit at the same time resulting in instantaneous increase in congestion. However, due to the feedback implicit in rate control, such instants will be few and on an average, the network will be utilized in an optimal manner. □

2.2.4 Performance Metric and Problem Statement

For the source s and its corresponding destination d, denote the estimation error covariances of the local estimate $\hat{x}_s(k)$ and the remote estimate $\hat{x}_{rs}(k)$ by $M_s(k)$ and $F_s(k)$, respectively. Since the pair (A_s, C_s) is observable, the local estimation error covariance $M_s(k)$ converges to a steady state value, denoted by M_s with a slight abuse of notation. According to the decoder structure DEC_s, the remote estimation error covariance $F_s(k)$ evolves as

$$F_s(k) = \begin{cases} A_s^{\tau_{sd}} M_s(k-\tau_{sd}) A_s^{\tau_{sd},T} + \sum_{i=0}^{\tau_{sd}-1} A_s^i Q_s A_s^{i,T} & \text{if a packet is received,} \\ A_s F_s(k-1) A_s^T + Q_s, & \text{otherwise,} \end{cases}$$

where $A^{\tau,T} \triangleq (A^\tau)^T$. Thus, $F_s(k)$ is a random variable. As a performance metric, we consider its expected value that evolves as

$$\mathbb{E}[F_s(k)] = \mathbb{E}\Big[p_s(k-\tau_{sd})(1-d_s(k))\Big(A_s^{\tau_{sd}} M_s(k-\tau_{sd}) A_s^{\tau_{sd},T} + \sum_{i=0}^{\tau_{sd}-1} A_s^i Q_s A_s^{i,T}\Big) + (1 - p_s(k-\tau_{sd})(1-d_s(k)))\Big(A_s F_s(k-1) A_s^T + Q_s\Big)\Big] \quad (2.4)$$

where $p_s(k-\tau_{sd})(1-d_s(k))$ is the packet reception probability and the expectation is taken with respect to the packet transmission process, packet drop process and delays in the network. Under the assumption that the system reaches a steady state, (2.4) can be written as

$$F_s(p_s,d_s) = p_s(1-d_s)\Big(\mathbb{E}\Big[A_s^{\tau_{sd}} M_s A_s^{\tau_{sd},T} + \sum_{i=0}^{\tau_{sd}-1} A_s^i Q_s A_s^{i,T}\Big]\Big) \\ + (1-p_s(1-d_s))\Big(A_s F_s(p_s,d_s) A_s^T + Q_s\Big), \tag{2.5}$$

where p_s, d_s and $F_s(p_s,d_s)$ denote the steady state values of $p_s(k)$ and $d_s(k)$ and $\mathbb{E}[F_s(k)]$, respectively.

Problem statement: Let $\mathbf{p}$ denote the vector of all steady state transmission probabilities, i.e. $\mathbf{p} = (p_1, p_2, \cdots, p_{|\mathcal{S}|})^T$, where $|\mathcal{S}|$ denotes the cardinality of set $\mathcal{S}$. We consider the estimation cost incurred for the source s as $c_s = tr(F_s(p_s,d_s))$, where tr denotes the trace. Further, the total cost of the system $C_{sys}(\mathbf{p})$ is chosen to be the sum of individual costs. For ease of notation, we will denote $\{tr(F_s(p_s,d_s)), tr(M_s), tr(A_s A_s^T), tr(Q_s)\}$ by $\{f_s(p_s,d_s), m_s, a_s, q_s\}$, respectively. Thus,

$$C_{sys}(\mathbf{p}) = \sum_{s\in\mathcal{S}} f_s(p_s,d_s) \quad \text{where,}$$
$$f_s(p_s,d_s) = p_s(1-d_s) tr\Big(\mathbb{E}\Big[A_s^{\tau_{sd}} M_s A_s^{\tau_{sd},T} + \sum_{i=0}^{\tau_{sd}-1} A_s^i Q_s A_s^{i,T}\Big]\Big) \\ + (1-p_s(1-d_s))(tr(A_s F_s(p_s,d_s) A_s^T) + q_s). \tag{2.6}$$

The problem is to find the optimal value of $\mathbf{p}$ which minimizes the cost function $C_{sys}(\mathbf{p})$ under the rate constraints. This problem can also be viewed as a resource (rate) allocation problem with an objective to minimize a system cost. We are particularly interested in decentralized solutions that ensure that the solution is scalable for large networks.

2.3 Rate Control Protocols

2.3.1 Bounds on Performance Metric

The following upper and lower bounds for the cost follow from algebraic manipulations on (2.6).

Lemma 2.3.1. ***(Cost function)*** *The steady state value $f_s(p_s, d_s)$ satisfies $f_s^l(p_s, d_s) < f_s(p_s, d_s) < f_s^u(p_s, d_s)$, where*

$$f_s^u(p_s, d_s) \triangleq \frac{p_s(1-d_s)m_s^u + (1-p_s(1-d_s))q_s}{1 - a_{s,max}(1-p_s(1-d_s))}, \tag{2.7}$$

$$f_s^l(p_s, d_s) \triangleq \frac{p_s(1-d_s)m_s^l + (1-p_s(1-d_s))q_s}{1 - a_{s,min}(1-p_s(1-d_s))}, \tag{2.8}$$

$$m_s^u = \begin{cases} (m_s + \frac{q_s}{a_{s,max}-1})\mathbb{E}[a_{s,max}^{\tau_{sd}}] - \frac{q_s}{a_{s,max}-1} & \text{if} \quad a_{s,max} \neq 1, \\ m_s + q_s\mathbb{E}[\tau_{sd}] & \text{otherwise,} \end{cases}$$

$$m_s^l = \begin{cases} (m_s + \frac{q_s}{a_{s,min}-1})\mathbb{E}[a_{s,min}^{\tau_{sd}}] - \frac{q_s}{a_{s,min}-1} & \text{if} \quad a_{s,min} \neq 1, \\ m_s + q_s\mathbb{E}[\tau_{sd}] & \text{otherwise,} \end{cases}$$

where $\lambda(A)$ denotes the eigenvalues of A, $a_{s,max} = \lambda_{max}(A_s A_s^T)$ and $a_{s,min} = \lambda_{min}(A_s A_s^T)$.

Proof. From (2.6), we have,

$$\begin{aligned} f_s(p_s, d_s) &= p_s(1-d_s)\Big(\mathbb{E}\Big[tr(A_s^{\tau_{sd},T} A_s^{\tau_{sd}} M_s) + \sum_{i=0}^{\tau_{sd}-1} tr(A_s^{i,T} A_s^i Q_s)\Big]\Big) \\ &\quad + (1-p_s(1-d_s))(tr(A_s^T A_s F_s(p_s, d_s)) + q_s) \\ &\leq p_s(1-d_s)\Big(\mathbb{E}\Big[m_s a_{s,max}^{\tau_{sd}} + q_s \sum_{i=0}^{\tau_{sd}-1} a_{s,max}^i\Big]\Big) + (1-p_s(1-d_s))(a_{s,max} f_s(p_s, d_s) + q_s), \end{aligned}$$

where we have used the following trace identities:

1. $tr(ABC) = tr(CAB)$,

2. $tr(\mathbb{E}[X]) = \mathbb{E}[tr(X)]$, and

3. $tr(M)\lambda_{min}^k(AA^T) \leq tr(M)\lambda_{min}(A^kA^{k,T}) \leq tr(A^kA^{k,T}M) \leq tr(M)\lambda_{max}(A^kA^{k,T})$ $\leq tr(M)\lambda_{max}^k(AA^T)$, for any positive semi-definite matrix M.

Simplifying and rearranging the last inequality, we get the desired upper bound. The lower bound can be obtained in a similar way, thus completing the proof. ■

In particular, for scalar processes, the upper and lower bounds in (2.7) and (2.8) are satisfied with equality. For analytical tractability, we replace f_s by f_s^u in the system cost. Thus, we approximate $C_{sys}(\mathbf{p}) \approx C(\mathbf{p}) \triangleq \sum_{s\in\mathcal{S}} f_s^u(p_s, d_s)$, where p_s is the transmission probability allotted to source s under the vector $\mathbf{p}$.

Lemma 2.3.2. ***(Convergence)*** *A sufficient condition for the convergence of* $\mathbb{E}[F_s(k)]$ *as (2.4) evolves is given by*

$$p_s(1-d_s) \geq \left(1 - \frac{1}{\rho^2(A_s)}\right)^+ \triangleq p_s^{min}, \tag{2.9}$$

$$\mathbb{E}[a_{s,max}^{\tau_{sd}}] < \infty \qquad \textit{if} \quad a_{s,max} \neq 1, \tag{2.10}$$

$$\mathbb{E}[\tau_{sd}] < \infty \qquad \textit{otherwise},$$

where $\rho(X)$ *denotes the spectral radius of matrix* X*.*

Proof. See [22] for condition (2.9). Condition (2.10) can be obtained from (2.7) in a straightforward manner. ■

Thus, we have the following constrained optimization problem

$$
\begin{aligned}
SYSTEM: \qquad & \min_{\mathbf{p}} \sum_{s \in \mathcal{S}} f_s^u(p_s, d_s(\mathbf{p})), \\
& \text{s.t.} \sum_{s: l \in R_s} p_s \leq c_l \qquad \forall\, l \in \mathcal{L}, \\
& p_s(1 - d_s(\mathbf{p})) \geq p_s^{min} \qquad \forall\, s \in \mathcal{S}, \\
& 0 \leq p_s \leq 1 \qquad \forall\, s \in \mathcal{S},
\end{aligned}
$$

where the notation $d_s(\mathbf{p})$ denotes the explicit relation between the drop probability and transmission probabilities. Assuming that a feasible region exists, we can use standard optimization techniques to obtain a globally optimal solution. However, this approach is not desirable for many reasons:

1. If the drop probability d_s is not a convex function of $\mathbf{p}$, then the system cost $C(\mathbf{p})$ may not be convex, thus making the problem difficult.

2. The method is not scalable since each source requires information about the transmission probabilities and process parameters of all the other sources.

3. It requires the functional relation between $\{d_s : s \in (S)\}$ and $\{p_s : s \in (S)\}$, which may be unavailable in a practical scenario.

We now proceed to transform the problem into a convex form and obtain a distributed solution.

2.3.2 Posing the Problem in the NUM Framework

To obtain a scalable and distributed solution, we employ a network cost minimization framework that is analogous to the primal formulation of the Network Utility Maximization framework ([4]).

Remark 2.3.3. ***(Advantage of the primal form)*** Since the communication network may also be used for data unrelated to estimation / control, the dynamics of the distributed solution should be at the sources and not at the links. This is important especially in heterogeneous networks, where different sources may have different interpretations of link prices. Thus, a single link price controller may not be suitable for all the sources. The primal solution requires changes to the standard TCP only at sources and not in the network. Thus, our solution is practically useful since implementation of the rate controllers needs to be done only at the source node, which is aware of estimation application. □

The NUM framework imposes some requirements on the costs. The costs should be separable among the sources. In other words, the cost associated with source s should depend only on the resource p_s. Moreover, the cost should be positive, monotonically decreasing and convex. However, the costs $\{f_s^u(p_s, d_s) : s \in \mathcal{S}\}$ in (2.7) are coupled among each other through the drop probabilities d_s and hence are neither separable nor convex. Therefore, we eliminate d_s from the costs and let this modified separable cost be denoted by $f_s^u(p_s, 0)$. To include the effect of the drop probabilities, we define a barrier of the form $B_l\left(\sum_{s:l\in R_s} p_s\right)$ corresponding to each link l, and add it to the total cost. The barrier maps the congestion level in the link to an additive cost to the system. Thus, we obtain the following relaxation of the $SYSTEM$ problem

$$
\begin{aligned}
USER: \qquad & \min_{\mathbf{p}} \sum_{s\in\mathcal{S}} f_s^u(p_s, 0) + \sum_{l\in\mathcal{L}} B_l\left(\sum_{s:l\in R_s} p_s\right), \\
& \text{s.t.} \sum_{s:l\in R_s} p_s \le c_l \qquad \forall\, l \in \mathcal{L}, && (2.11) \\
& p_s \ge p_s^{min} \ge 0 \qquad \forall\, s \in \mathcal{S}, && (2.12) \\
& p_s \le 1 \qquad \forall\, s \in \mathcal{S}. && (2.13)
\end{aligned}
$$

The choice of the barrier function requires some care. It should be a monotonically

increasing function of the total rate on a link. This ensures that as the congestion increases, the total system cost also increases. Thus, congestion control can be achieved by minimizing the system cost. By ensuring a steep increase in the barrier function as the rates approach capacity of the links, the capacity constraints can be explicitly incorporated in the system cost. Once we have satisfied the separability requirement, we can prove that the cost used in the $USER$ problem satisfies the remaining constraints. There are two terms in the cost function, that we consider one by one.

Lemma 2.3.4. ***(Properties of cost function)*** *The cost function $f_s^u(p_s, 0)$ is positive, monotonically decreasing and convex for $p_s > 1 - \frac{1}{a_{s,max}}$.*

Proof. The proof follows by differentiating $f_s^u(p_s, 0) = \frac{p_s m_s^u + (1-p_s)q_s}{1 - a_{s,max}(1-p_s)}$ twice and verifying that the terms in numerator and denominator are of appropriate signs. ■

To ensure the convexity of the barrier function, we assume that B_l is differentiable and define it as

$$B_l\left(\sum_{s:l\in R_s} p_s\right) \triangleq \int_0^{\sum_{s:l\in R_s} p_s} t_l(x)dx, \tag{2.14}$$

where t_l is the penalty function corresponding to link l. If t_l is a monotonically increasing function of the total rate on the link l, then B_l is convex. We will ensure this by choosing an appropriate penalty function in (2.17). Finally, we have the following result.

Lemma 2.3.5. ***(Incorporating constraints into cost)*** *The cost used in the problem $USER$ implicitly guarantees the constraints (2.11) and (2.12).*

Proof. The cost $f_s^u(p_s, 0)$ is positive and finite iff $p_s > 1 - \frac{1}{a_{s,max}}$. Since $a_{s,max} = \lambda_{max}(A_s A_s^T) \geq \rho^2(A_s)$, $f_s^u(p_s, 0)$ is positive and finite only for $p_s > 1 - \frac{1}{a_{s,max}} \geq 1 - \frac{1}{\rho^2(A_s)}$. Thus, the cost $f_s^u(p_s, 0)$ becomes infinite when p_s approaches p_s^{min}. Further, the

barrier function B_l on link l rapidly increases as the total rate on the link approaches the link capacity, thereby increasing the cost function. Thus, both (2.11) and (2.12) are satisfied. ■

2.3.3 Solution of the Optimization Problem

We have shown that if we choose the penalty function appropriately, then the total system cost in the $USER$ problem is positive and convex. Moreover, the problem constraints are implicitly included in the system cost. Thus, a gradient descent algorithm can be used to minimize the total system cost. We propose a rate controller of the form

$$PC_s : \qquad p_s(k+1) = p_s(k) - k_s \left(\frac{d}{dp_s} f_s^u(p_s, 0) + \sum_{l:l\in L(s)} t_l \left(\sum_{s:l\in R_s} p_s \right) \right), \quad (2.15)$$

where $k_s > 0$ is a sufficiently small step size. The quantity

$$q_{R_s} \triangleq \sum_{l:l\in L(s)} t_l \left(\sum_{s:l\in R_s} p_s \right)$$

can be viewed as the price of using the route R_s, which is the aggregate of prices of all the links on the route.

Remark 2.3.6. ***(Scalability)*** The proposed rate control protocol is scalable to large networks. The values of process parameters and transmission probabilities of other sources are not required to implement the algorithm. The only information that a source needs is the route price. This can be provided implicitly or explicitly by the network through $ACKs$ from the destination to the source. □

Besides being monotone increasing in the rates, the penalty functions t_l should be chosen such that the problem $USER$ closely approximates the problem $SYSTEM$. We observe here that the congestion in the network affects the system performance

through the drop probabilities. Since drop probabilities have a direct effect on the system performance, we choose a penalty function that depends on the drop probabilities. In turn, since the drop probability d_l on a link l depends on the total rate on the link, the penalty function also depends on the total rate on the link, as required by the optimization framework. In particular, we choose

$$t_l\left(\sum_{s:l\in R_s} p_s\right) = -\log\left(1 - d_l\left(\sum_{s:l\in R_s} p_s\right)\right). \tag{2.16}$$

Note that t_l is positive and monotonically increases to infinity as the total rate on the corresponding link approaches its capacity; thus the barrier function is indeed convex as required and can be explicitly written as

$$B_l\left(\sum_{s:l\in R_s} p_s\right) = \int_0^{\sum_{s:l\in R_s} p_s} -\log(1 - d_l(x))dx. \tag{2.17}$$

Also, the route price is given by

$$q_{R_s} = \sum_{l:l\in L(s)} -\log(1 - d_l) = -\log\left(\prod_{l:l\in L(s)} (1 - d_l)\right) = -\log(1 - d_s).$$

Remark 2.3.7. ***(Estimating the route price)*** The advantage of choosing a logarithmic penalty function is apparent from the preceding calculation. To calculate the route price, the probability controllers PC_s require only the route drop probability d_s. They do not require the prices of individual links along the route. Thus, no explicit field in the $ACKs$ is required to collect price information from the links. The route drop probability can be estimated merely based on whether $ACKs$ are received or not. □

Note that the different choices of the penalty/barrier function may change the way in which congestion control is handled. For example, in a conservative approach,

the barrier may be high for low link rates. We do not claim that the particular choice we have proposed provides the best performance in all the cases. Other choices may be beneficial depending on the system and application.

The barrier B_l is the integral of a logarithmic function between the interval [0,1]. Therefore, it does not diverge as the congestion increases. Ideally, when the network congestion is large, the barrier should be large as compared to the estimation cost. Thus, we scale down the cost $f_s^u(p_s, 0)$ (analogous to increasing the barrier function) by a constant β_s to satisfy this property. We choose $\beta_s = N_s(q_s - m_s^u(1 - a_{s,max}))$, where N_s is a large positive constant. The constant β_s is large when the process is more unstable or the process and measurement noises and delays are large. Thus, it acts like a normalization factor to the estimation error covariance. With this relaxation, the optimization problem becomes

$$USER: \qquad \min_{\mathbf{p}} \sum_{s\in\mathcal{S}} \frac{1}{\beta_s} f_s^u(p_s, 0) + N_s \sum_{l\in\mathcal{L}} B_l \left(\sum_{s:l\in R_s} p_s \right),$$

and the probability controller becomes

$$PC_s : p_s(k+1) = p_s(k) + k_s^{'} \left(\frac{1}{(1 - a_{s,max}(1 - p_s(k)))^2} + N_s \log(1 - d_s(k)) \right), \quad (2.18)$$

where $k_s^{'} = \frac{k_s}{N_s}$.

The probability controller structure in (2.18) can be implemented using a TCP-like structure under low network congestion conditions. In this regime, the route drop probabilities are also low, $\{d_s \ll 1, s \in \mathcal{S}\}$ which implies that $-\log(1 - d_s) \approx d_s$. Thus, (2.18) becomes

$$PC_s : \qquad p_s(k+1) = p_s(k) + k_s^{'} \left(\frac{1}{(1 - a_{s,max}(1 - p_s(k)))^2} - N_s d_s(k) \right). \quad (2.19)$$

Consider the following TCP-like probability controller, denoted by PC_s^{TCP}:

- If a packet is not transmitted in time slot k, then set $p_s(k+1) = p_s(k)$.
- If a packet is transmitted and ACK is received, then set $p_s(k+1) = p_s(k) + k_s^{'}$.
- If a packet is transmitted and ACK is not received, then set $p_s(k+1) = p_s(k) - k_s^{'}(N_s(1 - a_{s,max}(1 - p_s(k)))^2 - 1)$.

Proposition 2.3.8. ***(TCP-like probability controller)*** *The mean rate achieved by the TCP-like probability controller PC_s^{TCP} is upper bounded by the steady state rate of probability controller PC_s in (2.19).*

Proof. The mean rate achieved by the TCP-like probability controller PC_s^{TCP} (where the expectation is taken with respect to transmission and drop processes) is given by

$$\begin{aligned}
\mathbb{E}[p_s(k+1)] &= (1 - p_s(k))\mathbb{E}[p_s(k)] + p_s(k)(1 - d_s(k))\mathbb{E}\left[p_s(k) + k_s^{'}\right] \\
&\quad + p_s(k)d_s(k)\mathbb{E}\Big[p_s(k) - k_s^{'}\left(N_s(1 - a_{s,max}(1 - p_s(k)))^2 - 1\right)\Big], \\
&= \mathbb{E}[p_s(k)] + \lambda(k)k_s^{'}\left(\frac{1}{\mathbb{E}\left[(1-a_{s,max}(1-p_s(k)))^2\right]} - N_s d_s(k)\right) \\
&\leq \mathbb{E}[p_s(k)] + \lambda(k)k_s^{'}\left(\frac{1}{\left(1-a_{s,max}(1-\mathbb{E}[p_s(k)])\right)^2} - N_s d_s(k)\right),
\end{aligned}$$

where $\lambda(k) = p_s(k)\mathbb{E}\left[(1-a_{s,max}(1-p_s(k)))^2\right]$. The mean rate obtained by PC_s^{TCP} is thus upper-bounded by a probability controller similar to that in (2.19), except the scaling factor $\lambda(k) > 0$. ■

The modified probability controller is similar in structure to the standard TCP controller, which also regulates the rate based on the received $ACKs$. For rate control, the TCP controller changes the window size whereas the proposed probability controller changes transmission probabilities. Thus, the proposed controller can be easily implemented in current networks due to its resemblance to the TCP controller. A key difference between the two rate controllers is that TCP involves retransmissions as opposed to no retransmissions in the proposed protocol. This can be attributed

to the different end-objectives, i.e. reliability for TCP and estimation performance for the proposed probability controller. Nevertheless, both protocols solve an overall network optimization problem in a distributed manner.

2.4 Stability Analysis with Network Delays

We now consider the effect of network delays on the stability of the proposed probability controllers. For tractability, we assume that the delays are constant. Let the delay in the forward direction between source s and link l be denoted by τ_{sl}^f. Further, let the delay in backward direction between link l and source s via the corresponding destination d be denoted by τ_{sl}^b. Both the forward and backward delays are assumed to be positive integers. We assume that the total round trip time w.r.t. link l is constant for every link in the route, i.e. $\tau_s = \tau_{sl}^f + \tau_{sl}^b \quad \forall\, l \in \mathcal{L}$. Let $R = [r_{ij}]$ denote the $|\mathcal{L}| \times |\mathcal{S}|$ routing matrix, where

$$r_{ij} = \begin{cases} 1 & \text{if source } j \text{ uses link } i \ (i \in R_j) \\ 0 & \text{otherwise.} \end{cases}$$

Further, let y_l denote the aggregate rate on link l

$$y_l(k) = \sum_{s:l \in R_s} p_s(k - \tau_{sl}^f) = \sum_s r_{ls} p_s(k - \tau_{sl}^f). \tag{2.20}$$

The penalty function t_l at link l depends on y_l through the relation

$$t_l(k) = h_l(y_l(k)), \tag{2.21}$$

where t_l is the penalty function as denoted in (2.14) and h_l is a positive non-decreasing function. The route price q_s associated with route R_s can be written

as

$$q_s(k) = \sum_{l:l\in L(s)} t_l(k - \tau_{sl}^b) = \sum_l r_{ls} t_l(k - \tau_{sl}^b). \tag{2.22}$$

At the source s, probability controller updates $p_s(k)$ using the relation

$$p_s(k+1) = g_s(p_s(k), q_s(k)), \tag{2.23}$$

where g_s is the nonlinear function as described in (2.18). The equations (2.20)-(2.23) form a nonlinear feedback system and the presence of delays can make the network unstable. We wish to characterize the local asymptotic stability of the network around the equilibrium point and obtain conditions under which stability is guaranteed. We proceed by linearizing the system of equations around the equilibrium point. Let $\{p_s, y_l, t_l, q_s\}$ denote the equilibrium values for $\{p_s(k), y_l(k), t_l(k), q_s(k)\}$. Further, let $p_s(k) = p_s + \delta p_s(k)$, $y_l(k) = y_l + \delta y_l(k)$, $t_l(k) = t_l + \delta t_l(k)$, $q_s(k) = q_s + \delta q_s(k)$ be small perturbations around the equilibrium point. Linearizing (2.20)-(2.23), we obtain

$$\delta y_l(k) = \sum_s r_{ls} \delta p_s(k - \tau_{sl}^f), \tag{2.24a}$$

$$\delta t_l(k) = h_l^{'}(y_l)\delta y_l(k), \tag{2.24b}$$

$$\delta q_s(k) = \sum_l r_{ls} \delta t_l(k - \tau_{sl}^b), \tag{2.24c}$$

$$\delta p_s(k+1) = \alpha_s \delta p_s(k) + \beta_s \delta q_s(k), \quad \text{where} \tag{2.24d}$$

$$\alpha_s = \frac{\partial}{\partial p} g_s(p,q)|_{p_s,q_s}, \quad \beta_s = \frac{\partial}{\partial q} g_s(p,q)|_{p_s,q_s}.$$

For the source law in (2.18), we have

$$\alpha_s = 1 - \frac{2k_s^{'} a_{s,max}}{[1 - (a_{s,max}(1-p_s))]^3}, \quad \text{and} \quad \beta_s = -k_s^{'} N_s.$$

Denote by $\{y, t, q, p\}$ the vectors of aggregate rates, penalty functions, route prices

and transmission probabilities, respectively. Taking z transform of (2.24) and combining the variables in vector form we obtain

$$\begin{aligned}
&\delta y(z) = R^f \delta p(z), \qquad \delta t(z) = F\delta y(z),\\
&\delta q(z) = R^b \delta t(z), \qquad z\delta p(z) = \alpha\delta p(z) + \beta\delta q(z), \quad \text{where}\\
&R^f(z) \triangleq [r^f_{ij}(z)], \qquad r^f_{ij}(z) = r_{ij} z^{-\tau^f_{ji}},\\
&R^b(z) \triangleq [r^b_{ij}(z)], \qquad r^b_{ij}(z) = r_{ji} z^{-\tau^b_{ij}},\\
&F \triangleq diag\{h_l^{'}(y_l)\}_{l\in\mathcal{L}}, \qquad \alpha \triangleq diag\{\alpha_s\}_{s\in\mathcal{S}}, \qquad \beta \triangleq diag\{\beta_s\}_{s\in\mathcal{S}}.
\end{aligned}$$

Thus, the overall return ratio of the linearized system as seen by the sources becomes

$$\begin{aligned}
T(z) &= (zI-\alpha)^{-1}\beta R^b F R^f = [T_{ij}(z)], \qquad (2.25)\\
T_{ij}(z) &= \beta_i (z-\alpha_i)^{-1} \sum_l r_{li} r_{lj} h_l^{'} z^{-(\tau^f_{jl}+\tau^b_{il})}.
\end{aligned}$$

Theorem 2.4.1. ***(Stability under delays)*** *The system described by equations (2.20)-(2.23) is locally asymptotically stable if the following conditions are satisfied*

$$\begin{aligned}
&k_s^{'} < \min\left\{ \frac{[1-(a_{s,max}(1-p_s))]^3}{a_{s,max}}, \frac{2\sin\left(\frac{\pi}{2(2\tau_s+1)}\right)}{N_s \sum_j \sum_k r_{ki} r_{kj} h_k^{'}(y_k)} \right\} \forall\ s\in\mathcal{S},\ \textit{and}\\
&-1 \notin Co\left(\left\{ 2\sin\left(\frac{\pi}{2(2\tau_s+1)}\right) (e^{j\omega}-\alpha_s)^{-1} e^{-j\omega\tau_s} \right\} \right),
\end{aligned}$$

where Co denotes the convex hull.

Proof. Denote the spectrum of a square matrix Z by $\sigma(Z)$. Using $R^b(z) = diag\{z^{-\tau_s}\}_{s\in\mathcal{S}} R^{f,T}(z^{-1})$ and the properties of similar and diagonal matrices,

we can show that $\sigma(T(z)) = \sigma(A(z)B(z))$, where,

$$A(z) = diag\left\{\sqrt{\frac{\beta_s}{w_s}}\right\}_{s\in\mathcal{S}} R^{f,T}(z^{-1})FR^{f}(z)diag\left\{\sqrt{\frac{\beta_s}{w_s}}\right\}_{s\in\mathcal{S}},$$
$$B(z) = diag\{w_s(z-\alpha_s)^{-1}z^{-\tau_s}\}_{s\in\mathcal{S}}, \quad \text{and} \quad w_s = 2\sin\left(\frac{\pi}{2(2\tau_s+1)}\right) > 0.$$

Assuming that the system is open loop stable and using the generalized Nyquist stability criterion ([25]), the system is stable if the eigenloci of $T(z=e^{j\omega}), \quad \omega\in[0,\pi]$ do not cross the real axis to the left of -1. For open loop stability we should have $|\alpha_s|<1$, which (since $\alpha_s<1$) is equivalent to

$$k_s < \frac{[1-(a_{s,max}(1-p_s))]^3}{a_{s,max}}. \tag{2.26}$$

Assume that λ is an eigenvalue and v is the corresponding normalized eigenvector of $A(e^{j\omega})B(e^{j\omega})$. Then, $\lambda v = A(e^{j\omega})B(e^{j\omega})v$ or $\lambda = v^*A(e^{j\omega})B(e^{j\omega})v$, where v^* denotes the conjugate transpose of v. Since $A(e^{j\omega}) = A^T(e^{-j\omega}) > 0$, we have ([21])

$$\lambda \subset \rho(A(e^{j\omega}))\; Co\left(\left\{w_s(e^{j\omega}-\alpha_s)^{-1}e^{-j\omega\tau_s}\right\}\right), \tag{2.27}$$

where ρ is the spectral radius. Since the spectral radius is upper bounded by the maximum absolute row sum, we have

$$\begin{aligned}\rho(A(e^{j\omega})) &\le \max_{s\in\mathcal{S}}\sum_j\left\|\sum_k r_{ki}r_{kj}h_k^{'}(y_k)(\frac{\beta_s}{w_s})e^{-j\omega(\tau_{jk}^f-\tau_{ik}^f)}\right\| \\ &\le \max_{s\in\mathcal{S}}\frac{k_s^{'}N_s}{w_s}\sum_j\sum_k r_{ki}r_{kj}h_k^{'}(y_k) \overset{(a)}{\le} 1,\end{aligned} \tag{2.28}$$

where (a) follows from the theorem statement. The result follows from (2.26)-(2.28).

■

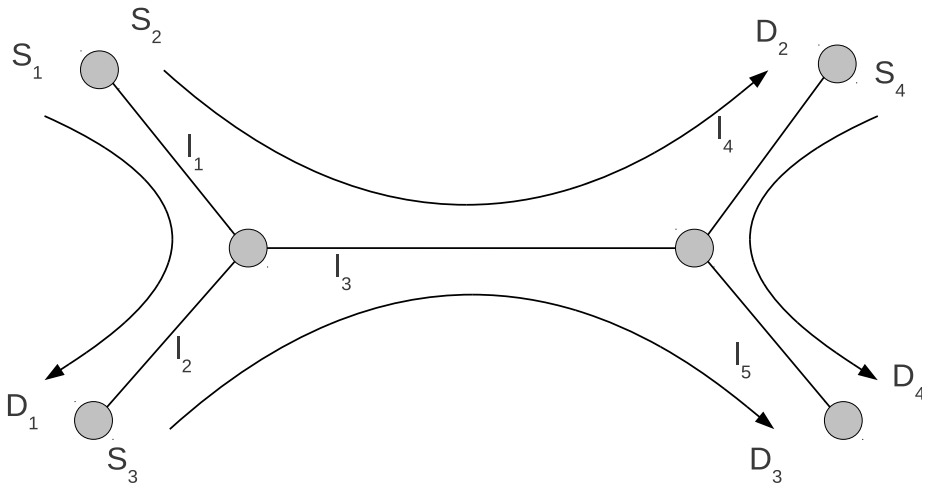


Figure 2.2. The network model used for simulations.

2.5 Simulation Results

Simulations were performed in Matlab to test the protocol performance. Consider the network shown in Fig. 2.2. There are four source destination pairs and five links in the network. Vector processes evolve at sources S_1 and S_2 and scalar processes evolve at sources S_3 and S_4, which are given as follows

$\{A_1, C_1, Q_1, R_1\} = \left\{ \begin{bmatrix} 0.5 & 0.6 \\ 1.1 & 0.1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, 3 \right\}$,

$\{A_2, C_2, Q_2, R_2\} = \left\{ \begin{bmatrix} 1 & 0.5 \\ 0.7 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \end{bmatrix}, \begin{bmatrix} 2.5 & 0 \\ 0 & 1.5 \end{bmatrix}, 2 \right\}$,

$\{A_3, C_3, Q_3, R_3\} = \{1.2, 1, 3.5, 3\}$, and $\{A_4, C_4, Q_4, R_4\} = \{1.1, 1, 2.5, 1.5\}$.

The link capacities are $\{c_1, c_2, c_3, c_4, c_5\} = \{1.5, 1.6, 1.8, 1.7, 1.4\}$, the step size $k_s' = 0.001$ and $N_s = 100$. The delays on the links are $\{d_1, d_2, d_3, d_4, d_5\} = \{1, 2, 2, 3, 4\}$.

For simulating the packet drops, we use a crude form of the standard RED protocol. Let μ be the link utilization factor, which is the ratio of total rate on an link to the link capacity. In RED protocol, the drop probability on a link is a linear function of the queue size, which depends on the link utilization factor. We assume a M/M/1 queuing model to calculate the queue size. Let $\{\mu_{min}, \mu_{max}\}$ denote the link utilization extremes and let $\{N_{min}, N_{max}\}$ denote the corresponding queue sizes.

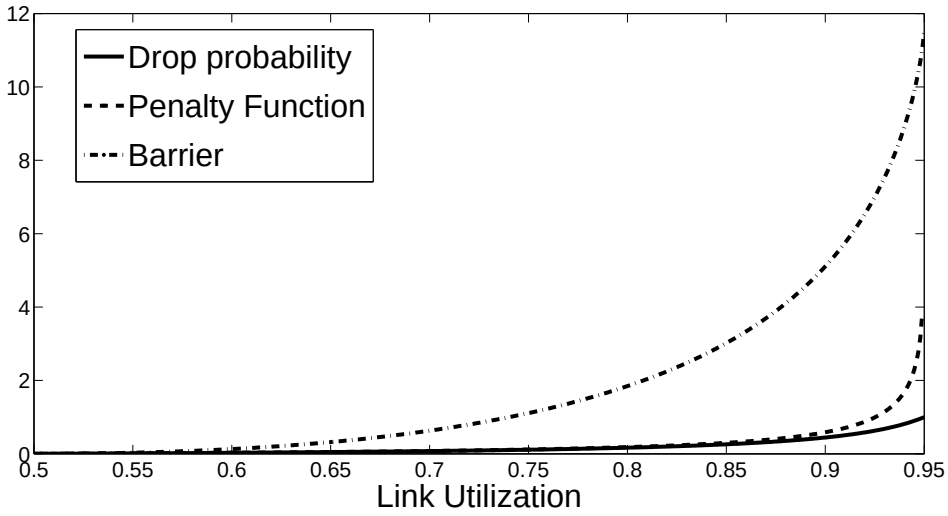


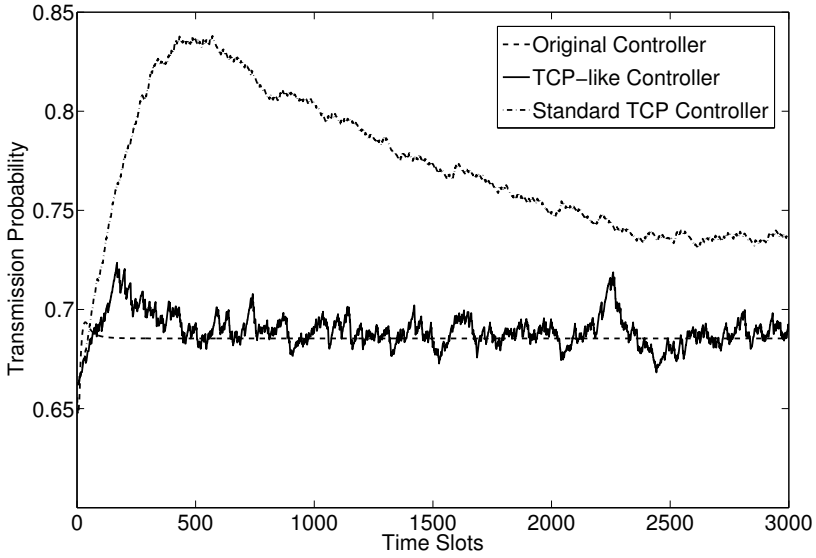
Figure 2.3. Link drop probability, penalty function and barrier for the RED scheme.

Then the link drop probability varies as

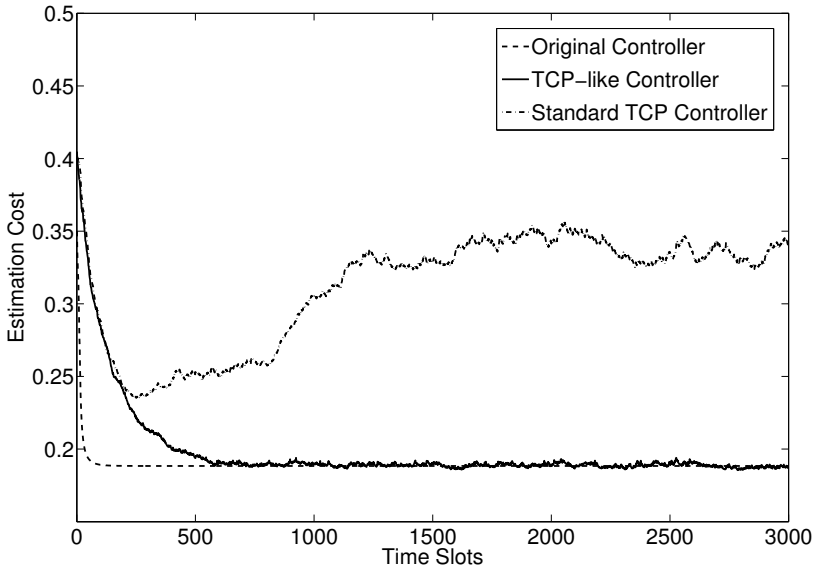
$$d_l = \begin{cases} 0 & \text{if } N < N_{min}, \\ \dfrac{N - N_{min}}{N_{max} - N_{min}} & \text{if } N_{min} \leq N \leq N_{max}, \\ 1 & \text{if } N > N_{max}\cdot, \end{cases}$$

where $N = \frac{\mu}{1-\mu}$ is the queue size. The values of $\{\mu_{min}, \mu_{max}\}$ are $\{0.5, 0.95\}$. We assume that the route drop probabilities are known to the sources.

Fig. 2.3 shows the link drop probability d_l, penalty function t_l and the scaled barrier $\beta_s B_l$ as a function of link utilization factor μ for a link that implements RED algorithm. For low rates, there are no drops. As the drop probability increase from 0 to 1, the penalty function becomes infinite. The barrier is scaled so that the congestion cost becomes large for large μ. All the three curves are positive, monotonically increasing and convex. Further, the penalty function is approximately equal to the drop probability for low values of link utilization.



(a) Transmission probabilities



(b) Estimation cost

Figure 2.4. Transmission probability and estimation costs achieved by various rate controllers.

Fig. 2.4 shows the temporal variation of the transmission probability of the second source ($p_2(k)$) and the $USER$ cost $C(\mathbf{p}(k))$ for the original probability controller PC_s(2.18), the TCP-like probability controller PC_s^{TCP} and the standard TCP rate controller. The transmission probabilities of the other sources also vary is a similar way and are omitted for clarity. We observe that PC_s achieves a steady state minimum cost of 0.1884 for the optimal transmission probability vector $\mathbf{p_{USER}} = [0.51, 0.69, 0.49, 0.43]$. It can be verified that the system parameters satisfy the conditions of theorem 2.4.1 and hence the overall system is stable. We also performed an exhaustive numerical search over the variable $\mathbf{p}$ to find the solution to the $USER$ problem. This exhaustive search yields the minimum value of the cost as 0.1863 which is quite close to the cost achieved by PC_s.

Similarly, an exhaustive numerical search over the variable $\mathbf{p}$ to find the solution of the $SYSTEM$ problem yields the minimum value of the cost as 0.41 which is achieved by $\mathbf{p_{SYS}} = [0.58, 0.62, 0.51, 0.46]$. We compare this with the $SYSTEM$ cost achieved by $\mathbf{p_{USER}}$, which is 0.47. We can observe that the solutions achieved by the proposed protocol for the $USER$ problem is close to the optimal solution of the $SYSTEM$ problem, thus verifying the approximation of the latter by the former.

Further, we observe from figure 2.4 that the cost achieved by PC_s^{TCP} fluctuates slightly with time due to its structure. More importantly, we can notice that the mean cost and the transmission probability achieved by PC_s^{TCP} coincides with the steady state cost and transmission probability of the original controller. This provides empirical evidence that under low network congestion conditions the TCP-like probability controller well approximates the original controller.

Moreover, as seen in Fig. 2.4, there is a big performance margin between the proposed controller and the TCP controller. This is because TCP controller is not suitable for estimation oriented applications since it minimizes a cost that is different from the estimation cost considered in this problem. To emphasize this point

further, we present a simple scenario in which the TCP controller results in an unstable system and the cost became infinite while the proposed controller maintains stability. Assume that the network consists of a single link with capacity $c_1 = 1.3$ shared by two sources. The process parameters are $\{A_1, C_1, Q_1, R_1\} = \{1.15, 1, 2, 2\}$ and $\{A_2, C_2, Q_2, R_2\} = \{2, 1, 2, 2\}$. The rest of the system parameters are same as before and we ignore delays. The minimum value of the transmission probabilities required to stabilize the estimation error covariance (Lemma 2.3.2) are $\{p_1^{min}, p_2^{min}\} = \{0.24, 0.75\}$. The TCP controller distributes the rate approximately equally among the two sources as $\{0.65, 0.65\}$. We can see that although the estimation error of the first process is stable, the estimation error of the second process becomes unbounded since the transmission probability is less than p_2^{min}. On the other hand, the optimal transmission probabilities achieved by the proposed controller is $\{0.36, 0.79\}$ and the error covariances remain stable. Thus, we can conclude that the standard TCP protocol may not be suitable for estimation oriented application since it caters to different set of applications, for example applications which require a notion of proportional fairness among the users.

2.6 Summary

We studied the problem of rate control for networked estimation in presence of congestion. A stochastic rate control protocol was proposed that optimizes the estimation performance of the network by varying the source transmission probabilities. The protocol was developed using a minimization framework analogous to NUM framework and is scalable for large networks. An approximated controller analogous to the standard TCP controller was also developed. The stability of the network was analyzed in presence of delays.

CHAPTER 3

DYNAMICAL PRIVACY IN MULTI-AGENT LINEAR TIME INVARIANT SYSTEMS

3.1 Background

A distributed dynamical system typically consists of multiple agents that collectively want to achieve a global system objective. For this purpose, the agents share their state information among each other and as a result, their dynamics become coupled with each other [26]. The distributed system may change over time due to variations in the system operating conditions such as the number of agents, interaction between the agents or changes in some underlying system parameters. Thus, a system administrator needs to continuously monitor the system to manage it more effectively. One possible way to perform this system monitoring is to collect the measured outputs of the agents at a central control center and then estimate the desired parameters using the outputs. For example, this type of architecture is used in a power grid network, where multiple geographically distributed sensors such as Phasor Measurement Units (PMUs) transmit their measurements to a control center. The control center then estimates the modes of the network model using suitable system identification methods [27].

Often, the dynamics of the overall coupled system depends on some underlying private parameters associated with each agent. These parameters contain sensitive information about the agents and should not be revealed to external entities. For example, in a consensus network [5], each node (agent) has multiple edges with associated edge weights. These edges represent coupling between the agents and collectively

form the network topology, which dictates the evolution(dynamics) of the consensus network. The edge weights contain sensitive information about the agents. For example, they contain information about loads and power flows in a power network [28]. As another example, consider a multi-agent LQR control problem wherein the agents are dynamically coupled, but have individual cost matrices, resulting in an overall decoupled cost function. The optimal control inputs (and the resulting state trajectory) depends on the state and input cost weighing matrices of the agents. In many applications such as biological systems, it has been shown that the cost matrices of an agent (human) represents the intent of a human ([29], [30] and references therein), which he/she would like to keep private. In many economic applications such as price determination, welfare planning and resource allocation, the agents solve an LQR problem to obtain optimal results [31]. In such cases, the cost matrices represent the pricing and welfare strategies of the agents, which they do not want to reveal to their competitors.

Due to the cyber-physical nature of the upcoming dynamical systems, there is a persistent threat of an intruder which may hack into the system and gain information about the private parameters of the agents (see the discussion in [32, 33] and the references therein). For instance, the intruder may snoop upon the measurements transmitted by the agents to the control center or it may hack the control center itself. It can then use the measurements to infer the sensitive parameters, thereby causing a privacy breach and gaining critical system level information. This information can further be used to plan an attack on the system. Therefore, protecting the privacy of these sensitive parameters of the agents is a crucial task for system operators.

3.1.1 Differential Privacy

In this dissertation, we use the notion of differential privacy(DP) to design the privacy mechanisms. The DP framework was originally proposed by Dwork [9] in

the context of computer databases which contain multiple entries that may represent sensitive information about individuals like their medical records, income, bank transactions etc. The response of a particular query submitted to the database should only provide statistical attributes of the population, without revealing any individual data. A particular simple solution to guarantee this privacy would be to anonymize the database entries. This can be done for example, by removing identifying information such as name, address or contact number of persons. However, it was shown that such anonymization techniques are not sufficient to guarantee privacy of the individuals [34]. This is because of the existence of *side information*, which can be obtained from external sources other than the database. The side information can then be correlated with the database responses to infer and identify particular individuals and their sensitive data. For example, in [34], it was shown that information from the IMDB movie database can be used to identify individuals (and their movie preferences) in an anonymized dataset of $500,000$ users released by Netflix for their data mining contest.

The motivation behind DP is that it is very difficult to quantify the side information and design a privacy mechanism that is robust to any possible side information. Therefore, it abstracts away from the notion of side information and provides a *differential* privacy guarantee, rather than an *absolute* guarantee. We next present the mathematical definitions of DP. Let the database, query and its output be denoted by D, q and $q(D)$, respectively. Let $\mathcal{M}$ denote the privacy mechanism that provides a response $\mathcal{M}_q(D)$ to the user.

Definition 3.1.1. ***(Differential privacy for databases)*** The privacy mechanism $\mathcal{M}$ is $\epsilon-$differentially private, if for any two databases D and D' that differ in at most one entry and for all $S \subset Range(\mathcal{M})$

$$\mathbb{P}[\mathcal{M}_q(D) \in S] \leq e^{\epsilon}\mathbb{P}[\mathcal{M}_q(D') \in S]. \tag{3.1}$$

The definition implies that under the randomized DP mechanism $\mathcal{M}$, the statistics of the output do not change significantly (within the multiplicative factor of e^{ϵ}) whether a user is present in a database or not. Thus, a user concerned about the leakage of its private information by participating in a database can be assured that regardless of the side information that an adversary might have, the probability of outputs will not differ significantly whether he/she is present or absent in the database. This *differential* guarantee would encourage more users to participate in the database. Note that ϵ^{-1} can be treated as the *privacy level*.

The DP mechanism randomizes the query output in order to mask the presence/absence of a user in the database. One way to do this is to add Laplacian noise to the query outputs as described below.

Lemma 3.1.2. ***(Laplacian mechanism)*** *Let $q(D) : D \to \mathbb{R}^k$ and D, D' be any two databases that differ in at most one entry. The mechanism $\mathcal{M}_q(D) = q(D) + N$ is $\epsilon-$differentially private if $N \sim Lap\left(0, \frac{\Delta_{q,1}}{\epsilon}\right)^k$, where $\Delta_{q,1}$ denotes the system sensitivity and is given by*

$$\Delta_{q,1} \triangleq \sup_{D,D'} \quad \|q(D) - q(D')\|_1 . \tag{3.2}$$

Proof. See [35], Theorem 2. ■

The sensitivity defines the maximum possible change in the output of a query (in terms of $1-$norm) applied on two databases that differ in a single user. The noise level is scaled according to the sensitivity: if it is high, a large amount of noise is needed to mask the presence/absence of the user, and vice versa. Next, we provide a more general definition of DP.

Definition 3.1.3. ***(Generalized DP for databases)*** The privacy mechanism $\mathcal{M}$ is $(\epsilon, \delta)-$differentially private, if for any two databases D and D' that differ in at

most one entry and for all $S \subset Range(\mathcal{M})$

$$\mathbb{P}[\mathcal{M}_q(D) \in S] \leq e^{\epsilon}\mathbb{P}[\mathcal{M}_q(D') \in S] + \delta \tag{3.3}$$

The (ϵ, δ)-DP relaxes the requirement of $\epsilon-$DP by an additive term $\delta < 0.5$ and can be achieved by adding a Gaussian noise to the query outputs.

Lemma 3.1.4. ***(Gaussian mechanism)*** *Let $q(D) : D \to \mathbb{R}^k$ and D and D' be any two databases that differ in at most one entry. The mechanism $\mathcal{M}_q(D) = q(D) + N$ is $(\epsilon, \delta)-$differentially private if $N \sim \mathbf{N}(0, \sigma^2 I_N)$, where*

$$\sigma \geq \frac{\Delta_{q,2}}{2\epsilon}(K + \sqrt{(K^2 + 2\epsilon)}),$$
$$K = \mathcal{Q}^{-1}(\delta) \quad \textit{and} \quad \Delta_{q,2} \triangleq \sup_{D,D'} \quad \|q(D) - q(D')\|_2 .$$

Proof. See [35], Theorem 3. ■

The Laplacian/Gaussian noises introduced by the above privacy mechanisms clearly introduce inaccuracies in the outputs of the queries. These noisy outputs degrade the utility of the database. The utility depends on various factors such as (i) the application for which the database is being used, (ii) how the database responses are used in a particular study, and so on. If a higher level of privacy is needed, the noise level needs to be increased, thereby further distorting the outputs and degrading the utility. Thus, a fundamental tradeoff exists between the privacy and utility in a database and characterizing this tradeoff is an important component of any study.

The DP framework was extended to dynamical systems in [35] in which the DP of individual input trajectories were guaranteed from a user who has access to the outputs. Similar to the static databases scenario, DP is achieved for dynamical systems by adding Laplacian/Gaussian noises to the system outputs. We use the DP framework of [35] in this dissertation to design privacy mechanisms. In this chapter,

we present a privacy mechanism for the class of linear time-invariant (LTI) dynamical systems that protects the privacy of the system level parameters associated with the agents which dictate the system dynamics. In this problem setting, ensuring DP of the system parameters means that for any two sets of parameters that are "not very different", the outputs of the corresponding linear dynamical systems will also be "statistically not very different". To ensure this property, each agent adds a synthetic noise to its state measurements before sending them to the control center. The noise is designed in such a way that the intruder cannot identify a "differential change" in the parameters from the noisy measurements, thereby keeping them private.

Recently, a number of works have appeared that present privacy mechanisms for dynamical systems. In [36], the authors propose a DP mechanism to keep the initial state private for the consensus problem and extend it in [37], to keep a reference trajectory private for a general distributed control system . In [38], a privacy mechanism involving careful noise addition and removal is presented to keep the initial state private and achieve exact consensus. Some papers have addressed privacy issues in optimization problems. In [39], [32] and [33], the authors present noisy update algorithms to protect private cost functions, constraints, and states, respectively. Further, [40] and [41] present mechanisms for privately solving optimization problems with linear and piecewise affine objectives, respectively. In [42], the authors present a stochastic gradient algorithm to solve a convex optimization problem wherein the noisy updates preserve DP. However, all these works aim to protect the privacy of the initial conditions, inputs, reference trajectories, cost functions etc. In contrast, our work aims to protect the parameters related to the system dynamics that are contained in the state evolution matrix of the LTI system. Note that the mapping from initial condition, input or reference trajectory to the output is linear whereas the mapping from the state matrix (and the parameters) to the output is non-linear. Characterizing this non-linear mapping poses additional challenges in our problem.

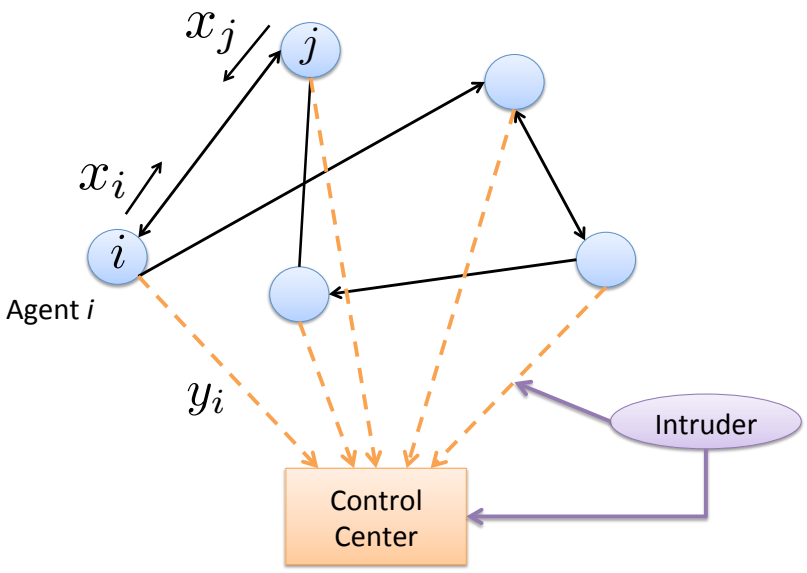


Figure 3.1. Dynamical system architecture

Moreover, all these works address privacy in a discrete time setting whereas our work presents privacy mechanism for continuous time systems.

3.2 Problem Setup

Figure 3.1 shows a distributed control system architecture consisting of N linear dynamical agents. The agents want to collectively achieve a global system objective. A directed solid line from agent i to agent j means that the evolution of agent j is coupled with agent i. The coupling can be present due to multiple reasons: (i) the dynamics of agent j is inherently coupled to that of agent i or, (ii) agent j uses the state information of agent i to update its own state, and others. The evolution of the whole system is represented as the following continuous-time LTI system

$$\dot{x}_c(t) = A(P)x_c(t) + B(P)u(t) + Fd(t) \quad t \geq 0, \tag{3.4}$$

with $x_c = \sigma\{[x_1^T, \cdots, x_N^T]^T\} \in \mathbb{R}^n$, where $x_i \in \mathbb{R}^{n_i}$ denotes the state of agent i with $\sum_{i=1}^N n_i = n$ and σ denotes the permutation operator. Further, $u = [u_1^T, \cdots, u_N^T]^T \in \mathbb{R}^m$ where $u_i \in \mathbb{R}^{m_i}$ is the control input to agent i with $\sum_{i=1}^N m_i = m$ and $d(t) \in \mathbb{R}^q$ denotes the disturbance in the system. Further, P_i denotes the set of parameters associated with the dynamics of agent i that it wishes to keep private and $P = \{P_1, P_2, \cdots, P_N\}$ denotes the collection of parameters of all agents. For instance, P_i may contain sensitive parameters that can reveal the coupling structure of agent i with other agents or its control preferences. We will illustrate this later using some examples. For the sake of analysis and without loss of generality, we treat the sets P_i and P as matrices and/or vectors, depending on the application. We study a class of systems in which the agents achieve the system objective by using a linear state feedback control law $u(t) = -K(P)x_c(t)$. Note that the agents share their state information among each other to implement the control law. The overall feedback system evolves as

$$\begin{aligned} \dot{x}_c(t) &= (A(P) - B(P)K(P))x_c(t) + Fd(t) \quad t \geq 0 \\ &\triangleq A_c(P)x_c(t) + Fd(t) \qquad P \in \mathcal{P}, \end{aligned} \tag{3.5}$$

where $\mathcal{P}$ denotes a given set of private parameters of all the agents. The output of the system is given by

$$y_c(t) = Cx(t), \tag{3.6}$$

where $y_c = \sigma\{[y_1^T, \cdots, y_N^T]^T\} \in \mathbb{R}^p$ where $y_i \in \mathbb{R}^{p_i}$ is the output of agent i with $\sum_{i=1}^N p_i = p$.

We make the following assumptions regarding the closed loop system in (3.5)-(3.6):

A1: *The closed loop state matrix $A_c(P)$ is stable or marginally stable for all $P \in \mathcal{P}$.*

Further, $\lim_{t\to\infty} x_c(t)$ *exists and is finite.*

A2: *The closed loop state matrix* $A_c(P)$ *is diagonalizable for all* $P \in \mathcal{P}$.

A3: *The system is fully observable, i.e. matrix* C *has full rank. Then, without loss of generality, we assume that* $C = I_n$ *and* $p = n$.

A4: *The pair* $(A_c(P), F)$ *is controllable for all* $P \in \mathcal{P}$.

A5: *The disturbance* $d(t)$ *is an impulse, i.e.* $d(t) = K_0\delta(t)$.

Stability assumption **A1** is standard for physical systems. It implies that $Re(\lambda_i(A_c(P))) \leq 0$ for $i \in \{1, 2, \cdots, n\}$ and in case the system is marginally stable, there is only a single pole on the imaginary axis which is located at the origin. Assumption **A2** enables us to design the privacy noise and will be used later in the proof of theorem 3.3.8. Assumption **A3** is valid since for efficient monitoring of the dynamical system, the control center typically has access to all the states of the system. Assumption **A4** is used in identification of the closed loop system, and will be explained later in this section. Assumption **A5** captures sudden faults in the system, which can be modeled as impulses.

The agents periodically sample their outputs at discrete time instants $t = 0, T_s, 2T_s, \cdots$ where $T_s > 0$ denotes the sampling time period. These measurements are sent to a control center as denoted by dashed lines in figure 3.1. The aim of the control center is to monitor the complete dynamical system using the measurements. Specifically, the control center wants to monitor the eigenvalues of the closed loop system $A_c(P)$. We will explain the eigenvalue identification procedure in detail in subsection 3.2.5. The discrete time measurements are denoted as

$$y(k) \triangleq y_c(kT_s) = Cx_c(kT_s), \quad k = 0, 1, 2, \cdots \tag{3.7}$$

where $y \in \mathbb{R}^p$. Since the measurements are available in discrete time to the control center, we convert the continuous time system (3.5) to discrete time. Since $d(t)$ is

an impulse (see assumption **A5**), we use the impulse invariant transform method to perform the discretization. The discrete time system can be written as

$$x(k+1) \triangleq x_c((k+1)T_s) = A_d(P)x(k) + F_d(P)d(k), \tag{3.8}$$

$$\text{where} \quad A_d(P) \triangleq e^{A_c(P)T_s}, \quad F_d(P) = A_d(P)F,$$

$$y(k) = Cx(k), \tag{3.9}$$

and $d(k)$ is an impulse in discrete time, i.e. $d(k) = K_0\delta(k)$. Note that the poles of the discrete time system are related to that of the continuous time system as

$$\lambda(A_d(P)) = e^{T_s\lambda(A_c(P))}. \tag{3.10}$$

We make the following assumptions regarding the discretized system:

A6: *The pair $(A_d(P), F_d(P))$ is controllable for all $P \in \mathcal{P}$.*

A7: *The sampling time period satisfies the following property*

$$T_s < \bar{T}_s \triangleq \frac{\pi}{\sup\limits_{P\in\mathcal{P},\, i\in\{1,2,\cdots,n\}} |Im(\lambda_i(A_c(P)))|}. \tag{3.11}$$

Assumption **A6** is used in identification of the discrete time closed loop system and assumption **A7** is made to prevent aliasing and enable identification of the continuous time system from the discrete time system. However, the sampling time cannot be arbitrarily small since it will cause the poles of the discrete time system to cluster around 1, creating numerical issues in the identification procedure. The identifications procedures will be explained in subsections 3.2.3 and 3.2.5. The evolution of the discrete time system (3.8) with the impulse input $d(k) = K_0\delta(k)$ (see

assumption **A5**) can be written as

$$x(k) = A_d^k(P)(x(0) + FK_0). \tag{3.12}$$

Next, we illustrate our problem setup using examples of consensus and optimal control.

3.2.1 Example I: Second-order Consensus Network

Consider a dynamical system with N agents whose objective is to collectively achieve consensus asymptotically. The system can be represented as a undirected graph $(\mathcal{N}, \mathcal{E}, G)$ where $\mathcal{N} = \{1, 2, \cdots, N\}$ denotes the set of nodes (agents), $\mathcal{E} \in \mathcal{N} \times \mathcal{N}$ denotes the set of edges where an edge between agents i and j is represented as pair (i, j) and it means that agents i and j are dynamically coupled with each other. Further, $G \in \mathbb{R}^{N \times N}$ denotes the non-negative weighted adjacency matrix with $[G]_{ij} = g_{ij} \geq 0$. Let $\mathcal{N}_i$ denote the set consisting of the neighbors of agent i, i.e. $\mathcal{N}_i = \{j : (i, j) \in \mathcal{E}\}$. The agents evolve according to the following dynamics

$$M_i \ddot{w}_i(t) = u_i(t) + \bar{f}_i d_i(t) \quad i = 1, \cdots, N \tag{3.13}$$

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} g_{ij}(w_i(t) - w_j(t)) - D_i \dot{w}_i(t), \tag{3.14}$$

where $w_i \in \mathbb{R}, d_i \in \mathbb{R}$ and $M_i \in \mathbb{R}, D_i \in \mathbb{R}$ denote the positive inertia and damping of the i^{th} agent, respectively. Such second order dynamics are commonly used in modeling transportation systems [43], robot motions or electro-mechanical power networks [28].

Definition 3.2.1. ***(Consensus)*** The agents achieve consensus if (i) $\lim_{t \to \infty} |w_i(t) - w_j(t)| = 0$ for all agents $i \neq j$ and (ii) $\lim_{t \to \infty} |\dot{w}_i(t)| = 0$ for all agents $i \in \mathcal{N}$.

The dynamics of the whole system can be written as

$$\dot{x}_c(t) = \underbrace{\begin{bmatrix} \mathbf{0}_{N\times N} & I_N \\ \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} \end{bmatrix}}_{A} x_c(t) + \underbrace{\begin{bmatrix} \mathbf{0}_{N\times N} \\ M^{-1} \end{bmatrix}}_{B} u(t) + \underbrace{\begin{bmatrix} \mathbf{0}_{N\times N} \\ M^{-1}\bar{F} \end{bmatrix}}_{F} d(t),$$

$$\overset{(a)}{=} \underbrace{\begin{bmatrix} \mathbf{0}_{N\times N} & I_N \\ -M^{-1}L & -M^{-1}D \end{bmatrix}}_{A_c(L)} x_c(t) + Fd(t), \tag{3.15}$$

where (a) follows from the control law (3.14), $x_i = [w_i, \dot{w}_i]^T \in \mathbb{R}^2$ is the state of agent i, $x_c = [w_1, \cdots, w_N, \dot{w}_1, \cdots, \dot{w}_N]^T \in \mathbb{R}^{2N}$, $u = [u_1, \cdots, u_N]^T \in \mathbb{R}^N$, $d = [d_1, \cdots, d_N]^T \in \mathbb{R}^N$, $M = diag(M_i, \cdots, M_N) \in \mathbb{R}^{N\times N}$, $D = diag(D_1, \cdots, D_N) \in \mathbb{R}^{N\times N}$, $\bar{F} = diag(\bar{f}_1, \cdots, \bar{f}_N) \in \mathbb{R}^{N\times N}$, and $L \in \mathbb{R}^{N\times N}$ is the symmetric positive semidefinite Laplacian matrix defined as

$$[L]_{ij} \triangleq l_{ij} = \begin{cases} \sum\limits_{j\in\mathcal{N}_i} g_{ij} & \text{if} \quad i = j, \\ -g_{ij} & \text{otherwise.} \end{cases} \tag{3.16}$$

We assume that the graph is connected. As a result, the closed loop state matrix $A_c(L)$ has one eigenvalue at 0 and all other eigenvalues have negative real part [44], and hence it satisfies assumption **A1**. We will later show that the dynamical system (3.15) achieves consensus. The laplacian matrix L represents the complete topology of the network and the edge weights contain crucial and sensitive information about the network, which should not be revealed [45]. For example, the edge weights may contain information about the voltages and loads in a power network [28]. Specifically, each agent (node) wants to keep private all the edge weights associated with it. Comparing the consensus evolution (3.15) with the general dynamical system (3.5), we get $P_i = \{l_{ij} : j \in \mathcal{N}_i\}$, $P = L$ and, $A_c(P) = A_c(L)$.

3.2.2 Example II: LQR Control

Consider a set of N agents with coupled dynamics that evolve as

$$\dot{x}_c(t) = Ax_c(t) + Bu(t) + Fd(t), \tag{3.17}$$

where $x_c = [x_1, \cdots, x_N]^T \in \mathbb{R}^n$. The agents aim to minimize the following infinite horizon decoupled quadratic cost function

$$\begin{aligned} J(x,u) &= \int_0^\infty \sum_{i=1}^{N} (x_i^T(t)Q_i x_i(t) + u_i^T(t)R_i u_i(t))dt \\ &= \int_0^\infty (x_c^T(t)Qx_c(t) + u^T(t)Ru(t))dt, \end{aligned} \tag{3.18}$$

where $Q_i \in \mathbb{R}^{n_i \times n_i} \geq 0$, $R_i \in \mathbb{R}^{m_i \times m_i} > 0$, $Q = diag(Q_1, \cdots, Q_N) \in \mathbb{R}^{n \times n}$ and $R = diag(R_1, \cdots, R_N) \in \mathbb{R}^{m \times m}$. The cost matrices Q_i, R_i contain private information about the control strategies of the agents. For example, in biological systems, the cost matrices can represent human intent [29, 30] and in economic applications, they can represent pricing and welfare strategies [31].

We make the following assumptions for the LQR problem:

A8.1: *The pair (A, B) is controllable.*

A8.2: *The input cost matrix $R = I_m$.*

A8.3: *Rank(B) = n.*

Assumption **A8.1** is standard in LQR problems. Assumption **A8.2** means that the input cost matrix R is specified for the system and the agents only wish to keep the state cost matrix Q private. Further, without loss of generality, we assume R to be an identity (otherwise, one can always do a coordinate transformation s.t. $R = I_m$). This is a standard assumption in most of the inverse control problems [46], [47] and also simplifies the analysis (c.f. lemma 3.3.12). Assumption **A8.3** ensures that the cost matrix Q can be uniquely identified from the optimal control gain defined below

(see [47], Theorem 7).

The optimal control law that minimizes the cost in (3.18) depends on Q, and is given by

$$u(t) = -K(Q)x_c(t), \qquad K(Q) = R^{-1}B^T V(Q), \tag{3.19}$$

where $V(Q)$ is the unique positive definite solution of the following Riccati equation

$$A^T V(Q) + V(Q)A - V(Q)BR^{-1}B^T V(Q) + Q = 0. \tag{3.20}$$

Since (A, B) is controllable, the closed loop matrix $A - BK(Q)$ is stable (cf. assumption **A1**). Comparing the closed loop evolution of the LQR system with the general dynamical system (3.5), we get $P_i = Q_i, P = diag\{P_i\} = Q$, and $A_c(Q) = A - BK(Q)$.

The above examples illustrate that our formulation is quite general and encompasses a wide variety of problems in which the agents' dynamics is (i) linear and (ii) contains private parameters. Next, we examine the privacy issues that may arise in such distributed control systems.

3.2.3 Privacy Issues in Dynamical System

Due to the cyber-physical nature of the dynamical system architectures, there always exist a threat of an external intruder that can gain an unauthorized access into the system. This is usually the case before planning an attack on the system, wherein the intruder wants to first obtain private information about the individual components (agents) of the system. For this purpose, the intruder might snoop upon the messages communicated by the agents to the control center, or it may hack into the control center itself, as shown in figure 3.1, thereby gaining access to the measurements $y(k)$. In addition to the measurements, the intruder may have some

information about the system that is obtained from some external sources. Towards this, we make the following assumption:

A9: *The intruder has information about the initial condition $x(0)$, F, the disturbance $d(t)$ and sampling time period T_s. Further, it may also have information about some non-private parameters of the system.*

This assumption implies that the intruder knows that the system is excited by an impulse disturbance, the magnitude of the impulse and how the disturbance affects the system. Also, some non-private parameters, for ex. A, B, C in the LQR example, are known to the intruder. Collectively, all the information about the system except the private parameters P is called *side information* or auxiliary information.

Remark 3.2.2. ***(Side information)*** Assumption **A9** is not restrictive to the problem because there always exists a possibility that such side information regarding the system is known to the intruder. In fact, one of the main motivations behind the differential privacy framework is to develop a privacy mechanism that abstracts away from arbitrary side information that the intruder might possess [9] (also see remark 3.2.8). Moreover, assumption **A9** represents the worst case scenario and the capabilities of the intruder will be further limited in case it does not have some of this information. □

The aim of the intruder is to infer the private parameters P using the measurements and the side information. In general, the measurements of the system depend on the parameters P in a non-linear fashion. Therefore, the intruder uses a Non-linear Least Squares (NLS) optimization procedure to identify the parameters. It uses an indirect identification method in which the discrete time state matrix is identified first, and then it is used to obtain the continuous time state matrix, from which the private parameters are extracted. The parameter identification method is described as follows

P1) *Using* (3.12)*, solve the following NLS problem to obtain a discrete time state*

matrix estimate using the discrete time measurements

$$\hat{A}_{d,T}(\hat{P}_T) = \underset{Z \in \mathbb{R}^{n \times n}}{\arg\min} \sum_{k=0}^{T} \left\| y(k) - CZ^k(x(0) + FK_0) \right\|_2^2, \tag{3.21}$$

where T is the identification time horizon.

P2) *Obtain a continuous time state matrix estimate from the discrete time state matrix estimate using $\hat{A}_{c,T}(\hat{P}_T) = \frac{1}{T_s} log(\hat{A}_{d,T}(\hat{P}_T))$.*

P3) *Extract the private parameter $\hat{P}_T$ from $\hat{A}_{c,T}(\hat{P}_T)$.*

Since the discrete time system (3.8)-(3.9) is fully observable (assumption **A3**) and controllable (assumption **A6**), the discrete time state matrix $A_d(P)$ can be uniquely and accurately identified by the intruder from the discrete outputs and the impulse input in step **P1** [48] . From the discrete time state matrix, the intruder can then accurately obtain the continuous time state matrix $A_c(P)$ in step **P2**, since condition (3.11) on the sampling time period guarantees that there is no aliasing. From $A_c(P)$, the intruder can then accurately identify the parameters P. Thus, $\hat{P}_T = P$ and the intruder gains access to the sensitive parameters, which is clearly undesirable. Therefore, a privacy mechanism needs to be integrated into the dynamical system that prevents such kind of privacy breaches. In the next subsection, we present a noise adding privacy mechanism using the notion of differential privacy.

Remark 3.2.3. ***(Non-identifiability of the private parameters)*** In some cases, it may be possible that the mapping from P to $A_c(P)$ may be many to one (for ex., LQR problem without assumption **A8.3**). In other cases, the intruder may not have enough side information to infer P from $A_c(P)$. For example, in the consensus problem, if the intruder does not know M, then it cannot identify L from $A_c(L)$. It can only identify $M^{-1}L$ (see (3.15)). In all such cases, the intruder will not be able to uniquely identify the parameters from the measurements. □

Remark 3.2.4. ***(Intruder vs control center)*** We emphasize the fact that the

objectives of the intruder and the control center are different. The control center wants to monitor the eigenvalues (explained in subsection 3.2.5) of the system to verify that it is functioning correctly. On the other hand, the aim of the intruder is to perform an attack into the system for which it requires information about the private parameters, and the sole knowledge of eigenvalues of the system will not be sufficient. □

3.2.4 Differential Privacy Mechanism

We motivated and introduced the DP framework in the context of static databases in subsection 3.1.1. In this subsection, we present DP definitions pertaining to the dynamical system considered in this problem, using the DP framework developed in [35]. In order to prevent the intruder from accurately identifying the private parameters, the the agents add noise to the measurements before transmitting them to the control center. The privacy noise ensures the following *differential* property: if the private parameters are *"changed within some specified limits"*, then the corresponding measurements appear *"probabilistically almost similar"* to the intruder. In other words, the DP noise masks any change in the parameters to the intruder. Thus, the intruder will not be able to distinguish between the two parameters with a high confidence level, thereby preserving their privacy. Next, we provide formal definitions of differential privacy specific to this problem.

Definition 3.2.5. ***(Adjacency)*** Two parameters P and $P^{'}$ (which are essentially matrices) are β-adjacent (denoted by $adj(\beta)$) if for some $\beta \geq 0$ we have

$$\left\| P - P^{'} \right\|_2 \leq \beta. \tag{3.22}$$

Remark 3.2.6. ***(Generalized adjacency)*** In the DP definition for static databases [9] and for dynamical systems [35], adjacency is defined w.r.t. the change of data/input

of *one* agent while keeping the data/inputs of other agents unchanged. In contrast, our definition of adjacency is more general and allows changes in the parameters of one or more agents. □

As mentioned before, the agents add noise to the measurements according to the following DP mechanism

$$\mathcal{M}: \qquad \tilde{y}(k) = y(k) + w(k), \tag{3.23}$$

where $w(k) \in \mathbb{R}^n$ is the noise. We will specify the properties of the privacy noise in section 3.3.

Let $\tilde{y}_P$ denote the noisy measurements of the of the system with private parameter P. Note that $\tilde{y}_P[0:T] \in \mathbb{R}^{n(T+1)}$ and let $\mathcal{R}^{n(T+1)}$ denote the $\sigma - algebra$ generated by it. Next, we provide the definition of differential privacy.

Definition 3.2.7. ***(Differential privacy)*** The mechanism $\mathcal{M}$ in (3.23) is ϵ-differentially private upto time T if for any two β-adjacent parameters P and $P^{'}$ and for all $S \subset \mathcal{R}^{n(T+1)}$

$$\mathbb{P}[\tilde{y}_P[0:T] \in S] \leq e^{\epsilon}\mathbb{P}[\tilde{y}_{P'}[0:T] \in S], \tag{3.24}$$

where $\epsilon > 0$ is the DP parameter. The definition says that if the parameter changes from P to $P^{'}$ that is β-adjacent to P, then the corresponding measurement probabilities change only within a factor of e^{ϵ}. Note that a smaller value of ϵ implies a higher level of privacy and vice versa.

Remark 3.2.8. ***(Differential vs absolute privacy)*** We would like to emphasize that the DP mechanism does not guarantee absolute privacy of the parameters. In some scenarios, the intruder may be able to obtain an extensive amount of side information and it can identify some of the private parameters even without using

the measurements. The only guarantee DP provides is that there will only be a marginal privacy loss (determined by ϵ) due to any changes in the private parameters of the agents within a specified limit (determined by β). Thus, it abstracts away from any possible side information that the intruder might have [10]. □

The privacy noise will prevent the intruder from accurately identifying the private parameters. Instead, the intruder will use the noisy measurements in step **P1** of the parameter identification procedure, which can be re-written as (for comparison, see (3.21))

$$\hat{A}_{d,T}(\hat{P}_T) = \underset{Z\in\mathbb{R}^{n\times n}}{\arg\min} \sum_{k=0}^{T} \left\| \tilde{y}_P(k) - CZ^k(x(0) + FK_0) \right\|_2^2, \tag{3.25}$$

and the steps **P2** and **P3** remain the same. The inaccuracy in the identification will preserve the privacy of the parameters. The parameter identification error suffered by the intruder can be quantified as

$$\mathcal{E}_P = \mathbb{E}\left[\left\| \hat{P}_T - P \right\|_F \right], \tag{3.26}$$

where the expectation $\mathbb{E}$ is taken $w.r.t$ the noise. We will present numerical simulation results regarding the identification error in section 3.4, and show that it increases with increase in privacy level (or noise).

Remark 3.2.9. ***(Advanced identification techniques)*** We have presented one technique for parameter identification. There may be alternative and possibly advanced techniques that the intruder may employ to obtain a better estimates from the noisy measurements. However, a fundamental property of DP mechanism is its resilience to post processing, i.e. one cannot weaken the DP guarantee by processing the differentially private outputs using *any* technique [35]. Therefore, the DP mechanism is robust to any identification/processing technique used by the intruder and

hence, finding an optimal technique is not the central premise of this problem. □

Next, we explain how the noisy DP mechanism affects the eigenvalue identification procedure at the control center.

3.2.5 Eigenvalue Identification by Control Center

As mentioned earlier, the objective of the control center is to identify the eigenvalues of the continuous time closed loop state matrix $A_c(P)$. This identification is performed using the noisy measurements $\tilde{y}(k)$ that the control center receives from the agents at discrete times $t = 0, T_s, 2T_s, \cdots$. First, the control center identifies the eigenvalues of the discrete time state matrix $A_d(P)$ by identifying the coefficients of its characteristic polynomial. Next, the eigenvalues of the continuous time system are computed using the eigenvalues of the discrete time system. The upper bound on the sampling time in (3.11) prevent aliasing during this reverse sampling step. For identification purposes, the impulse disturbance $d(k)$ acts like an input to the discrete time closed loop system in (3.8)-(3.9). We denote the transfer function matrix from d to y as

$$H(z) = C(zI_n - A_d(P))^{-1}F_d \quad \text{with,} \tag{3.27}$$

$$[H]_{ij} = \frac{\sum_{k=1}^{n} b_{ij}^{(k)} z^{-k}}{1 + \sum_{k=1}^{n} a^{(k)} z^{-k}}, \tag{3.28}$$

where $a_d(z) = 1 + \sum_{k=1}^{n} a^{(k)} z^{-k}$ is the characteristic polynomial of $A_d(P)$. Let $a = [a^{(1)}, a^{(2)}, \cdots, a^{(n)}]^T, b_i^{(k)} = [b_{i1}^{(k)}, b_{i2}^{(k)}, \cdots, b_{iq}^{(k)}]^T, b_i = [(b_i^{(1)})^T, (b_i^{(2)})^T, \cdots, (b_i^{(n)})^T]^T$. Us-

ing the transfer function, the noisy outputs can be written as

$$\tilde{y}(k) = \varphi^T(k)\theta + v(k), \tag{3.29}$$

$$\text{where,} \quad v(k) = \sum_{i=1}^{n} a^{(i)} w(k-i),$$

$$\varphi(k) = \begin{bmatrix} -\tilde{y}^T(k-1) \\ -\tilde{y}^T(k-2) \\ \vdots \\ -\tilde{y}^T(k-n) \\ I_p \otimes \begin{bmatrix} d(k-1) \\ d(k-2) \\ \vdots \\ d(k-n) \end{bmatrix} \end{bmatrix}_{(n+pqn)\times p} \quad \text{and} \quad \theta = \begin{bmatrix} a \\ b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix}_{(n+pqn)\times 1} \tag{3.30}$$

The data vector φ is constructed from the noisy measurements and the impulse disturbance. Further, θ is the vector consisting of all the numerator and denominator coefficients of the transfer function $H(z)$. Note that the vector a contains the coefficients of the characteristic polynomial of $A_d(P)$, from which the eigenvalues can be calculated. Thus, first θ is estimated using the input-output data and then a is extracted from the estimate.

The identification is performed using the instrumental variable (IV) method, which is stated as follows

$$\begin{aligned} \hat{\theta}_T &= sol\left\{ \frac{1}{T}\sum_{k=1}^{T} \zeta(k)[\tilde{y}(k) - \varphi(k)^T\theta] = 0 \right\}, \\ &= \left(\frac{1}{T}\sum_{k=1}^{T} \zeta(k)\varphi(k)^T \right)^{-1} \frac{1}{T}\sum_{k=1}^{T} \zeta(k)\tilde{y}(k), \\ &= \theta + \left(\frac{1}{T}\sum_{k=1}^{T} \zeta(k)\varphi(k)^T \right)^{-1} \frac{1}{T}\sum_{k=1}^{T} \zeta(k)v(k), \end{aligned} \tag{3.31}$$

where *sol* denotes the solution of an equation, T is the time horizon for identification, and $\zeta(k)$ are appropriately chosen instruments that are uncorrelated with the noise $v(k)$. We do not present the details of the choice of instruments here. These results are a part of standard identification theory and we refer the interested reader to [49] for further details.

Let $\hat{a}_T$ denote the estimate of a that is extracted from $\hat{\theta}_T$. The eigenvalue identification procedure can be summarized as follows:

E1) *Obtain the estimate* $\hat{\theta}_T$ *using the IV method in* (3.31).

E2) *Extract* $\hat{a}_T$ *from* $\hat{\theta}_T$ *using its structure given in* (3.30).

E3) *Obtain the eigenvalue estimates of the discrete time system* $\hat{\lambda}(A_d(P))$ *from* $\hat{a}_T$.

E4) *Obtain the eigenvalue estimates of the continuous time system using* $\hat{\lambda}(A_c(P)) = \frac{1}{T_s}\log(\hat{\lambda}(A_d(P)))$.

Let a_c denote the coefficients of the characteristic polynomial of $A_c(P)$ and let $\hat{a}_{c,T}$ denote its estimate, which is calculated from the estimated eigenvalues $\hat{\lambda}(A_c(P))$. We evaluate the eigenvalue identification performance in terms of the estimation error of these coefficients

$$\mathcal{E}_{a_c} = \mathbb{E}\left[\|\hat{a}_{c,T} - a_c\|_2\right], \tag{3.32}$$

where the expectation is taken w.r.t the privacy noise.

3.3 Noise Design for Differential Privacy

In the previous section, we introduced the noise adding privacy mechanism in (3.23). In this section, we present the properties of the privacy noise $w(k)$ that guarantee that the mechanism $\mathcal{M}$ satisfies the DP criterion of (3.24). As standard in the literature [35, 50], the noise provides DP if it satisfies the following two conditions (i) it is Laplacian and white, and (ii) the noise level is calibrated according to the

sensitivity of the system. We also use the Laplacian mechanism for providing DP. Since we want to protect the parameters P from an intruder that has access to the measurements $y(k)$, we compute the sensitivity from P to y for determining the noise level. If this sensitivity is low, then the measurements do not change significantly with the change the parameters. Therefore, a lower level of noise needs to be added to the measurements to mask the change in the parameters, and vice versa. Thus, the required noise level is proportional to the sensitivity of the system. Next, we present the formal definition of sensitivity specific to our problem.

Definition 3.3.1. ***(Sensitivity)*** The system sensitivity at time instant $k \geq 0$ is defined as

$$\Delta(k) = \sup_{P,P':adj(\beta)} \|y_P(k) - y_{P'}(k)\|_1 . \tag{3.33}$$

The sensitivity characterizes the maximum possible difference (in terms of 1-norm) between the measurements resulting from any two possible adjacent parameters. It depends on a number of system parameters and we will characterize it later in this section. Next, we show that we can use the sensitivity to design the privacy noise.

Theorem 3.3.2. ***(Noise design for DP)*** *The mechanism $\mathcal{M}$ in (3.23) is ϵ-differentially private upto time T if $w(k)$ is white Laplacian with*

$$w(k) \sim Lap(0, c_k)^p \quad and \quad \sum_{k=0}^{T} \frac{\Delta(k)}{c_k} \leq \epsilon. \tag{3.34}$$

Proof. From the mechanism $\mathcal{M}$, we have $\tilde{y}_P[0:T] = y_P[0:T] + w[0:T]$. Let f_P and $f_{P'}$ denote the probability density functions of $\tilde{y}_P[0:T]$ and $\tilde{y}_{P'}[0:T]$, respectively.

Further, let $x = [x_0^T, x_1^T, \cdots, x_T^T]^T \in \mathbb{R}^{n(T+1)}$ denote the integration variable. Then

$$\begin{aligned}
\mathbb{P}[\tilde{y}_P[0:T] \in S] &= \int_S f_P(x)dx \\
&\overset{(a)}{=} \int_S \prod_{k=0}^{T} \frac{1}{(2c_k)^n} e^{\frac{-\|x_k - y_P(k)\|_1}{c_k}} dx_k \\
&\overset{(b)}{\leq} \int_S \prod_{k=0}^{T} \frac{1}{(2c_k)^n} e^{\frac{-\left\|x_k - y_{P'}(k)\right\|_1}{c_k}} e^{\frac{\left\|y_P(k) - y_{P'}(k)\right\|_1}{c_k}} dx_k \\
&\overset{(c)}{\leq} e^{\left(\sum_{k=0}^{T} \frac{\Delta(k)}{c_k}\right)} \int_S \prod_{k=0}^{T} \frac{1}{(2c_k)^n} e^{\frac{-\left\|x_k - y_{P'}(k)\right\|_1}{c_k}} dx_k \\
&\overset{(d)}{\leq} e^{\epsilon} \mathbb{P}[\tilde{y}_{P'}[0:T] \in S],
\end{aligned}$$

where (a) follows from the joint Laplacian distribution of $w[0:T]$, (b) follows from the triangle inequality

$$-\|x-p\|_1 \leq -\left\|x-p'\right\|_1 + \left\|p-p'\right\|_1$$

(c) follows from the definition of sensitivity and (d) follows from the condition given in the theorem. Thus, the DP condition (3.24) is satisfied. ■

From the preceding theorem, it is clear that in order to design the noise, we need to characterize the sensitivity of the system. However, it is difficult to obtain an exact expression for the sensitivity. Therefore, we next obtain an upper bound on the sensitivity which can be used to design the noise level.

3.3.1 Upper Bound on Sensitivity

Let $\bar{x}(P)$ denote the steady state value of the discrete time system (3.12) (also see assumption **A1**). If $A_d(P)$ is stable, then $\bar{x}(P)$ is trivially zero. For marginally stable systems (i.e. a single eigenvalue on the unit circle located at 1: see assumption

A1), using assumption **A2**, the steady state value can be readily obtained as

$$\begin{aligned} \bar{x}(P) &= \bar{A}_d(P)(x(0) + FK_0), \quad \text{where} \\ \bar{A}_d(P) &\triangleq \lim_{k\to\infty} A_d^k(P) = \nu_1(A_d(P))\tilde{\nu}_1(A_d(P)) \quad \text{with,} \\ \tilde{\nu}_1(A_d(P))&\nu_1(A_d(P)) = 1, \end{aligned} \tag{3.35}$$

where ν_λ and $\tilde{\nu}_\lambda$ denote the right and left eigenvectors associated with the eigenvalue λ, respectively. We make the following assumption regarding the steady state.

A10: *The steady state value $\bar{x}(P)$ is independent of the private parameters P.*

This assumption is trivially satisfied for stable systems since the steady state is zero. For marginally stable systems, this assumption guarantees that the sensitivity defined in (3.33) decays with time and the privacy noise level remains bounded. We will further comment on this assumption later in the section. In view of this assumption, we drop the dependence of $\bar{x}(P)$ and $\bar{A}_d(P)$ on the privacy parameter P and denote them by $\bar{x}$ and $\bar{A}_d$, respectively.

For marginally stable systems, we use the fact that the system describing the evolution of the error defined as $e(k) \triangleq x(k) - \bar{x}$, is stable. Towards this, we have the following lemma.

Lemma 3.3.3. ***(Error evolution)*** *The dynamics of the error can be represented as*

$$\begin{aligned} e(k+1) &= \tilde{A}_d(P)e(k) \quad \textit{for} \quad k \geq 0, \quad \textit{where} \\ \tilde{A}_d(P) &= \begin{cases} A_d(P) & \textit{if } A_d(P) \textit{ is stable,} \\ A_d(P) - \bar{A}_d & \textit{if } A_d(P) \textit{ is marginally stable,} \end{cases} \\ \textit{and} \quad e(0) &= x(0) + FK_0. \end{aligned} \tag{3.36}$$

Proof. See appendix A.1. ■

Next, we derive the eigenvalues of the error dynamics for the marginally stable case and show that it is stable.

Lemma 3.3.4. ***(Error eigenvalues)*** *For marginally stable systems, the set of eigenvalues of* $\tilde{A}_d(P)$ *is*

$$\lambda(\tilde{A}_d(P)) = \{0, \{\lambda_i(A_d(P)) : \lambda_i(A_d(P)) \neq 1\}_{i=1}^{n}\}. \tag{3.37}$$

Proof. See appendix A.1. ■

From the preceding lemma, it is clear that $\rho(\tilde{A}_d(P)) < 1$ and therefore the error dynamics is stable.

Corollary 3.3.5. ***(Simultaneous diagonalizability)*** $A_c(P), A_d(P)$ *and* $\tilde{A}_d(P)$ *are simultaneously diagonalizable.*

Proof. Since $A_c(P)$ is diagonalizable (assumption **A2**) and $A_d(P) = e^{A_c(P)T_s}$, it is trivial to observe that $A_c(P)$ and $A_d(P)$ are simultaneously diagonalizable. Further, from the proof of Lemma 3.3.4, we can observe that both $A_d(P)$ and $\tilde{A}_d(P)$ have same eigenvectors and thus, they are simultaneously diagonalizable. ■

Now, the measurements can be represented in terms of the error as

$$y(k) = C(e(k)+\bar{x}) = C(\tilde{A}_d^k(P)(x(0) + FK_0) + \bar{x}). \tag{3.38}$$

Next, we define some global quantities of the system, which will be used in obtaining the upper bound to the sensitivity. Using corollary 3.3.5, let $A_c(P) = X^{-1}(P)\Lambda_c(P)X(P), A_d(P) = X^{-1}(P)\Lambda_d(P)X(P)$ and $\tilde{A}_d(P) = X^{-1}(P)\tilde{\Lambda}_d(P)X(P)$, where $\Lambda_c(P), \Lambda_d(P)$ and $\tilde{\Lambda}_d(P)$ are diagonal matrices consisting of the eigenvalues

of $A_c(P), A_d(P)$ and $\tilde{A}_d(P)$, respectively. Then, let

$$\kappa_{max} \triangleq \sup_{P\in\mathcal{P}} \quad \kappa(X(P)) = \sup_{P\in\mathcal{P}} \quad \|X(P)\|_2\|X^{-1}(P)\|_2 \tag{3.39}$$

$$\rho_{max} \triangleq \sup_{P\in\mathcal{P}} \quad \rho(\tilde{A}_d(P)). \tag{3.40}$$

Next, we provide two known results which will be used to derive the upper bound to the sensitivity.

Lemma 3.3.6. ***(Perturbation of matrix powers [51])*** *Let A and $\tilde{A}$ be two $n\times n$ diagonalizable matrices for which $\rho(A) < 1$ and $\rho(\tilde{A}) < 1$. Let X and $\tilde{X}$ be invertible matrices which yield diagonal matrices S and $\tilde{S}$ via similarity transformations $X^{-1}AX = S$ and $\tilde{X}^{-1}\tilde{A}\tilde{X} = \tilde{S}$. Then,*

$$\|A^k - \tilde{A}^k\|_1 \leq \kappa(X)\kappa(\tilde{X})kn\|A-\tilde{A}\|_2 \max\{\rho(A), \rho(\tilde{A})\}^{k-1}.$$

Proof. See appendix A.1. ■

Lemma 3.3.7. ***(Perturbation of matrix exponential [52])*** *Let A and $\tilde{A}$ be two square, diagonalizable matrices. Let X and $\tilde{X}$ be invertible matrices which yield diagonal matrices S and $\tilde{S}$ via similarity transformations $X^{-1}AX = S$ and $\tilde{X}^{-1}\tilde{A}\tilde{X} = \tilde{S}$. Further, let $t > 0$. Then,*

$$\|e^{At} - e^{\tilde{A}t}\|_2 \leq \|A - \tilde{A}\|_2 t\kappa^2(X, \tilde{X})e^{\kappa(X,\tilde{X})\|A-\tilde{A}\|_2 t},$$

where $\kappa(X, \tilde{X}) \triangleq \max\{\kappa(X), \kappa(\tilde{X})\}$.

Proof. The proof follows from [52], Theorem 3. ■

Next, we derive the upper bound on the sensitivity of the system.

Theorem 3.3.8. ***(Sensitivity bound)*** *The sensitivity* $\Delta(k)$ *in* (3.33) *can be upper bounded as*

$$\Delta(k) \leq \bar{\Delta}(k) \triangleq \alpha_1 \alpha_2 e^{\alpha_2} k \rho_{max}^{k-1}, \tag{3.41}$$

where $\alpha_1 = \kappa_{max}^3 n \|C(x(0) + FK_0)\|_1, \quad \alpha_2 = \kappa_{max} T_s \delta \qquad$ *and,*

$\delta = \sup_{P,P':adj(\beta)} \|A_c(P) - A_c(P')\|_2.$

Proof. Let P and P' be two β-adjacent parameters. For the impulse input $d(k)$, we have

$$\begin{aligned}
\|y_P(k) - y_{P'}(k)\|_1 &\overset{(a)}{=} \|C\tilde{A}_d^k(P)(x(0) + FK_0) - C\tilde{A}_d^k(P')(x(0) + FK_0)\|_1 \\
&\overset{(b)}{\leq} \|C(x(0) + FK_0)\|_1 \|\tilde{A}_d^k(P) - \tilde{A}_d^k(P')\|_1 \\
&\overset{(c)}{\leq} \|C(x(0) + FK_0)\|_1 \kappa_{max}^2 n k \rho_{max}^{k-1} \|A_d(P) - A_d(P')\|_2 \\
\overset{(d)}{\leq} \|C(x(0) + FK_0)\|_1 \kappa_{max}^4 n T_s k \rho_{max}^{k-1} &\|A_c(P) - A_c(P')\|_2 e^{\kappa_{max} T_s \|A_c(P) - A_c(P')\|_2}
\end{aligned}$$

where (a) follows from (3.38) (b) follows from the submultiplicative property of matrix norm, (c) follows from assumption **A2**, lemma 3.3.6, definitions of κ_{max} and ρ_{max} in (3.39) and (3.40) and the fact that $\tilde{A}_d(P) - \tilde{A}_d(P') = A_d(P) - A_d(P')$ and (d) follows from lemma 3.3.7, and (3.39). The theorem then follows using the definition of δ. ■

The sensitivity bound in (3.41) can be used to determine the privacy noise level, as shown in the next corollary.

Corollary 3.3.9. ***(Noise design using sensitivity bound)*** *The mechanism* $\mathcal{M}$ *in (3.23) is* ϵ*-differentially private upto time* T *if* $w(k)$ *is white Laplacian with*

$$w(k) \sim Lap(0, c_k)^p \quad and \quad \sum_{k=0}^{T} \frac{\bar{\Delta}(k)}{c_k} \leq \epsilon. \tag{3.42}$$

Proof. It can be easily verified that (3.42) implies that the conditions of theorem 3.3.2 are satisfied and the mechanism $\mathcal{M}$ is ϵ-differentially private. ■

Observe that the sensitivity bound $\bar{\Delta}(k)$ in (3.41) decays exponentially for large k. This is a direct consequence of assumption **A10**. Note that without assumption **A10**, the sensitivity (and the sensitivity bound) would remain constant asymptotically. As a result, the noise level required to ensure DP would increase with T (see (3.42)). Clearly, such unbounded noise in the system is undesirable. Utilizing the decaying behavior of the sensitivity bound, we next show that DP can be guaranteed using an exponentially decaying noise.

Lemma 3.3.10. ***(DP through decaying noise)*** *The mechanism $\mathcal{M}$ in (3.23) is ϵ-differentially private upto time T if $w(k)$ is white Laplacian with the distribution $w(k) \sim Lap(0, ck\gamma^k)^n$ with $0 < \rho_{max} < \gamma < 1$ and*

$$c \geq \underline{c} \triangleq \frac{\alpha_1 \alpha_2 e^{\alpha_2}}{\epsilon \rho_{max}} \frac{1 - (\frac{\rho_{max}}{\gamma})^{T+1}}{1 - \frac{\rho_{max}}{\gamma}}. \tag{3.43}$$

Proof. It can be easily verified that (3.43) implies that the condition in (3.42) is satisfied and the mechanism is ϵ-differentially private ■

Note that the lower bound $\underline{c}$ in (3.43) is bounded for all T. Thus, the privacy noise level also remains bounded.

In order to compute the upper bound given in (3.41), we need to further characterize the sensitivity δ from the private parameters to the continuous time closed loop state matrix, as contained in the constant α_2 (see theorem 3.3.8). This sensitivity depends on the structure of $A_c(P)$, and is specific to the application for which privacy is being designed. Next, we characterize this quantity for the two examples considered earlier in section 3.2: second-order consensus and LQR control.

3.3.2 Sensitivity for Second-order Consensus

Consider the consensus example in (3.15). Since L is symmetric and $L\mathbf{1}_N = \mathbf{0}_N$, we have $\nu_1(A_d(L)) = \nu_0(A_c(L)) = \gamma_1 \begin{bmatrix} \mathbf{1}_N \\ \mathbf{0}_N \end{bmatrix}$ and $\tilde{\nu}_1(A_d(L)) = \tilde{\nu}_0(A_c(L)) = \gamma_2 \begin{bmatrix} \mathbf{1}_N^T D & \mathbf{1}_N^T M \end{bmatrix}$ where $\gamma_1 \in \mathbb{R}$ and $\gamma_2 \in \mathbb{R}$ are some scalars. Using (3.35), the steady state value for consensus can be obtained as

$$\bar{x} = \frac{1}{\sum_{i=1}^{N} d_i} \begin{bmatrix} \mathbf{1}_N \\ \mathbf{0}_N \end{bmatrix} \begin{bmatrix} \mathbf{1}_N^T D & \mathbf{1}_N^T M \end{bmatrix} (x(0) + FK_0). \tag{3.44}$$

It can be verified that $\bar{x}$ in (3.44) satisfies the conditions in definition 3.2.1 and thus, the agents achieve consensus. Further, observe that $\bar{x}$ does not depend on matrix L, and assumption **A10** is satisfied.

Lemma 3.3.11. ***(Sensitivity bound for consensus)*** *The sensitivity δ for the second-order consensus in* (3.15) *is upper bounded by*

$$\delta_{CONS} \leq \bar{\delta}_{CONS} \triangleq \|M^{-1}\|_2 \beta.$$

Proof. Using the structure of $A_c(L)$ in (3.15), we have

$$\|A_c(L) - A_c(L')\|_2 = \|M^{-1}(L - L')\|_2 \overset{(a)}{\leq} \|M^{-1}\|_2 \beta,$$

where (a) follows from submultiplicative property of the norm and the fact that L and L' are adjacent. ■

3.3.3 Sensitivity for LQR Control

We first define the following quantity which captures the minimum distance of the poles of the system from the imaginary axis

$$\mu_{min} \triangleq \inf_{P\in\mathcal{P}, i\in\{1,2,\cdots,n\}} |Re(\lambda_i(A_c(P)))|. \tag{3.45}$$

Further, we define separation of a $n \times n$ matrix A as

$$sep(A) = \min\{\|SA + A^T S\|_2 \colon S = S^T \in \mathbb{R}^{n\times n}, \|S\|_2 = 1\} \tag{3.46}$$

Next, we present a known result regarding the perturbation of the Riccati equation.

Lemma 3.3.12. ***(Perturbation of Riccati equation [53])*** *Consider two Riccati equations of the form* (3.20) *that differ only w.r.t. the private parameters, given as* Q *and* Q'*.*

$$\textit{If} \quad \|Q - Q'\|_2 < \frac{sep^2(A_c(Q))}{4\|BR^{-1}B^T\|_2}, \quad \textit{then,} \tag{3.47}$$

$$\|V(Q) - V(Q')\|_2 \leq \frac{2\|Q - Q'\|_2}{sep(A_c(Q))}. \tag{3.48}$$

Proof. The proof follows from [53], Theorem 2.1. ■

Next, we present the sensitivity bound for the LQR problem.

Lemma 3.3.13. ***(Sensitivity bound for LQR control)*** *If*

$$\beta < \frac{\mu_{min}^2}{\kappa_{max}^4 \|BR^{-1}B^T\|_2}, \tag{3.49}$$

then the sensitivity δ *for the LQR control is upper bounded by*

$$\delta_{LQR} \leq \frac{\|BR^{-1}B^T\|_2 \kappa_{max}^2 \beta}{\mu_{min}} < \frac{\mu_{min}}{\kappa_{max}^2}. \tag{3.50}$$

Proof. From Theorem 3.2 in [53], and the definitions of κ_{max} (in (3.39)) and μ_{min} (in (3.45)), we have

$$sep(A_c(Q)) \geq \frac{2\mu_{min}}{\kappa_{max}^2} \quad \forall \quad Q \in \mathcal{P}. \tag{3.51}$$

Further, since Q and Q' are β - adjacent, we have

$$\|Q - Q'\|_2 \leq \beta < \frac{\mu_{min}^2}{\kappa_{max}^4 \|BR^{-1}B^T\|_2} \leq \frac{sep^2(A_c(Q))}{4\|BR^{-1}B^T\|_2},$$

and thus, the condition (3.47) in lemma 3.3.12 is satisfied. Next,

$$\begin{aligned} \|A_c(Q) - A_c(Q^{'})\|_2 &= \|BR^{-1}B^T(V(Q) - V(Q'))\|_2 \\ &\overset{(a)}{\leq} \|BR^{-1}B^T\|_2 \frac{2\|Q - Q'\|_2}{sep(A_c(Q))}, \\ &\overset{(b)}{\leq} \frac{\|BR^{-1}B^T\|_2 \kappa_{max}^2 \beta}{\mu_{min}}, \end{aligned}$$

where (a) follows from the submultiplicative property of the norm and lemma 3.3.12, and (b) follows from the the β-adjacency of Q and Q' and (3.51). The last inequality in (3.50) follows from the condition on β given in (3.49). ■

3.4 Numerical Illustration

In this section, we present numerical simulations for the second-order consensus example to analyze the effect of the privacy noise on the parameter and eigenvalue identification procedures. We consider a connected consensus network with $N = 3$

agents for which the Laplacian matrix is given as

$$L = \begin{bmatrix} 10 & -6 & -4 \\ -6 & 9.5 & -3.5 \\ -4 & -3.5 & 7.5 \end{bmatrix}.$$

The other system parameters are given as $M = diag(4, 4, 6)$, $D = diag(1, 2, 1)$, $\bar{F} = diag(1, 0, 0)$, $x(0) = \mathbf{0}_6$ and the disturbance is an impulse with $K_0 = 10^5\mathbf{1}_3$. The coefficients of the characteristic polynomial of A_c are $a_c = [1, 0.92, 6.38, 3.61, 9.07, 2.46, 0]^T$ with $||a_c||_2 = 11.99$. For the above system, $\bar{T}_s = 1.54$ (see (3.11)) and the agents sample the measurements with $T_s = 1$ to satisfy assumption **A7**. The agents implement the privacy mechanism $\mathcal{M}$ in (3.23) and add Laplacian noise to the measurements before sending them to the control center.

To design the privacy noise, we use the sensitivity upper bounds obtained in theorem 3.3.8 and lemma 3.3.11. For the given consensus network, $\kappa(X(L)) = 7.45$ (see (3.39)) and $\rho(\tilde{A}_d(L)) = 0.86$ (see (3.36)). Note that in order to compute κ_{max} in (3.39) and ρ_{max} in (3.40), we need a characterization of the set $\mathcal{P}$ of all possible private parameters. Without loss of generality, we avoid this explicit characterization and choose $\kappa_{max} = 8$ and $\rho_{max} = 0.9$. In scenarios where the set $\mathcal{P}$ is given explicitly, κ_{max} and ρ_{max} can be easily computed. We choose the adjacency parameter $\beta = 2$ (see definition 3.2.5). With the given choice of parameters, we have $\alpha_1 = 7.68 \times 10^6$ and $\alpha_2 = 4$ (see (3.41) and lemma 3.3.11). We design the decaying Laplacian noise using lemma 3.3.10. We choose the noise decay factor $\gamma = 0.99$ and simulate the system for $T = 1000$ time steps. Using these parameter values, the noise lower bound in (3.43) becomes $\underline{c} = 2.28 \times 10^{11}\epsilon^{-1}$ and we choose $c = \underline{c}$ for adding the privacy noise. We select the range of privacy noise level $c = [0, 1]$. This represents an asymptotically high SNR regime (> 58dB) and asymptotically low privacy level ($\epsilon^{-1} < 4.4 \times 10^{-12}$). This high SNR regime guarantees numerical stability of the identification methods

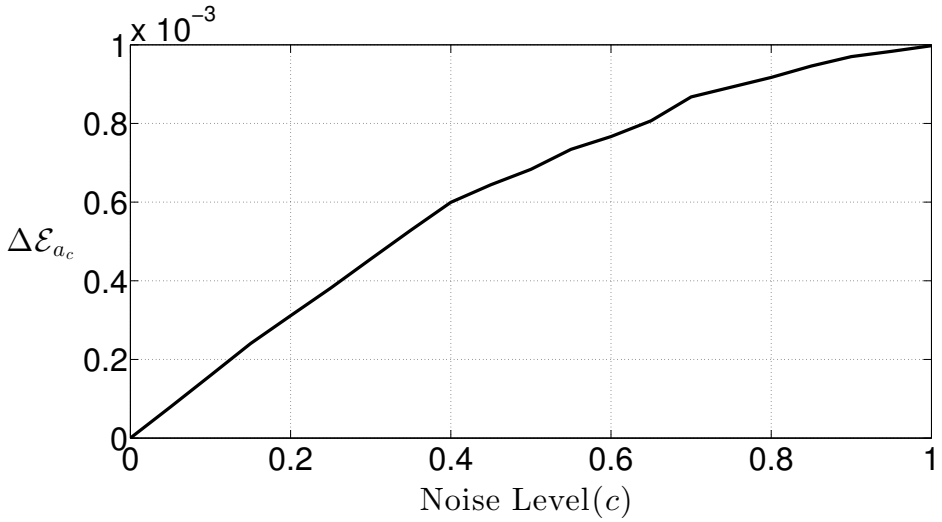


Figure 3.2. Eigenvalue identification performance

in Matlab.

Figure 3.2 characterizes the effect of privacy noise on the eigenvalue identification procedure at the control center. The identification is performed in Matlab using the `iv4` function in its system identification toolbox, which implements a four stage IV identification method [49]. The eigenvalue identification performance is characterized in terms of expected relative coefficient error as $\Delta\mathcal{E}_{a_c} \triangleq \frac{\mathcal{E}_{a_c}}{||a_c||_2}$ (see (3.32)). We approximate the expected error by averaging the error values over 1000 iterations for each noise level. We can observe that the identification error increases with the privacy noise (privacy level).

Figure 3.3 shows the effect of privacy noise on the parameter identification procedure. In second-order consensus networks, it is always not feasible for an intruder to gain information about the agents' inertias and damping values. We assume that the intruder does not have access to M and D matrices and thus, it can only identify $\bar{L} \triangleq M^{-1}L$ (instead of topology L) using the measurements (see remark 3.2.3). Therefore, it extracts $\hat{\bar{L}}_T$ from the estimate $\hat{A}_{c,T}(\hat{\bar{L}}_T)$ (see structure of $A_c(L)$ in (3.15))

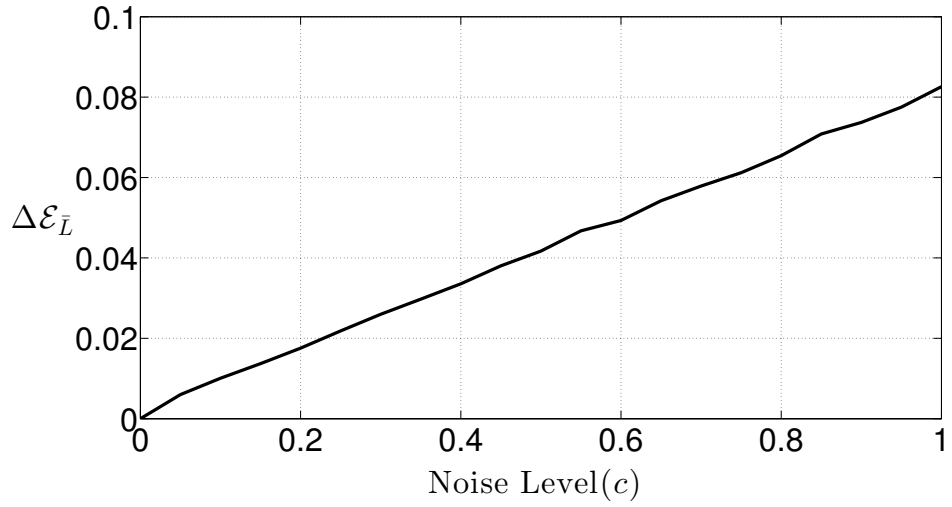


Figure 3.3. Parameter identification performance

in step **P3** of the parameter identification procedure. The relative parameter identification error is defined as $\Delta\mathcal{E}_{\bar{L}} \triangleq \frac{\mathbb{E}\left[\left\|\hat{\bar{L}}_T - M^{-1}L\right\|_F\right]}{\|M^{-1}L\|_F}$. As seen in fig. 3.3, the identification error is close to 0 for $c = 0$. Thus, the private topology is accurately identified if there is no privacy mechanism in the system, leading to a privacy breach. As the privacy noise level increases, the identification error also increases making it more difficult for the intruder to identify the topology. Comparing figures 3.2 and 3.3, we can observe that relative error of eigenvalue identification is of order of magnitude 2 less than that of the parameter identification. The privacy noise causes significantly larger performance degradation for the intruder as compared to the control center. This implies that our mechanism can provide privacy without significant performance loss.

3.5 Summary

In this chapter, we develop a noise adding differential privacy mechanism to protect the privacy of the sensitive parameters associated with the dynamics of the agents

in LTI multi-agent systems. We derive an upper bound to the sensitivity of the system and use it to design the privacy noise. We present two concrete applications of our privacy framework: the second-order consensus network and LQR control. Our numerical simulations for the consensus problem show that for the asymptotic regime of low privacy and high SNR, the performance degradation suffered by the intruder due to privacy noise is significantly higher when compared to that of the control center.

CHAPTER 4

PRIVACY VS COOPERATION IN MULTI-AGENT SYSTEMS

4.1 Background

Several problems in control theory, optimization and robotics require cooperation among multiple agents. Prototypical examples include consensus [5], flocking [54], formation control [6], coverage control [55], and distributed optimization [7, 8]. Typically, cooperation requires information exchange, which may lead to leakage of private information with undesirable consequences. For example, in smart metering systems where users send their energy consumption data for power network optimization, the data can reveal information about their personal lives, such as daily schedules etc. [56], [57]. In autonomous vehicle scenarios where the vehicles communicate and share their position/velocity data, it can reveal their past or future travel plans. Even if the agents are trustworthy, a possibility exists for an intruder to eavesdrop on the messages exchanged among the agents and gather their private information. Privacy concerns have recently been addressed by introducing dedicated mechanisms as part of the cooperation protocol [32, 33, 35, 36, 39–41]. In most of these privacy mechanisms, each agent deliberately adds noise to the data communicated to other agents, thereby preventing them (or an eavesdropper) from recovering the sensitive data of individual agents by accurately processing the distorted messages.

Most of the recently proposed privacy mechanisms are based on techniques originally developed to protect static databases and usually degrade the performance when applied to dynamical systems. For instance, in applications involving wide

area control of power grids, adding noise to the data may result in loss of stability. In distributed systems, if agents use noisy information from other agents to update their own state, the resulting behavior differs from the desired one. Furthermore, both due to the dynamical nature of the system and the fact that information originating from one agent may traverse to another agent through multiple paths in distributed systems, the noise introduced by the agents at one time step can adversely affect the state evolution of a multi-agent system multiple times in the future. Notice that the second reason arises because of the cooperative nature of the system where one agent uses information from other agents. Thus, intuitively, there should be a tradeoff between the 'cooperation level' and performance in a distributed system when the agents are trying to keep their information private. If the noise level introduced to maintain privacy is too high, then cooperation might even impede the system functionality. On the other hand, if the agents do not transmit any information to each other, perfect privacy is achieved, at the expense of the benefits of cooperation. In this dissertation, we address an outstanding and important question whether cooperation leads to improved performance in the presence of a privacy mechanism, and whether a fundamental tradeoff exists between the two.

We consider a scenario where the objective of the agents is to cooperatively minimize a common quadratic cost function of their states by sharing their state information among each other. Several problems such as consensus and formation control fall into this class. In addition, the agents wish to keep their states private during this process. We propose a noise adding privacy mechanism for the agents to keep their states private over time. Note that we focus on *privacy of the agents' state trajectory*, rather than the state at any specific time (as in [33]). We adopt the Differential Privacy (DP) framework originally proposed by [9], and later extended to dynamical systems in [35]. We characterize the noise level ensuring a desired level of privacy.

Next, we introduce a method for the agents to adapt their cooperation level in response to the privacy noise. In many scenarios, in addition to optimizing a global cost, the agents also have individual goals that do not require cooperation. For example, in intelligent transport systems, vehicles cooperate to reduce road congestion while each of them also wants to reduce its own travel time (ex. congestion game in [58]). Individual objectives also exist in multi-objective optimization problems, wherein multiple conflicting goals are considered and in optimization problems with separable cost functions. When the agents wish to remain private (by sharing noisy data), it is intuitive that they should cooperate less and focus more on their individual goals. Thus, the cooperation level can be characterized based on whether the agents are willing to cooperatively minimize the global cost, or they want to selfishly minimize their individual costs. We formalize this notion by defining a new cost that is a convex combination of the global and individual costs, wherein the weighing factor represents the *cooperation level*. We then characterize the combined effect of cooperation level and privacy noise on the system performance.

Several secure multiparty computation schemes exist in literature which compute a function of agents' variables while keeping them private [59, 60]. However, in these schemes, there always exists a possibility that some agent(s) obtain auxiliary information and use it to infer other agents' private variables. Moreover, majority of agents can collude to infer the remaining agents' sensitive information. To address these issues, we use the differential privacy framework in this work. DP abstracts away from any auxiliary information that the agents might have and it is also resilient to post processing of data [10, 35].

As mentioned in subsection 3.1.1 many recent studies have proposed privacy mechanisms for multi-agent systems in the context of consensus [36, 38] and optimization problems [32, 33, 39–41]. All of these works develop privacy mechanisms and analyze their effect of the on the system performance in terms of sub-optimality, accuracy,

convergence etc. In contrast, we also develop similar privacy mechanism but address fundamentally different questions such as: (i) How does the system performance change if the agents vary the amount of cooperation among each other?, (ii) for a higher privacy level, is it beneficial for the agents to reduce cooperation? Our analysis highlights this previously unidentified tradeoff in multi-agent systems.

The cooperation level in our framework can be viewed as a weighting factor for the noisy state information received from the neighbors and used to update the agents states. Related works include [61], in which the authors analyze consensus in the presence of noise, and show that almost sure convergence can be guaranteed by using time decaying weighing factor in the updates. In [62], the authors find the optimal edge weights for the consensus problem that minimize the expected deviation among the agents. These works are specifically developed for the consensus algorithm, and may not work for other problems. In contrast, to elucidate the relation between the cooperation and privacy levels in multi-agent systems, we develop techniques that are applicable to more general quadratic optimization problems and not only limited to consensus.

4.2 Problem Formulation

In this section we present our multi-agent cooperative optimization problem and characterize its solution. Additionally, we describe a noise adding mechanism that preserves the privacy of the agents' states over time.

4.2.1 Distributed Quadratic Optimization Framework

Consider a distributed system with a set of $N \geq 2$ agents denoted by $\mathcal{N} = \{1, 2, \cdots, N\}$. The agents collectively aim to minimize a common objective that is

given by

$$\mathbf{P}: \qquad \min_{x} \quad J_{co}(x) = \frac{1}{2}x^T Q x + r^T x + s, \tag{4.1}$$

where the vector $x = [x_1, x_2, \cdots, x_N]^T \in \mathbb{R}^N$ denotes the states of all agents. Further, Q is a non-zero $N \times N$ real matrix, $r \in \mathbb{R}^N$ and $s \in \mathbb{R}$. Let q_{ij} and r_i denote the entries of Q and r, respectively. Note that the states of the agents are coupled with each other via the quadratic term $\frac{1}{2}x^T Q x$ in the cost function. Specifically, we say that the agents i and j are uncoupled if both q_{ij} and q_{ji} are zero, and that they are coupled otherwise. Let $\mathcal{N}_i$ denote the *neighbor set* or the set of agents whose states are coupled to the state of agent i, and let $N_i = |\mathcal{N}_i|$. We place the following assumptions on the cost function $J_{co}(x)$:

A.1) Q is symmetric and positive semi-definite. Further, if 0 is an eigenvalue of Q, then: (i) its algebraic multiplicity is 1, and (ii) $r = \mathbf{0}_N$ in (4.1).

A.2) Each row(or column) of $Q - \text{diag}(Q)$ has atleast one non-zero entry.

Assumption **A.2** implies that there is no uncoupled agent in the system, that is, $N_i \neq 0$ for each $i \in \mathcal{N}$. This assumption is not restrictive because a system with n uncoupled agents can be studied via a reduced system with $N - n$ coupled agents. Assumption **A.1** implies that the minimization problem **P** is convex and admits a finite (but not necessarily unique) optimum. Let the set of all the optimum solutions of **P** be denoted by $\mathcal{X}^*$. An optimum $x^* \in \mathcal{X}^*$ can be achieved by the agents with a distributed, iterative gradient descent algorithm. In such an algorithm, the update rule of agent i is

$$x_i(k+1) = x_i(k) - \gamma_1 \frac{\partial}{\partial x_i} J_{co}(x(k)) = x_i(k) - \gamma_1 \left(q_{ii}x_i(k) + \sum_{j \in \mathcal{N}_i} q_{ij}x_j(k) + r_i \right), \tag{4.2}$$

where $\gamma_1 > 0$ is the step size and $x_i(0)$ is the initial state of agent i. As evident from the above iteration, agent i requires state information $x_{\mathcal{N}_i}$ from all its neighbors for its own state update. We assume that the agents can communicate their state information to each other without any distortion. The gradient descent algorithm for all agents can be collectively represented as

$$\mathbf{S_1}: \qquad x(k+1) = x(k) - \gamma_1(Qx(k) + r) = (I_N - \gamma_1 Q)x(k) - \gamma_1 r \triangleq A_1 x(k) + b_1, \tag{4.3}$$

where $A_1 = I_N - \gamma_1 Q$, $b_1 = -\gamma_1 r$ and initial state $x(0) = [x_1(0), x_2(0), \cdots, x_N(0)]^T$. Since the cost gradient is linear, algorithm $\mathbf{S_1}$ can be represented as a discrete time invariant linear system. The optimum of problem $\mathbf{P}$ is given by the steady state solution of $\mathbf{S_1}$. Next, we state the condition under which the steady state solution exists.

Lemma 4.2.1. ***(Convergence of algorithm*** $\mathbf{S_1}$***)*** *Let* $\gamma_1 < 2\rho(Q)^{-1}$. *Then, the algorithm* $\mathbf{S_1}$ *in* (4.3) *converges asymptotically, that is* $\lim_{k\to\infty} x(k) = x^*$ *for an* $x^* \in \mathcal{X}^*$.

Proof. First, assume $Q > 0$. For $i = 1, 2, \cdots, N$ we have $0 < \gamma_1\lambda_i(Q) \leq \gamma_1\lambda_1(Q) < 2$. Since $\lambda_i(A_1) = 1 - \gamma_1\lambda_i(Q)$, the above condition is equivalent to $-1 < \lambda_i(A_1) < 1$. Thus, all eigenvalues of A_1 lie inside the unit circle and a steady state solution of (4.3) is achieved. Assume now that Q has a 0 eigenvalue. Then, by assumption **A.1**, $b_1 = 0$ and A_1 has a single eigenvalue at 1 and all other eigenvalues lie inside the unit circle. Thus, the linear system in (4.3) is marginally stable and a finite steady state solution is achieved. ■

Let the steady state of (4.3) be denoted by m_1. Then,

$$m_1 = A_1 m_1 + b_1, \tag{4.4}$$

and the optimum cost achieved by the agents is given by

$$J_{co}^{*} = \frac{1}{2} m_1^T Q m_1 + r^T m_1 + s. \tag{4.5}$$

Remark 4.2.2. ***(Examples)*** A number of problems fit into our quadratic cost framework, including consensus and formation control. We will discuss the consensus example in detail in Section 4.4. We also study the 1D centroidal Voronoi tessellation problem in which the agents implement a linear algorithm to minimize a convex cubic cost. This shows that our framework can be extended to problems involving convex non-quadratic costs that can be optimized by a linear algorithm. □

4.2.2 Privacy Mechanism

In the cooperative algorithm $\mathbf{S_1}$, the agents update their state upon communicating the state information with their neighbors. Thus, algorithm $\mathbf{S_1}$ is not private and in fact, an intruder may reconstruct the state trajectories of the agents with access to only a few messages communicated by the agents. To ensure privacy, we consider the Differential Privacy (DP) mechanism that protects the state of the agents over time, where each agent adds an artificial random noise to its state before communicating it with other agents. The noise ensures that "*any two different instances*" of the communicated state trajectories are "*statistically not very different*", which prevents the intruder from accurately obtaining the actual state information of the agents; thus, maintaining their privacy.

Remark 4.2.3. ***(Privacy of state trajectory)*** Note that different instances of the state trajectory arise from different initial states $x(0)$. However, in addition to the initial state, agents wish to keep their positions/velocities private at all times because accurate state information at any time instant can potentially reveal the complete future state trajectory. □

The noisy DP mechanism can be written as

$$\mathcal{M}: \qquad \tilde{x}_i(k) = x_i(k) + n_i(k), \tag{4.6}$$

where $\tilde{x}_i(k)$ denotes the state communicated to the neighbors of agent i and $n_i(k)$ is the random privacy noise. Let $n(k) = [n_1(k), n_2(k), \cdots, n_N(k)]^T$. We adopt the differential privacy framework developed in [35] to design the noise that ensures privacy of the state trajectories. We begin with the definition of adjacency.

Definition 4.2.4. ***(Adjacency)*** Given a finite $\beta \geq 0$, two state trajectories $x[0:\infty]$ and $x'[0:\infty]$ are β-adjacent (denoted by $adj(\beta)$) if

$$\|x[0:\infty] - x'[0:\infty]\| \leq \beta. \tag{4.7}$$

It should be noticed that in the classic definitions of DP for static databases [9] and for dynamical systems [35], adjacency is defined with respect to the change of trajectory of one agent only, while keeping the trajectories of other agents unchanged. In contrast, our definition of adjacency allows simultaneous changes in the trajectories of one or more agents.

Remark 4.2.5. ***(Common steady state value)*** The adjacency definition in (4.7) implicitly requires that the two instances of the state trajectories (resulting from two different initial conditions) vary only for transient periods and have a common steady state value. This holds true if $Q > 0$, since it is easy to observe (see (4.4)) that the steady state value does not depend on the initial condition $x(0)$. However, when $Q \geq 0$ with a single eigenvalue at 0 (see **A.1**), then the steady state value might depend on the initial condition. Let $\mathcal{X}_0^m$ denote the set of all initial conditions that result in a steady state value of m. Then, the privacy mechanism guarantees DP only among those trajectories that result from initial conditions contained in the set

$\mathcal{X}_0^m$. □

Let $\tilde{x}[0:\infty]$ and $\tilde{x}'[0:\infty]$ denote the corresponding noisy communicated state trajectories. Note that $\tilde{x}[0:T] \in \mathbb{R}^{N(T+1)}$ and let $\mathcal{R}^{N(T+1)}$ denote the $\sigma - algebra$ generated by it. Next, we provide the definition of differential privacy.

Definition 4.2.6. ***(Differential privacy)*** The mechanism $\mathcal{M}$ in (4.6) is (ϵ, δ)-differentially private if for any two β-adjacent trajectories $x[0:\infty]$ and $x'[0:\infty]$ and for all $S \subset \mathcal{R}^{N(T+1)}$ and for all $T \geq 0$ it holds

$$\mathbb{P}[\tilde{x}[0:T] \in S] \leq e^{\epsilon}\mathbb{P}[\tilde{x}'[0:T] \in S] + \delta, \tag{4.8}$$

where $\epsilon > 0$ and $0 < \delta < 0.5$ are privacy parameters. □

Definition 4.2.6 implies that for any two beta adjacent trajectories, the statistics of the corresponding noisy communicated state trajectories differ only within a multiplicative factor of e^{ϵ} and an additive factor of δ. A standard way to guarantee DP is to choose an i.i.d. Gaussian noise and scale its variance according to the adjacency parameter β, as stated in the next lemma.

Lemma 4.2.7. ***(Ensuring differential privacy)*** *The mechanism $\mathcal{M}$ in (4.6) is (ϵ, δ)-differentially private if $n(k)$ is white Gaussian noise with distribution $n(k) \sim \mathbf{N}(0, \sigma^2 I_N)$, where*

$$\sigma \geq \frac{\beta}{2\epsilon}(K + \sqrt{(K^2 + 2\epsilon)}), \text{ and } K = \mathcal{Q}^{-1}(\delta).$$

Proof. See Theorem 3 in [35]. Since the quantity that needs to be protected and that is communicated is same (i.e. the state trajectory), the sensitivity is trivially upper bounded by the adjacency parameter β. ■

In Lemma 4.2.7, the relation between σ and the privacy parameters (ϵ, δ) implies that the noise variance is a monotonically decreasing function of ϵ and δ. Also, note

from the definition of DP in (4.8) that a smaller value of ϵ and δ implies larger privacy of the agents. Thus, the noise variance σ can be treated as synonymous to the privacy level of the system, and for the ease of presentation, we present our results directly in terms of noise level σ (instead of the privacy parameters ϵ and δ).

In the presence of privacy noise, the evolution of the algorithm $\mathbf{S_1}$ in (4.3) is modified as

$$\mathbf{S_1^{priv}}: \qquad x(k+1) = A_1 x(k) + b_1 + H_1 n(k), \tag{4.9}$$

where $H_1 \triangleq A_1 - \text{diag}(A_1)$ is obtained by replacing the diagonal elements of A_1 with zero entries, since only non-diagonal entries in A_1 represent coupling between the agents. Note that $H_1 = -\gamma_1 \tilde{Q}$ where $\tilde{Q} = Q - \text{diag}(Q)$.

4.2.3 Performance Degradation due to the Privacy Mechanism

The noise introduced by the privacy mechanism makes the states of the agents private. However, it also adversely affects the system performance. Due to the stochastic nature of algorithm $\mathbf{S_1^{priv}}$, we analyze the system performance by calculating the expected cost achieved by the agents in the presence of noise. The algorithm $\mathbf{S_1^{priv}}$ in (4.9) can be viewed as a linear system driven by a constant input and Gaussian privacy noise. Thus, the state of the agents at each time instant has a normal distribution, denoted by $x(k) \sim \mathbf{N}(m_1(k), P_1(k))$. The evolution of the mean and the covariance of the states of the agents is given by

$$m_1(k+1) = A_1 m_1(k) + b_1, \quad \text{and} \tag{4.10}$$

$$P_1(k+1) = A_1 P_1(k) A_1^T + \sigma^2 H_1 H_1^T, \tag{4.11}$$

with $m_1(0) = x(0)$ and $P_1(0) = 0$. If A_1 is stable (i.e. $Q > 0$ in (4.1)), then the mean and covariance reach a finite steady state value, denoted by m_1 and P_1, respectively. Note that we have overloaded the notation of m_1 with the noiseless case presented in

section 4.2.1 since the steady state solution of (4.3) and (4.10) are the same. Thus, the mean m_1 satisfies (4.4) and the covariance P_1 satisfies the following Lyapunov equation:

$$P_1 = A_1 P_1 A_1^T + \sigma^2 H_1 H_1^T. \tag{4.12}$$

If A_1 is marginally stable (that is, Q in (4.1) has a single eigenvalue at 0, see assumption **A.1**), then m_1 exists and is finite. Yet, the covariance P_1 may become unbounded, and the system becomes unstable in the stochastic sense. We now present the performance result.

Theorem 4.2.8. ***(Performance in the presence of privacy noise)*** *Assume $Q > 0$. At steady state, the expected cost achieved by the agents implementing the algorithm $\mathbf{S_1^{priv}}$ in (4.9) is given by*

$$J_{co}^*(\sigma) \triangleq \mathbb{E}[J_{co}(x)] = \frac{1}{2} tr(QP_1) + \frac{1}{2} m_1^T Q m_1 + r^T m_1 + s. \tag{4.13}$$

where the expectation $\mathbb{E}[.]$ is taken w.r.t the privacy noise and P_1 depends on σ.

Proof. Since $Q > 0$, its Cholesky decomposition exists, denoted by $Q = L^T L$. Further, let x denote the random steady state and let $y = Lx$. If $x \sim \mathbf{N}(m_1, P_1)$, then $y \sim \mathbf{N}(Lm_1, LP_1L^T)$. We have,

$$\begin{aligned}
\mathbb{E}[J_{co}(x)] &= \mathbb{E}[\frac{1}{2} x^T Q x + r^T x + s] \\
&= \frac{1}{2}\mathbb{E}[y^T y] + r^T m_1 + s \\
&= \frac{1}{2} tr(\mathbb{E}[yy^T]) + r^T m_1 + s \\
&= \frac{1}{2} tr(LP_1L^T + Lm_1(Lm_1)^T) + r^T m_1 + s \\
&= \frac{1}{2}(tr(QP_1) + m_1^T Q m_1) + r^T m_1 + s,
\end{aligned}$$

where we have used the fact that $tr(.)$ is a linear and invariant under cyclic permutations. ■

The performance degradation due to the privacy noise is obtained by comparing (4.5) and (4.13), and is given by

$$J^*_{co}(\sigma) - J^*_{co} = \frac{1}{2} tr(QP_1),$$

which increases with the noise level σ. Because the agents share noisy state information, full cooperation and use of the distorted information for the algorithm updates affect the agents performance. In the other extreme case, if the agents forgo cooperation, then they will be completely private, since no state information will be exchanged among them, but will probably not achieve the optimum of problem $\mathbf{P}$. Thus, a mechanism is needed for the agents to adapt their level of cooperation to maximize their performance in the presence of privacy noise. In the next section, we define a notion of *cooperative level*, and present modified optimization algorithms that incorporate the cooperation level as a parameter.

4.3 Cooperation Level in Multi-agent Systems

In this section, we introduce and motivate our notion of cooperation level in private multi-agent systems. We calculate the expected cost achieved by the agents for a particular level of cooperation and privacy noise and use it to characterize the optimum cooperation level.

4.3.1 A Notion of Cooperation Level

Agents cooperate to implement algorithm $\mathbf{S_1}$. However, as discussed above, full cooperation may not be optimal if agents also want to preserve privacy. To formalize this, we introduce a cooperation parameter in the algorithm $\mathbf{S_1}$. In many scenarios, in addition to minimizing the system cost J_{co}, the agents also have individual goals for which no cooperation is required. As explained in the background, such conflicting

goals are ubiquitous in optimization and game theory. We formalize the individual agent goals by the following cost function

$$J_{nco}(x) = \frac{1}{2}x^T\bar{Q}x + \bar{r}^T x + \bar{s}, \tag{4.14}$$

and assume that

A.3) $\bar{Q}$ is diagonal and positive definite.

Assumption **A.3** implies that the states of all the agents are decoupled in $J_{nco}(x)$. As a result, no cooperation is required to minimize the decoupled cost function $J_{nco}(x)$.

We utilize these individual agent goals to introduce *cooperation level* in our framework. The costs $J_{co}(x)$ and $J_{nco}(x)$ represent two extremes on the cooperation scale. To minimize the former, full cooperation is necessary among the agents, while no cooperation is required for the latter. When the agents wish to keep private, it is prudent for them to give more weight to their individual goals as compared to the system goal. Following this reasoning and to capture the intermediate cooperation behavior, we consider a new cost function which is precisely the convex combination of $J_{co}(x)$ and $J_{nco}(x)$:

$$J_\alpha(x) = \alpha J_{co}(x) + (1-\alpha)J_{nco}(x), \tag{4.15}$$

where the parameter $\alpha \in [0, 1]$ is the agents' *cooperation level*. Note that the new cost $J_\alpha(x)$ is convex due to convexity of $J_{co}(x)$ and $J_{nco}(x)$. The gradient descent algorithm that minimizes J_α inherently introduces the cooperation level in our framework and

in presence of privacy noise can be written as

$$\begin{aligned}
\mathbf{S}_\alpha^{\mathbf{priv}}: \qquad x(k+1) &= x(k) - \gamma\left(\alpha(Qx(k)+r) + (1-\alpha)(\bar{Q}x(k)+\bar{r})\right) + H_\alpha n(k) \\
&\triangleq A_\alpha x(k) + b_\alpha + H_\alpha n(k), \quad \text{where} \qquad (4.16) \\
Q_\alpha &= \alpha Q + (1-\alpha)\bar{Q}, \quad A_\alpha = I_N - \gamma Q_\alpha, \\
b_\alpha &= -\gamma r_\alpha, \quad r_\alpha = \alpha r + (1-\alpha)\bar{r}, \\
H_\alpha &\triangleq A_\alpha - \mathrm{diag}(A_\alpha) = -\gamma\alpha(Q - \mathrm{diag}(Q)),
\end{aligned}$$

and $\gamma > 0$ is the step size.

Remark 4.3.1. ***(Auxiliary cost function)*** Note that the decoupled cost function J_{nco}, the new cost function J_α and its minimizing algorithm $\mathbf{S}_\alpha^{\mathbf{priv}}$ merely act as a means to introduce the cooperation level in our problem. The goal of the agents is to minimize the global cost J_{co}, and thus we measure the performance the algorithm achieves also in terms of this global cost. □

By varying the cooperation level in $\mathbf{S}_\alpha^{\mathbf{priv}}$, the agents can achieve a range of private solutions of $\mathbf{P}$. In practice, agents should select a cooperation level that maximizes the system performance, as we will show in the next subsection. Notice that agents are still required to exchange their state information for all values of the cooperation level $0 < \alpha \leq 1$, and that the cooperation level determines the weight given by an agent to the information coming from its neighbors.

Remark 4.3.2. ***(Alternative approaches for variable cooperation level)*** Different methods exist to capture the notion of cooperation level. For instance, agents may filter the exchanged measurements to reduce the effect of privacy noise on the performance, and use the filter weights to measure their cooperation level. Yet, our formulation modulates cooperation in a natural and explicit way by balancing the individual and global goals of the agents, and it allows us to directly characterize critical tradeoffs between privacy and performance in multi-agent systems. □

Remark 4.3.3. ***(Selection of the decoupled cost)*** There may be scenarios in which the agents do not have individual goals. In such cases, we can construct an artificial decoupled cost J_{nco} to capture the non-cooperation extreme. Several choices of the matrix $\bar{Q}$ are possible. For instance, $\bar{Q}$ can be chosen to consist of the diagonal elements of Q, or it can be an arbitrary matrix that satisfies assumption **A.3**(see examples in Section 4.4). Thus, our framework is applicable for a wide variety of multi-agent optimization problems. The selection of a decoupled cost that optimizes the agents performance is left as the subject of future research. □

4.3.2 Performance Analysis with Privacy and Cooperation

We now analytically characterize the expected cost for a given privacy and cooperation level. Note that due to assumptions **A.1** and **A.3**, both Q_α and A_α are symmetric. Moreover, since the privacy noise has a normal distribution, the state $x(k)$ in algorithm $\mathbf{S}_\alpha^{\mathbf{priv}}$ is also normal. Let the steady state mean and covariance of $x(k)$ in algorithm $\mathbf{S}_\alpha^{\mathbf{priv}}$ be denoted by m_α and P_α, respectively. Next, we present conditions under which the steady state mean and covariance exist. Note that Lemma 4.2.1 presents such condition for $\alpha = 1$.

Lemma 4.3.4. ***(Convergence of*** S_α^{priv} ***for*** $\alpha \neq 1$***)*** *Let* $\alpha \neq 1$*. Then, the steady state mean and covariance of algorithm* $\mathbf{S}_\alpha^{\mathbf{priv}}$ *exist if*

$$\gamma < \frac{2}{\max\{\rho(Q), \rho(\bar{Q})\}}. \tag{4.17}$$

Proof. The proof is similar to that of lemma 4.2.1 by using the following facts: (i) $Q_\alpha > 0$ for $\alpha \neq 1$, and (ii) from Weyl's inequality [63], $\rho(Q_\alpha) \leq \max\{\rho(Q), \rho(\bar{Q})\}$. ■

Analogous to (4.4) and (4.12), the steady state mean and covariance of $\mathbf{S}_\alpha^{\mathbf{priv}}$

satisfy

$$m_\alpha = A_\alpha m_\alpha + b_\alpha$$

$$\Rightarrow 0 = Q_\alpha m_\alpha + r_\alpha \quad \text{and,} \tag{4.18}$$

$$P_\alpha = A_\alpha P_\alpha A_\alpha^T + \sigma^2 H_\alpha H_\alpha^T. \tag{4.19}$$

A closed form expression of P_α can be written as

$$P_\alpha = \sigma^2\gamma^2\alpha^2 \sum_{k=0}^{\infty} A_\alpha^k (Q - diag(Q))^2 (A_\alpha^T)^k. \tag{4.20}$$

Similarly to (4.13), the cost achieved by algorithm $\mathbf{S}_\alpha^{\mathbf{priv}}$ is

$$J(\alpha, \sigma) = J_{priv}(\alpha, \sigma) + J_{ico}(\alpha) \quad \text{where,} \tag{4.21}$$

$$J_{priv}(\alpha, \sigma) \triangleq \frac{1}{2} tr(QP_\alpha) \quad \text{and,} \tag{4.22}$$

$$J_{ico}(\alpha) \triangleq \frac{1}{2} m_\alpha^T Q m_\alpha + r^T m_\alpha + s. \tag{4.23}$$

Notice that $J_{ico}(\alpha)$ represents the cost achieved by the agents for any *intermediate cooperation* level α in the absence of privacy noise. Further, the cost term $J_{priv}(\alpha, \sigma)$ quantifies the effect of the privacy noise at a given cooperation level since the covariance P_α depends on noise level σ. However, we omit that dependence in the notation for the ease of presentation. Moreover, P_α also depends on the step size γ. We do not analyze this dependence because γ dictates the number of iterations for algorithm $\mathbf{S}_\alpha^{\mathbf{priv}}$ to converge, which is not the primary issue addressed here. Note that the optimum for $\mathbf{P}$ is achieved only when $\alpha = 1$ and $\sigma = 0$ (that is, when the agents fully cooperate and no privacy noise is present). To clarify the notation, $J_{co}^* = J(1, 0)$, $J_{co}^*(\sigma) = J(1, \sigma)$ and $J_{ico}(\alpha) = J(\alpha, 0)$. Also note that the functions J, J_{priv} and J_{ico} are continuous functions in their respective variables.

Intuitively, in the absence of privacy noise, the performance should increase as the agents cooperate more and should equal the best performance when they cooperate fully ($\alpha = 1$). The following lemma proves this fact and justifies our definition of cooperation level.

Lemma 4.3.5. ***(Performance without privacy)*** *The cost $J_{ico}(\alpha)$ in (4.23) is monotonically decreasing for $\alpha \in [0, 1]$.*

Proof. By differentiating $J_{ico}(\alpha)$ with respect to α, we obtain

$$J'_{ico}(\alpha) = (m_\alpha^T Q + r^T) m'_\alpha. \tag{4.24}$$

For $\alpha = 1$, from (4.18) we have, $Qm_1 + r = 0$. Thus, $J'_{ico}(1) = 0$.
For $\alpha \in [0, 1)$, $Q_\alpha > 0$. Thus, Q_α is invertible, and $Q_\alpha^{-1} > 0$. Differentiating (4.18), we get

$$(Q - \bar{Q}) m_\alpha + Q_\alpha m'_\alpha + r - \bar{r} = 0, \tag{4.25}$$

$$\overset{(a)}{\Rightarrow} m'_\alpha = -\frac{Q_\alpha^{-1}(Qm_\alpha + r)}{1 - \alpha}, \tag{4.26}$$

where (a) follows from (4.18). Thus, we have

$$J'_{ico}(\alpha) = -\frac{1}{1 - \alpha}(Qm_\alpha + r)^T Q_\alpha^{-1}(Qm_\alpha + r).$$

and the derivative is non-positive, which completes the proof. ■

Lemma 4.3.5 implies that, in absence of privacy noise, it is beneficial for the agents to cooperate fully. Instead, in the presence of privacy noise, agents can achieve a range of private solutions of $\mathbf{P}$ by varying the cooperation level. In practice, agents should select a cooperation level that minimizes the cost $J(\alpha, \sigma)$. The optimum cooperation

level for the agents for a given level of privacy noise σ is characterized as

$$\alpha^*(\sigma) = \arg\min_{\alpha} J(\alpha, \sigma). \tag{4.27}$$

Remark 4.3.6. ***(Finding the optimum cooperation level)*** The optimum cooperation level $\alpha^*(\sigma)$ can be approximated numerically by discretizing the interval $[0, 1]$ for α, and evaluating the cost $J(\alpha, \sigma)$ at each point. □

Next, we show that under some conditions on cost functions J_{priv} and J_{ico}, we can characterize the behavior of $\alpha^*(\sigma)$.

Theorem 4.3.7. ***(Characterizing the optimum cooperation level)*** *For all $\sigma > 0$, let $J_{priv}(\alpha, \sigma)$ be strictly increasing for all $\alpha \in (0, 1]$. Further, let $J_{priv}(\alpha, \sigma)$ and $J_{ico}(\alpha)$ be strictly convex for $\alpha \in [0, 1)$. Then, $\alpha^*(\sigma)$ is a monotonically decreasing function of σ.*

Proof. Let $'$ denote the derivative or partial derivative w.r.t. α. Using (4.20), we have $J_{priv}(\alpha, \sigma) = \sigma^2 f(\alpha)$, where $f(\alpha) \triangleq \frac{1}{2}\alpha^2\gamma^2 tr[Q\sum_{k=0}^{\infty} A_\alpha^k (Q - diag(Q))^2 (A_\alpha^T)^k]$. Since $J_{priv}(\alpha, \sigma)$ is assumed to be strictly increasing in the theorem statement, $f(\alpha)$ is also strictly increasing for $\alpha \in (0, 1]$. Also, it can be readily observed that $f'(0) = 0$ and $f''(0) > 0$. From the proof of Lemma 4.3.5, we have $J'_{ico}(0) \leq 0$ and $J'_{ico}(1) = 0$. Thus, $J'(0, \sigma) \leq 0$ and $J'(1, \sigma) > 0$. Also, $J(\alpha, \sigma)$ is strictly convex for $\alpha \in [0, 1)$. Thus, $\alpha^*(\sigma) \in [0, 1)$ is unique and $J'(\alpha^*(\sigma), \sigma) = 0$.

We prove the theorem by contradiction. Suppose $\alpha^*(\sigma)$ is not monotonically decreasing function. Then, there exist $0 < \sigma_1 < \sigma_2$ such that $0 \leq \alpha^*(\sigma_1) < \alpha^*(\sigma_2) < 1$. Further,

$$J'(\alpha^*(\sigma_1), \sigma_1) = \sigma_1^2 f'(\alpha^*(\sigma_1)) + J'_{ico}(\alpha^*(\sigma_1)) = 0, \quad \text{and}$$
$$J'(\alpha^*(\sigma_2), \sigma_2) = \sigma_2^2 f'(\alpha^*(\sigma_2)) + J'_{ico}(\alpha^*(\sigma_2)) = 0.$$

Subtracting, we get

$$\begin{aligned} 0 &= \sigma_1^2 f'(\alpha^*(\sigma_1)) - \sigma_2^2 f'(\alpha^*(\sigma_2)) + J'_{ico}(\alpha^*(\sigma_1)) - J'_{ico}(\alpha^*(\sigma_2)) \\ &\overset{(a)}{\leq} \sigma_1^2 [f'(\alpha^*(\sigma_1)) - f'(\alpha^*(\sigma_2))] + J'_{ico}(\alpha^*(\sigma_1)) - J'_{ico}(\alpha^*(\sigma_2)) \overset{(b)}{<} 0, \end{aligned}$$

where (a) follows since f is an increasing function and (b) follows from the strict convexity of f and J_{ico}. Thus, there is a contradiction and therefore, the theorem follows. ■

Due to the convexity and increasing properties of the functions involved in Theorem 4.3.7, we are able to obtain a nice characterization of the optimum cooperation level. This result implies that under the conditions given in Theorem 4.3.7, it is always beneficial for the agents to reduce their cooperation level if they want to increase their privacy level. It characterizes an important and previously unidentified tradeoff between privacy and cooperation is multi-agent systems. Next, we show how to design the artificial cost function J_{nco}(in cases where individual costs are not present) which guarantees that the conditions of Theorem 4.3.7 hold true.

Corollary 4.3.8. ***(Design of artificial cost function)*** *Assume that Q satisfies the following properties:*

(i) $Q > 0$ and $\frac{\lambda_1(Q)}{\lambda_N(Q)} < 1.5$,

(ii) $diag(Q) = \mu I_N$ for some $\mu > 0$.

Then, there exists a $\bar{Q} = \delta I_N$ with $\lambda_1(Q) < \delta < 1.5\lambda_N(Q)$ and $\gamma < \delta^{-1}$ such that $J_{priv}(\alpha, \sigma)$ is strictly increasing for $\alpha \in (0, 1]$, and $J_{ico}(\alpha)$ and $J_{priv}(\alpha, \sigma)$ are strictly convex for $\alpha \in [0, 1)$.

Proof. See appendix A.2. ■

The above corollary guarantees that $\alpha^*(\sigma)$ is a monotonically decreasing function. It should be noticed, however, that the conditions presented in Theorem 4.3.7 and

Corollary 4.3.8 are not necessary, and $\alpha^*(\sigma)$ may or may not exhibit similar behavior if these conditions are not satisfied.

4.4 Consensus and Voronoi Tessellation

In this section, we illustrate our results through two prototypical problems, namely the consensus and the one-dimensional Voronoi tessellation problems.

4.4.1 Consensus with Privacy and Cooperation

Consensus algorithms are used by autonomous agents to agree on a common value in a distributed fashion. The algorithm involves sharing of state information among the agents. The iterations of the consensus algorithm in a discrete time setting can be represented as [5]

$$x(k+1) = (I_N - \gamma L)x(k), \tag{4.28}$$

where $L = [l_{ij}]$ is the Laplacian matrix of the graph representing the agents interaction. We consider an undirected consensus graph for which the Laplacian is symmetric, positive semi-definite with positive diagonal entries and non-positive non-diagonal entries. For such a graph, $\mathbf{1}_N$ is an eigenvector of L associated with the eigenvalue 0, that is, $L\mathbf{1}_N = 0$ and $\lambda_N(L) = 0$. Further, we assume that the graph is connected, which is equivalent to $\lambda_{N-1}(L) > 0$. Then, the algorithm in (4.28) asymptotically achieves average consensus, that is, $\lim_{k\to\infty} x(k) = \mu\mathbf{1}_N$, where $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i(0)$.

The consensus algorithm in (4.28) can be viewed as a gradient descent algorithm to minimize the following disagreement function [5]

$$J_{co}(x) = \frac{1}{2}x^T L x = \frac{1}{4}\sum_{i=1}^{N}\sum_{j:j\in\mathcal{N}_i} -l_{ij}(x_i - x_j)^2. \tag{4.29}$$

Thus, the consensus problem fits into our framework (Problem (4.1) and solution (4.3)) with $Q = L, r = 0$, and $s = 0$. Further, it also satisfies assumptions **A.1** and **A.2**. To introduce the cooperation level, we select the following decoupled cost function

$$J_{nco}(x) = \frac{1}{2N}x^T x - \frac{1}{N}a^T x + \frac{1}{2N}a^T a + b^2,$$

where $a \in \mathbb{R}^N$ and $b \in \mathbb{R}$. Note that the optimum of $J_{nco}(x)$ is achieved at $x = a$ and the optimum cost is b^2. Comparing the above cost function with (4.14), we obtain $\bar{Q} = \frac{1}{N}I_N, \bar{r} = -\frac{1}{N}a$, and $\bar{s} = \frac{1}{2N}a^T a + b^2$. Thus,

$$Q_\alpha = \alpha L + \frac{1-\alpha}{N}I_N,\ A_\alpha = I_N - \gamma Q_\alpha,\ b_\alpha = \frac{\gamma(1-\alpha)}{N}\ a.$$

Further, the step size can be chosen according to Lemma 4.3.4 to guarantee convergence of the consensus algorithm. The resulting cost with privacy mechanism becomes

$$J(\alpha, \sigma) = \frac{1}{2}(tr(LP_\alpha) + m_\alpha^T L m_\alpha).$$

Notice that, for $\alpha = 1$, A_1 becomes marginally stable and thus, the covariance matrix P_1 becomes unbounded resulting in instability (see discussion below (4.12)). Thus, the agents have to choose a cooperation level $0 \leq \alpha < 1$ for the cost to remain finite.

We now consider a specific consensus example with $N = 4$ agents, and the fol-

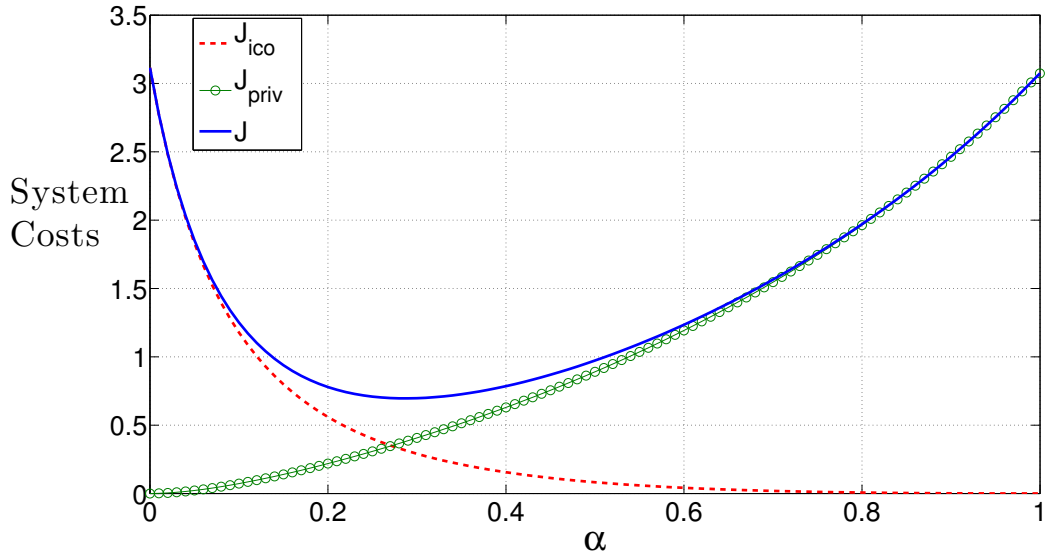


Figure 4.1. System costs as a function of cooperation level for privacy noise level $\sigma = 3$.

lowing Laplacian matrix

$$L = \begin{bmatrix} 0.8 & -0.14 & -0.15 & -0.51 \\ -0.14 & 1.4 & -0.85 & -.41 \\ -0.15 & -0.85 & 1.1 & -0.1 \\ -0.51 & -0.41 & -0.1 & 1.02 \end{bmatrix}.$$

Let $a = [1.5, 1, 3, 0]^T$ and $x(0) = [0.2, 0.6, 1.2, 2]^T$. Figure 4.1 shows the system costs J, J_{priv}, and J_{ico} in (4.21)-(4.23) as a function of α for $\sigma = 3$. We can observe that J_{ico}, which is the system cost in absence of privacy noise, is monotonically decreasing (c.f. lemma 4.3.5). This validates our understanding that it is always beneficial for the agents to have full cooperation if they do not desire any privacy.

The effect of privacy noise is included in the cost J_{priv}. It is interesting to observe its behavior at different cooperation levels. Note that the noise has only a marginal effect on the cost at smaller cooperation levels. In fact, when the agents

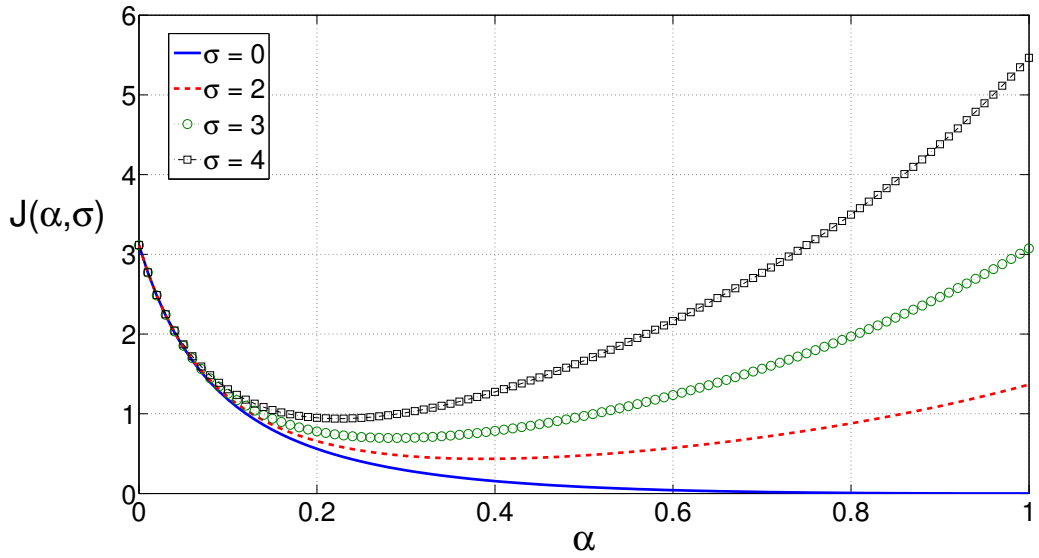


Figure 4.2. Consensus cost as a function of cooperation and privacy.

do not cooperate ($\alpha = 0$), the noise does not affect the performance at all – which is natural because the agents do not share any information. In contrast, the effect of noise is significantly higher at larger cooperation levels, because the agents use the noisy states in their updates. Thus, the cost J_{priv} is a monotonically increasing function of α. The two cost curves J_{ico} and J_{priv} highlight the trade-off between having full cooperation vs. no cooperation. As evident from the resulting overall cost curve J, an intermediate optimum cooperation level should be chosen to achieve best performance.

Figure 4.2 shows the overall cost J achieved as a function of the cooperation level for various levels of privacy noise. Observe that for each cooperation level, the cost increases with the noise level. These curves highlight that the optimum cooperation level changes with the privacy noise level which can be seen explicitly in figure 4.3. Note that along with the fact that J_{priv} is an increasing function, both J_{priv} and J_{ico} are strictly convex. Thus, $\alpha^*(\sigma)$ is a monotonically decreasing function (c.f.

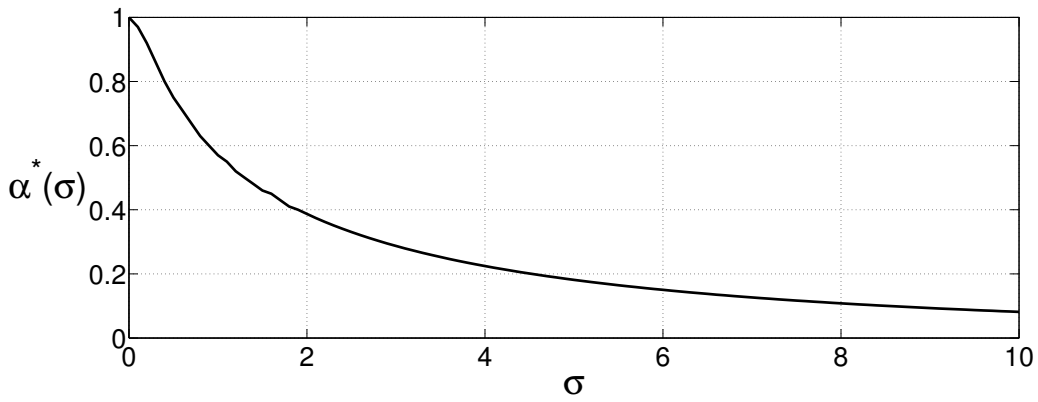


Figure 4.3. Variation of the optimum cooperation level with noise for Consensus.

theorem 4.3.7), which implies that it is always better for the agents to reduce their cooperation level if they desire to have a higher level of privacy. Finally, note that the consensus example does not satisfy the conditions in corollary 4.3.8 (see discussion below corollary 4.3.8).

4.4.2 Centroidal Voronoi Tessellation with Privacy and Cooperation

In this subsection, we show that our results and the intuition gained from the consensus problem, hold even when some of our assumptions are not satisfied; thus suggesting that a tradeoff between privacy and cooperation exists in a large class of distributed systems. We study a 1-dimensional centroidal Voronoi tessellation (CVT) problem over the interval $\Omega = [0, 1]$. The goal of a CVT problem is to divide Ω into N regions denoted by $\{\Omega_i\}_{i\in\mathcal{N}}$, and find N points in Ω, denoted by $\{x_i\}_{i\in\mathcal{N}}$ such that (i) Ω_i is the Voronoi region[1] of x_i, and (ii) x_i is the centroid of Ω_i. The CVT problem

[1]The Voronoi region for x_i is defined by $V_{x_i} = \{y : |y - x_i| \leq |y - x_j| \quad \forall\, j \neq i\}$.

can be expressed as the following optimization problem:

$$\min_{x} \quad J_{co}(x) = \int_0^1 \min_{i \in \mathcal{N}} (x_i - y)^2 dy. \tag{4.30}$$

The 1D-CVT problem can be viewed as an optimal resource allocation problem, and it has wide applications including data compression and scalar quantization [64].

Without loss of generality, we assume that $x_1 \leq x_2 \leq \cdots \leq x_N$. Then, the cost in (4.30) can be written as

$$J_{co}(x) = \frac{1}{24} \sum_{i=1}^{N} \left[(x_i - x_{i-1})^3 + (x_{i+1} - x_i)^3 \right], \tag{4.31}$$

where $x_0 \triangleq -x_1$ and $x_{N+1} \triangleq 2 - x_N$ are dummy variables introduced for ease of analysis. It can be easily verified that $J_{co}(x)$ is convex. Further, the i^{th} component of its gradient can be written as

$$\frac{\partial J_{co}(x)}{\partial x_i} = \frac{1}{4}(2x_i - x_{i-1} - x_{i+1})(x_{i+1} - x_{i-1}). \tag{4.32}$$

By examining (4.32), it follows that the optimum value x^* that minimizes $J_{co}(x)$ also minimizes a different cost function, denoted by $\tilde{J}_{co}(x)$, whose i^{th} component of the gradient is

$$\frac{\partial \tilde{J}_{co}(x)}{\partial x_i} = \frac{1}{4}(2x_i - x_{i-1} - x_{i+1}). \tag{4.33}$$

Since $J_{co}(x)$ and $\tilde{J}_{co}(x)$ have the same optimum, we can use the gradient of either cost functions in the gradient descent algorithm for solving the CVT problem. However, the gradient of $J_{co}(x)$ is non-linear whereas the gradient of the $\tilde{J}_{co}(x)$ is linear. We choose the linear gradient since it results in a linear gradient descent algorithm and fits into our framework (4.3). Using the individual components in (4.33), the gradient

can be written as

$$\nabla \tilde{J}_{co}(x) = Qx + r, \tag{4.34}$$

where $r = [0, 0, \cdots, 0, -0.5]^T$, and Q is the following tri-diagonal matrix:

$$Q = \frac{1}{4}\begin{bmatrix} 3 & -1 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & 0 & \cdots & 0 \\ \vdots & & & \vdots & & & \vdots \\ 0 & 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 3 \end{bmatrix}.$$

Lemma 4.4.1. ***(Properties of** Q**)*** *Q is positive definite and $\rho(Q) = 1$.*

Proof. From [65], Theorem 5, we get

$$\lambda_i(Q) = \frac{1}{2}\left(1 + \cos\frac{(i-1)\pi}{N}\right) \quad \text{for} \quad i = 1, 2, \cdots, N.$$

Thus, $\lambda_1(Q) = 1$ and $\lambda_N(Q) > 0$ and Lemma follows. ■

The new cost function can then be written as $\tilde{J}_{co}(x) = \frac{1}{2}x^T Q x + r^T x$ and the gradient descent to solve the CVT problem in (4.30) becomes

$$\mathbf{S_{CVT}}: \qquad x(k+1) = x(k) - \gamma_1(Qx(k) + r) \triangleq A_1 x(k) + b_1. \tag{4.35}$$

The algorithm $\mathbf{S_{CVT}}$ with $\gamma_1 = 1$ is the well known Lloyd's Algorithm [64] for solving CVT problems in 1-D.

Remark 4.4.2. ***(Generalization for non-quadratic cost functions)*** Although the CVT problem has a cubic cost function, its optimum can be achieved via a linear

algorithm so that our framework is still applicable. This suggests that our framework may be applicable to general non-quadratic convex cost functions, provided that the optimum can be achieved via a linear algorithm. □

To introduce the cooperation level, we consider the following quadratic cost

$$J_{nco}(x) = \frac{1}{N}\sum_{i=1}^{N}\int_0^1 (x_i - y)^2 dy = \frac{1}{N}x^T x - \frac{1}{N}\mathbf{1}_N^T x + \frac{1}{3}. \tag{4.36}$$

Observe that the decoupled cost in (4.36) is the counterpart of the coupled cost in (4.30), and that $J_{nco}(x)$ satisfies assumption **A.3**. Further, note that $x = [0.5, 0.5, \cdots, 0.5]^T$ is the optimum of $J_{nco}(x)$. By comparing the above cost function with (4.14), we obtain $\bar{Q} = \frac{2}{N}I_N, \bar{r} = -\frac{1}{N}\mathbf{1}_N$, and $\bar{s} = \frac{1}{3}$, and the corresponding values of $Q_\alpha, A_\alpha, b_\alpha$ and H_α can be calculated using (4.16). Since Q_α is positive definite, the stability condition (4.17) for the CVT problem reduces to $\gamma < 2$. We next analyze the system cost and performance.

The cost in (4.31) can be simplified as $J_{co}(x) = \frac{1}{3}x_1^3 + \frac{1}{12}\sum_{i=1}^{N-1}(x_{i+1} - x_i)^3 + \frac{1}{3}(1 - x_N)^3$. We make the linear transformation $z = Gx + g$ to obtain

$$\underbrace{\begin{bmatrix} x_1 \\ x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_N - x_{N-1} \\ 1 - x_N \end{bmatrix}}_{z} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots \\ & & \vdots & & \\ 0 & \cdots & 0 & -1 & 1 \\ 0 & 0 & \cdots & 0 & -1 \end{bmatrix}}_{G} x + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_{g}.$$

Let m_α and P_α denote the steady state mean and covariance achieved by algorithm $\mathbf{S}_\alpha^{\mathbf{priv}}$ in (4.16) for the CVT problem. Further, let η_α and V_α denote the steady state mean and covariance of z, and let η_i and v_{ij} denote the elements of η_α and V_α,

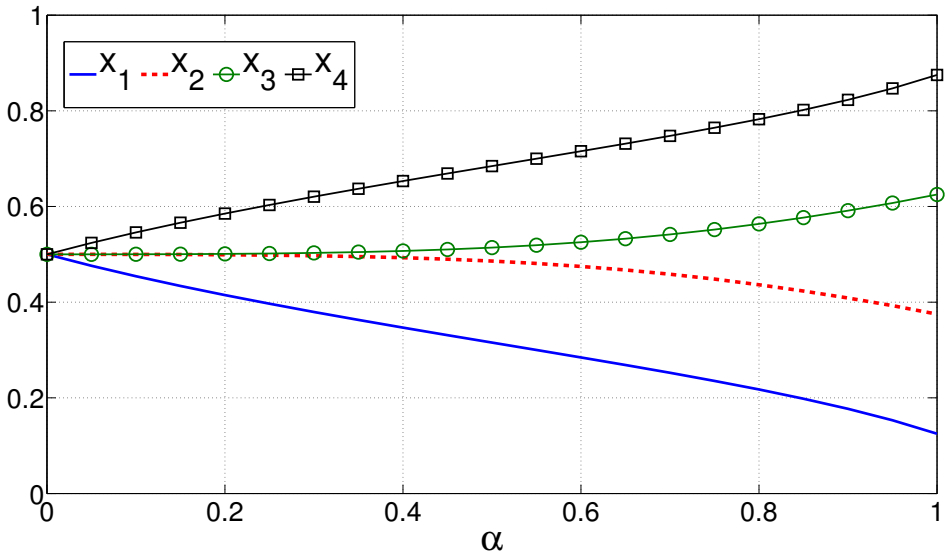


Figure 4.4. Steady states achieved by $\mathbf{S}_\alpha$ for the CVT problem in the absence of noise.

respectively. Due to the linear transformation, we have $\eta_\alpha = Gm_\alpha + g$ and $V_\alpha = GP_\alpha G^T$. Recall that, for a scalar random variable with distribution $y \sim \mathbf{N}(\nu, \theta^2)$, it holds $\mathbb{E}[y^3] = \nu^3 + 3\nu\theta^2$. Thus, the expected cost for the CVT problem becomes

$$J(\alpha, \sigma) = \frac{1}{3}(\eta_1^3 + 3\eta_1 v_{11}) + \frac{1}{12}\sum_{i=2}^{N}[\eta_i^3 + 3\eta_i v_{ii}] + \frac{1}{3}(\eta_{N+1}^3 + 3\eta_{N+1} v_{N+1,N+1}). \quad (4.37)$$

Consider now an example with $N = 4$ agents and $\gamma = 1$. Figure 4.4 shows how the steady state values achieved by the agents vary with the cooperation level, when they use algorithm $\mathbf{S}_\alpha^{\mathbf{priv}}$ in (4.16) in the absence of noise. Observe that, when $\alpha = 0$, the optimum of $J_{nco}(x)$ in (4.36) is $x = [0.5, 0.5, 0.5, 0.5]^T$, so that the agents occupy the same location due to the lack of cooperation. When $\alpha = 1$, the agents cooperate completely and achieve the solution $x = [\frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}]^T$, which is the optimum of $J_{co}(x)$ in (4.30). Figure 4.4 shows the solution achieved by algorithm $\mathbf{S}_\alpha^{\mathbf{priv}}$ for the intermediate values of α.

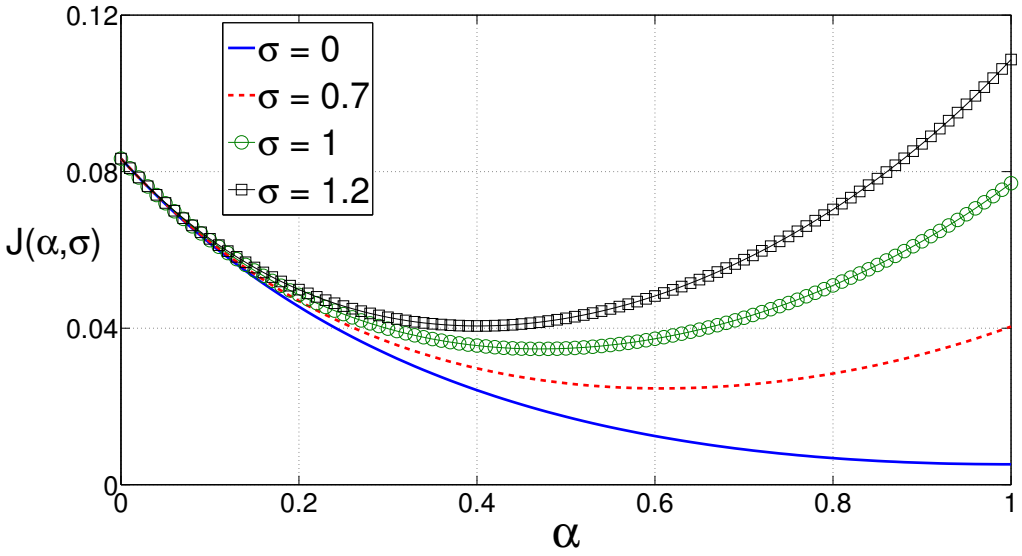


Figure 4.5. CVT cost as a function of cooperation level and noise level.

Figure 4.5 shows the system cost as a function of the cooperation level for various values of σ. Similar to the consensus example, we observe that the cost is a convex function of α and the optimum α is less than 1. Moreover, the cost increases when the noise level increases.

The analysis of the cost function simplifies if we consider the simple, albeit useful, case of $N = 2$ agents. In this case, we can algebraically solve the steady state equations (4.18) and (4.19) to obtain

$$m_\alpha = \begin{bmatrix} \frac{1}{2} - \frac{\alpha}{4} & \frac{1}{2} + \frac{\alpha}{4} \end{bmatrix}, \text{ and } P_\alpha = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_1 \end{bmatrix},$$

where $p_1 = \frac{\alpha^2\sigma^2(8-\alpha^2)}{32(4-\alpha^2)}$ and $p_2 = \frac{\alpha^4\sigma^2}{32(4-\alpha^2)}$. Thus,

$$J(\alpha, \sigma) = \frac{1}{48} + \frac{(1-\alpha)^2}{16} + \frac{\sigma^2\alpha^2(4+\alpha)}{32(2+\alpha)}. \tag{4.38}$$

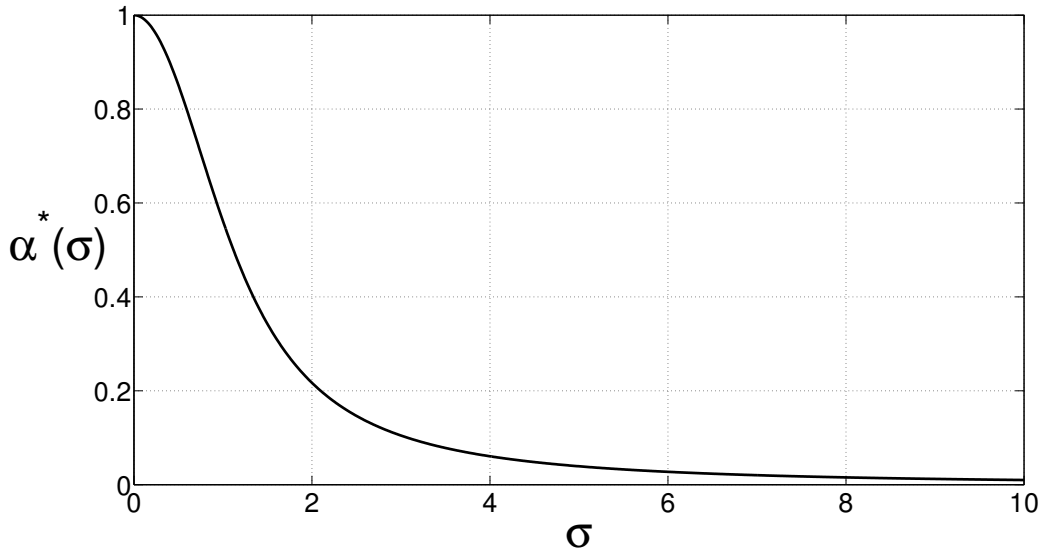


Figure 4.6. Variation of the optimum cooperation level with noise for CVT problem.

It can be easily verified that this cost is monotonically increasing w.r.t. σ and is convex with respect to α. Consequently, an optimum α exists for each value of σ, and it is given by

$$\alpha^*(\sigma) = \frac{\sqrt{(1+2\sigma^2)^2+8} - (1+2\sigma^2)}{2}. \tag{4.39}$$

As shown in Figure 4.6, the function $\alpha^*(\sigma)$ is monotonically decreasing. This indicates that it is best for the agents to cooperate fully when no noise is present, and to reduce their cooperation level when the privacy noise increases.

Remark 4.4.3. ***(Similarity between consensus and CVT results)*** By comparing Fig. 4.5 and Fig. 4.6 in the CVT example (with cubic cost) with Fig. 4.2 and Fig. 4.3 in the consensus example (with quadratic cost), we observe a similar pattern in the variation of the cost and the optimum cooperation level. This observation strengthens our belief that a similar tradeoff should appear in problems with general

non-quadratic convex cost function that can be minimized via linear iterations. This generalization is left as a subject of future research. □

4.5 Summary

We consider a multi-agent system where the agents cooperatively optimize a quadratic cost function while ensuring privacy of their states over time. We provide a framework in which the agents can respond to the privacy noise present in the system by varying their cooperation level. We show that there exists an optimum cooperation level that minimizes the cost. We studied two examples of consensus and Voronoi Tessellations, and showed that they fit into our framework. The results obtained illustrate a tradeoff between performance, privacy and cooperation, and suggest that, to optimize performance, agents should decrease their cooperation level if they want to increase the privacy level.

CHAPTER 5

CONCLUSION AND FUTURE DIRECTIONS

In this dissertation, we studied three problems in the field of designing scalable protocols for networked control systems and protecting privacy of agent's sensitive parameters in multi-agent cyber-physical systems. In chapter 2, we studied a distributed estimation problem where multiple plants and estimators perform remote estimation over a shared rate limited communication network. We designed a rate control protocol that reduces congestion in the network and optimizes the estimation performance of the overall system. We proposed a probabilistic transmission scheme and implemented rate control by varying the transmission probabilities. We also analyzed the stability of the protocol in presence of fixed network delays. There exists a possibility of various extensions and modifications of this problem.

- The probabilistic data transmission scheme may sometimes result in a scenario when the data is not transmitted for a long time interval, although such instances will occur with low probability. It may result in significant increase of the estimation error performance. To prevent this behavior, we can augment the probabilistic schemes with other transmission schemes. For example, we can study alternate schemes that transmit data in a periodic or event triggered fashion and compare their performance with the probabilistic scheme. A critical issue for design of such transmission scheme is the structure of the cost function. The transmission scheme should result in a separable cost structure, thus enabling scalable design of network protocols.

- In our solution, we used the steepest descent algorithm for optimizing the transmission rates. Alternatively, we can use other optimization algorithms like Newtons's method. It may also be possible that various components in the distributed system use different optimization algorithms. The algorithms may have different rates of convergence and this may affect the stability of the systems. In such heterogeneous networks, it is important to characterize the rate of convergence of optimization algorithms, and analyze the system stability.

- In our work, we formulated the $SYSTEM$ problem that had coupling among its cost terms. To obtain a scalable solution, we relaxed the problem and transformed it into NUM framework to obtain a $USER$ problem. There is a performance loss during this transformation since the $USER$ problem is a scalable approximation to the $SYSTEM$ problem. Quantifying this performance loss (or obtaining bounds on it) would provide robust performance guarantee in the problem.

- In the first part of the problem, we considered random delays in the network and formulated the cost function. We then assumed that the network is stable in presence of these delays and analyzed the steady state case. In the latter part, we derived the stability condition of the system under fixed transmission delays in the links. However, data packets may experience variable delays over a route due to queuing delays at the intermediate routers. Therefore, the stability analysis needs to be extended for time varying or stochastic delays.

This study also motivates several more general research directions. In our work, we proposed a primal form of the distributed solution. It requires modifications only at the source and destinations, without changing the network architecture and functionality. This structure of the proposed protocol is similar to TCP and enables both the protocols to co-exist in the same communication network. It is of interest

to obtain analytical results characterizing the equilibrium and rate allocation in such heterogeneous network when these two different protocols operate together.

The rate control problem is formulated at the transport layer of the OSI model and the solution obtained is similar to TCP, which is also a transport layer protocol. Similar design problems can be formulated at other layers and each layer can be designed separately for control oriented purposes. The NUM framework that we utilized to obtain a solution is also applicable at other network layers [17]. For example, the optimization problem can be formulated at network or MAC layer and can be solved to obtain optimal routing and channel contention schemes, respectively. Thus, a natural extension of the work is to formulate new class of control and estimation problems at other layers, and obtain distributed solutions.

In chapters 3 and 4, we studied problems related to privacy of agents in multi-agent cyber-physical systems. We utilized the notion of Differential Privacy (DP) in this dissertation and developed noise adding privacy mechanisms to prevent leakage of information about sensitive parameters/states of the agents. In chapter 3, we considered a multi-agent LTI system which is monitored by a control center using measurements from the agents. We developed a Laplacian DP mechanism to protect the privacy of sensitive parameters contained in the dynamics of the agents from an intruder that hacks into the control center. We derived an upper bound to the sensitivity of the system from the private parameters to outputs, and used it to determine the privacy noise level. We illustrated our framework through the examples of second-order consensus and LQR control. Our numerical results show that for an asymptotic regime of low privacy and high SNR, the privacy mechanism causes marginal degradation in the eigenvalue identification performance at the control center, as compared to the parameter identification performance by the intruder. Some possible extensions of this work are

- Since the exact sensitivity of the system is difficult to calculate, we obtained

an upper bound to design the noise level. It would be of interest to determine the tightness of this upper bound and determine conditions for which it is tight/loose. Also, one may possibly explore alternative techniques to obtain better bounds.

- We assumed that the capabilities of the intruder is limited to snooping on the measurements of the agents sent to the control center or hacking into the control center. However, there may be scenarios in which the intruder may hack into one or more agents, thereby gaining their state information. This raises the issue of agents not trusting each other and additional privacy mechanisms need to be implemented in the system to protect the parameters in such cases.

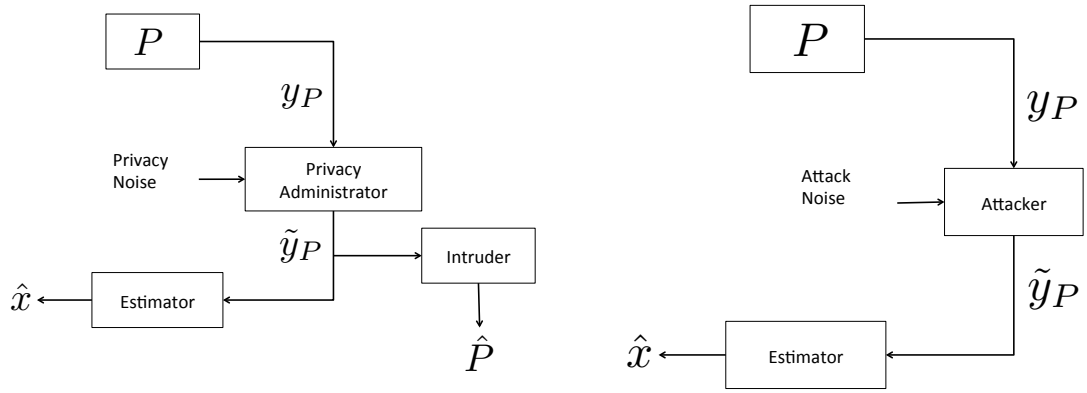
In chapter 4, we studied a multi-agent system in which the agents cooperatively solve a quadratic optimization problem and presented a Gaussian DP mechanism to protect privacy of their states over time. We showed that in presence of the privacy mechanism, full cooperation degrades the system performance since the agents use each other's noisy states. We developed a framework in which the agents can vary their cooperation level in response to privacy noise and obtained an optimal cooperation level for a given privacy level that minimizes the performance degradation. We derived conditions under which it is always beneficial for the agents to reduce their cooperation if they want a higher level of privacy in the system. We illustrated this privacy vs cooperation tradeoff through the examples of consensus and centroidal Voronoi tessellations. The possible extension to this work include the following.

- We considered a multi-agent quadratic optimization problem and showed that a privacy vs cooperation tradeoff exists in this framework. Although quadratic optimization has wide applications, it would be interesting to see if such tradeoff exists in more general optimization problems, possibly with convex cost functions.

- To introduce the cooperation level into our problem, we formulated and used a decoupled cost function. The performance of the system depends on this decoupled cost and therefore, identifying a decoupled cost function that maximizes the overall system performance would improve the results. Furthermore, a more rigorous framework to characterize the cooperation level of the system needs to be developed in which the cooperation level could be solely calculated based on the cost function that the agents are trying to minimize.

Apart from these extensions, there are some future directions related to privacy in cyber-physical systems that can be pursued. One possible problem would be to analyze a scenario when one or more agents in the system are non-conforming and do not implement the privacy mechanism. For example, this can happen if an agent determines that its parameters/state are not sensitive or the agent is compromised by an intruder. It can result in privacy breach of other conforming agents as well. Quantifying the privacy loss in such cases would help in designing robust privacy mechanisms that are resilient to such non-conforming agents. It will also help to identify the weakest link (agent) in the system, which can result in maximum privacy loss, if compromised.

Another possible direction would be to examine the relation between privacy and security in CPS. We explain this idea via a simple example, without being rigorous in our notations. Fig. 5.1a shows a simple privacy setup consisting of a plant P whose state is estimated by an estimator (denoted by $\hat{x}$) using the measurements y_P. An intruder also tries to estimate some private parameters of the plant (denoted by $\hat{P}$) using the measurements. To prevent this, a privacy administrator implements a DP mechanism that distorts the measurements by adding random noise. Assume that the plant P belongs to a set $\mathcal{P}$ and the measurements belong to set $\mathcal{Y}$. The DP noise guarantees that for any two *adjacent* plants P and P' and for all $S \subset \mathcal{Y}$, the



(a) Privacy setup

(b) Security setup

Figure 5.1. Privacy vs security framework

following holds

$$e^{-\epsilon} \leq \frac{\mathbb{P}(\tilde{y}_P \in S)}{\mathbb{P}(\tilde{y}_{P'} \in S)} \leq e^{\epsilon} \tag{5.1}$$

The goal of the system administrator is to minimize the effect of privacy noise on the estimation performance, while maintaining privacy as given by the following problem

$$\begin{aligned} \mathbf{P1}: \quad \min \quad & \mathbb{E}\left[\left(\hat{x}(y_P) - \hat{x}(\tilde{y}_P)\right)^2\right] \\ s.t. \quad & (5.1) \text{ holds.} \end{aligned}$$

Now consider a security setup in fig. 5.1b with the same plant and estimator but with an attacker present between the two. The attacker distorts the measurements using random noise and its goal is to cause maximum estimation degradation at the

estimator while remaining stealthy

$$\begin{aligned} \mathbf{P2}: \quad \max \quad & \mathbb{E}\left[\left(\hat{x}(y_P) - \hat{x}(\tilde{y}_P)\right)^2\right] \\ s.t. \quad & D_{KL}\left(y_P || \tilde{y}_P\right) \leq \epsilon, \end{aligned} \tag{5.2}$$

where D_{KL} denotes the Kullback-Leibler divergence and is a measure of the difference between the probability distributions of two random variables. Notice the similarity between the conditions (5.1) and (5.2). The DP condition (5.1) states that the outputs of two adjacent plants should appear almost statistically similar to the intruder. The stealthiness condition (5.2) also guarantees that the noise added by the attacker should not change the distribution of the output significantly. Although the DP criteria in (5.1) is defined using probabilistic notion and the stealthiness criteria in (5.2) uses $K - L$ divergence, both imply that the corresponding outputs should not be significantly different, statistically. Furthermore, the objectives of the privacy administrator and the attacker are contrary to each other. This suggests that the privacy and security problems **P1** and **P2** are related, and in some sense *dual* to each other. This relation needs to be examined further. It can provide significant insights and connect the results in fields of security and privacy.

APPENDIX A

PROOFS

A.1 Selective Proofs for Chapter 3

Proof of Lemma 3.3.3 The statement is trivial when the dynamics is stable. For marginally stable case, we begin by stating the following facts

1. Since $\bar{A}_d = \nu_1(A_d(P))\tilde{\nu}_1(A_d(P))$ with $\tilde{\nu}_1(A_d(P))\nu_1(A_d(P)) = 1$, we have $A_d(P)\bar{A}_d = \bar{A}_d A_d(P) = \bar{A}_d^2 = \bar{A}_d$. Thus, for $k \geq 1$, and an impulse input, we have

$$\bar{A}_d x(k) = \bar{A}_d A_d(P) x(k-1) = \bar{A}_d x(k-1) \cdots = \bar{A}_d x(1) = \bar{A}_d A_d(P)(x(0)+K_0 B) = \bar{x}.$$

2. $(A_d(P) - \bar{A}_d)\bar{x} = \mathbf{0}_n$.

Next, for $k \geq 1$ we have

$$\begin{aligned}
e(k+1) = A_d(P)x(k) - \bar{x} &\overset{(a)}{=} (A_d(P) - \bar{A}_d)x(k) \\
&\overset{(b)}{=} (A_d(P) - \bar{A}_d)x(k) - (A_d(P) - \bar{A}_d)\bar{x} \\
&= (A_d(P) - \bar{A}_d)e(k) \\
&\vdots \\
&= (A_d(P) - \bar{A}_d)^k e(1) \\
&= (A_d(P) - \bar{A}_d)^k (A_d(P)(x(0)+ FK_0) - \bar{A}_d(x(0)+ FK_0)) \\
&= (A_d(P) - \bar{A}_d)^{k+1}(x(0) + FK_0),
\end{aligned}$$

where (a) follows from fact 1 and (b) follows from fact 2. ■

Proof of Lemma 3.3.4 Since $\bar{A}_d = \nu_1(A_d(P))\tilde{\nu}_1(A_d(P))$ with $\tilde{\nu}_1(A_d(P))\nu_1(A_d(P)) = 1$, we have

$$\tilde{A}_d(P)\nu_1(A_d(P)) = A_d(P)(I_n - \bar{A}_d)\nu_1(A_d(P)) = \mathbf{0}_n.$$

Thus, 0 is an eigenvalue of $\tilde{A}_d(P)$.

Next, for $\lambda_i(A_d(P)) \neq 1$, we have

$$\begin{aligned}\bar{A}_d\nu_i(A_d(P)) = \bar{A}_d A_d(P)\nu_i(A_d(P)) = \bar{A}_d\lambda_i(A_d(P))\nu_i(A_d(P))\\ \Rightarrow (1 - \lambda_i(A_d(P)))\bar{A}_d\nu_i(A_d(P)) = \mathbf{0}_n\\ \Rightarrow \bar{A}_d\nu_i(A_d(P)) = \mathbf{0}_n.\end{aligned}$$

Using this, we get

$$\begin{aligned}\tilde{A}_d(P)\nu_i(A_d(P)) &= A_d(P)(I_N - \bar{A}_d)\nu_i(A_d(P))\\ &= A_d(P)\nu_i(A_d(P)) = \lambda_i(A_d(P))\nu_i(A_d(P)).\end{aligned}$$

Thus, all eigenvalues $\lambda_i(A_d(P)) \neq 1$ are also the eigenvalues of $\tilde{A}_d(P)$. ■

Proof of Lemma 3.3.6 The Lemma follows from Corollary 2.4 in [51] by noting that $f(s) = s^k$, $f^{'}(s) = ks^{k-1}$. Further, since all the eigenvalues of A and $\tilde{A}$ lie inside the closed unit circle, the closed convex hull of the eigenvalues of matrices A and $\tilde{A}$ (denoted by $co(A, \tilde{A})$) is a subset of the closed unit circle. Thus,

$$\max_{s \in co(A,\tilde{A})} f^{'}(s) \leq \max_{s:|s|\leq 1} k|s|^{k-1} = k \max\{\rho(A), \rho(\tilde{A})\}^{k-1}.$$

The Lemma then follows using the following matrix norm inequalities $\frac{||.||_1}{\sqrt{N}} \leq ||.||_F \leq \sqrt{N}||.||_2$. ■

Proof of Corollary 4.3.8 Let $'$ denote the derivative or partial derivative w.r.t. α. First, we derive conditions under which the cost term $J_{ico}(\alpha)$ is convex w.r.t. α. By differentiating (4.24) with respect to α we obtain

$$J''_{ico}(\alpha) = (m'_\alpha)^T Q m'_\alpha + (Qm_\alpha + r)^T m''_\alpha.$$

From the proof of lemma 4.3.5, for $\alpha = 1$, we have $Qm_1 + r = 0$ and $m'_1 = 0$. Thus, $J''_{ico}(1) = 0$. Now let $\alpha \in [0, 1)$. By differentiating (4.25), we get

$$\begin{aligned} &2(Q - \bar{Q})m'_\alpha + Q_\alpha m''_\alpha = 0, \\ &\overset{(a)}{\Rightarrow} m''_\alpha = \frac{2}{1-\alpha} Q_\alpha^{-1}(Q - \bar{Q})Q_\alpha^{-1}(Qm_\alpha + r), \end{aligned}$$

where (a) follows from (4.26). Substituting the derivatives m'_α and m''_α we get

$$J''_{ico}(\alpha) = \frac{1}{1-\alpha}(Qm_\alpha + r)^T Q_\alpha^{-1} \left(3Q - 2\bar{Q} + \frac{\alpha}{1-\alpha} Q\right) Q_\alpha^{-1}(Qm_\alpha + r).$$

Condition (iii) in the corollary guarantees that $3Q - 2\bar{Q} > 0$ and thus $J''_{ico}(\alpha) > 0$ for $\alpha \in (0, 1]$.

Next, we derive conditions under which the cost term $J_{priv}(\alpha, \sigma)$ is convex w.r.t. α. Recalling (4.16), let $A_\alpha = \alpha A + B$, where $A \triangleq \gamma(\bar{Q} - Q)$ and $B \triangleq I_N - \gamma\bar{Q} = (1 - \gamma\delta)I_N$. Further, let $H_\alpha = -\gamma\alpha\tilde{Q}$ where $\tilde{Q} \triangleq Q - \mathrm{diag}(Q)$. Differentiating (4.19) and substituting the above expressions, we get

$$P'_\alpha = A_\alpha P'_\alpha A_\alpha + \underbrace{AP_\alpha B^T + BP_\alpha A^T + 2\alpha A P_\alpha A^T + 2\sigma^2\gamma^2\alpha\tilde{Q}^2}_{W}. \tag{A.1}$$

Note that due to (iii) and (iv), $A > 0$ and $B > 0$. Further, we have the following facts (a) $QA_\alpha = A_\alpha Q$ and (b) $Q\tilde{Q} = \tilde{Q}Q$ (by (ii)). Using (a), (b) and (4.20), it can

be easily observed that $AP_\alpha B^T = BP_\alpha A^T > 0$. Thus, $W > 0$ and (A.1) resembles to a Lyapunov equation. Hence, we conclude that $P'_\alpha > 0$.

Taking derivative of (A.1), we get

$$P''_\alpha = A_\alpha P''_\alpha A_\alpha + \underbrace{2AP'_\alpha B^T + 2BP'_\alpha A^T + 2A(P_\alpha + 2\alpha P'_\alpha)A^T + 2\sigma^2\gamma^2\tilde{Q}^2}_{Z}. \qquad \text{(A.2)}$$

Again using (a) and (b) and taking the derivative of (4.20), we get $AP'_\alpha B^T = BP'_\alpha A^T > 0$.Thus, $Z > 0$ and (A.2) resembles to a Lyapunov equation. Hence, we get $P''_\alpha > 0$. Thus, $J'_{priv}(\alpha, \sigma) = \frac{1}{2}tr(QP'_\alpha) > 0$ and $J''_{priv}(\alpha, \sigma) = \frac{1}{2}tr(QP''_\alpha) > 0$ and the proof is complete. ■

BIBLIOGRAPHY

1. R. M. Murray, K.J. Astrom, S. P. Boyd, R. W. Brockett, and G. Stein. Future directions in control in an information-rich world. *IEEE Control Systems Magazine*, 23(2):20–33, 2003.

2. J.P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proceedings of IEEE*, 95(1):138–162, 2007.

3. Transmission control protocol.rfc-793, 1981.

4. S. Shakkottai and R. Srikant. Network optimization and control. *Foundations and Trends in Networking*, 2(3):271–379, 2007.

5. R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(2):215–233, 2007.

6. R. L. Raffard, C. J. Tomlin, and S. P. Boyd. Distributed optimization for cooperative agents: Application to formation flight. *IEEE Conf. on Decision and Control*, pages 2453–2459, 2004.

7. A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

8. H. Terelius, U. Topcu, and R. M. Murray. Decentralized multi-agent optimization via dual decomposition. *Proceedings of the 18th IFAC World Congress*, 44(1): 11245–11251, 2011.

9. C. Dwork. Differential privacy. *Proceedings of ICALP*, 4052:1–12, 2006.

10. C. Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–95, 2011.

11. P. Anatsaklis and J. Baillieul. Special issue on networked control systems. *Proceedings of the IEEE*, 95(1), 2007.

12. E. Garone, B. Sinopoli, and A. Casavola. Communication protocols for optimal control over lossy networks. *Annual Allerton Conference on Communication, Control and Computing*, 2007.

13. L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S.S. Sastry. Foundations of control and estimation over lossy networks. *Proceedings of the IEEE: Special Issue on Networked Control Systems*, 95(1), 2007.

14. V. Jacobson. Congestion avoidance and control. *ACM Computer Communication Review*, 18:314–329, 1988.

15. F.P. Kelly. Mathematical modeling of the internet. *Mathematics Unlimited-2001 and Beyond (Editors: B. Engquist and W. Schmid)*, pages 685–702, 2001.

16. F.P. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

17. M. Chiang, S.H. Low, A.R. Calderbank, and J.C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of IEEE*, 95(1), 2007.

18. A.T. Al-Hammouri, M.S. Branicky, V. Liberatore, and S.M.Phillips. Decentralized and dynamic bandwidth allocation in networked control systems. *WPDRTS*, 2006.

19. R. Johari and D. K. H. Tan. End-to-end congestion control for the internet: Delays and stability. *IEEE/ACM Transactions on Networking*, 9(6):818–832, 2001.

20. S. H. Low and D. E. Lapsley. Optimization flow control-i: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, 1999.

21. G. Vinnicombe. On the stability of networks operating tcp-like congestion control. *Proceedings of the 15th IFAC World Congress on Automatic Control*, 2002.

22. V. Gupta, A.F. Dana, J. Hespanha, R.M. Murray, and B. Hassibi. Data transmission over networks for estimation and control. *IEEE Transactions on Automatic Control*, 54(8):1807–1819, 2009.

23. S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, pages 397–413, 1993.

24. R. Caceres, N.G. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal loss characteristics. *IEEE Transactions on Information Theory*, 45:2462–2480, 1999.

25. C. A. Desoer and Y. T. Wang. On the generalized nyquist stability criterion. *IEEE Transactions on Automatic Control*, 25:187–196, 1980.

26. M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

27. G. B. Giannakis, V. Kekatos, N. Gatsis, S-J. Kim, H. Zhu, and B. F. Wollenberg. Monitoring and optimization for power grids: A signal processing perspective. *IEEE Signal Processing Magazine*, 30(5):107–128, 2013.

28. S. Nabavi and A. Chakrabortty. Topology identification for dynamic equivalent models of large power system networks. *American Control Conference*, pages 1138 – 1143, 2013.

29. M. C. Priess, R. Conway, J. Choi, J. M. Popovich, and C. Radcliffe. Solutions to the inverse lqr problem with application to biological systems analysis. *IEEE Transactions On Control Systems Technology*, 23(2):770–777, 2014.

30. M. Monfort, A. Liu, and B. Ziebart. Intent prediction and trajectory forecasting via predictive inverse linear-quadratic regulation. *AAAI Conference on Artificial Intelligence*, 2015.

31. D. Kendrik. Control theory with applications to economics. *Handbook of Mathematical Economics*, 1:111–158, 2000.

32. S. Han, U. Topcu, and G. J. Pappas. Differentially private distributed constrained optimization. *IEEE Transactions on Automatic Control*, 2016.

33. M.T. Hale and M. Egerstedt. Cloud-enabled multi-agent optimization with constraints and differentially private states. *http://arxiv.org/abs/1507.04371*, 2015.

34. A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. *IEEE Symposium on Security and Privacy*, 2008.

35. J. Le Ny and G. J. Pappas. Differentially private filtering. *IEEE Transactions on Automatic Control*, 59(2):341–354, 2014.

36. Z. Huang, S. Mitra, and G. Dullerud. Differentially private iterative synchronous consensus. *In Proceedings of the ACM Workshop on Privacy in the Electronic Society, WPES*, pages 81–90, 2012.

37. Z. Huang, Y. Wang, S. Mitra, and G. Dullerud. On the cost of differential privacy in distributed control systems. *3rd ACM International Conference on High Confidence Networked Systems (HiCoNS)*, 2014.

38. Y. Mo and R. M. Murray. Privacy preserving average consensus. *IEEE Transactions on Automatic Control*, 2016.

39. Z. Huang, S. Mitra, and N. Vaidya. Differentially private distributed optimization. *Proceedings of ICDCN '15*, 2015.

40. J. Hsu, A. Roth, T. Roughgarden, and J. Ullman. Privately solving linear programs. *Automata, Languages, and Programming: ICALP 14*, pages 612–624, 2014.

41. S. Han, U. Topcu, and G. J. Pappas. Differentially private convex optimization with piecewise affine objectives. *IEEE Conference on Decision and Control*, pages 2160–2166, 2014.

42. S. Song, K. Chaudhuri, and A. D. Sarwate. Stochastic gradient descent with differentially private updates. *IEEE Global Conference on Signal and Information Processing*, pages 245–248, 2013.

43. W. Ren and R. W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer-Verlag, 2008.

44. F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM Review*, 43(2):235–286, 2001.

45. V. Katewa, A. Chakrabortty, and V. Gupta. Protecting privacy of topology in consensus networks. *American Control Conference*, 2015.

46. H. Kong, G. Goodwin, and M. Seron. A revisit to inverse optimality of linear systems. *International Journal of Control*, 85(10):1506–1514, 2012.

47. B. Molinari. The stable regulator problem and its inverse. *IEEE Transactions on Automatic Control*, 18(5):454–459, 1973.

48. R. Bellman and K. J. Astrom. On structural identifiability. *Mathematical Biosciences*, 7:329 – 339, 1970.

49. L. Ljung. *System Identification: Theory for the User*. Prentice Hall, 1999.

50. C. Dwork. Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography Conference*, pages 265–284, 2006.

51. M. I. Gil. Perturbations of functions of diagonalizable matrices. *Electronic Journal of Linear Algebra*, 20:303–313, 2010.

52. C. V. Loan. The sensitivity of the matrix exponential. *SIAM Journal on Numerical Analysis*, 14(6):971–981, 1977.

53. S. F. Xu. Sensitivity analysis of the algebraic riccati equations. *Numerische Mathematik*, (75):121 – 134, 1996.

54. R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

55. J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

56. P. McDaniel and S. McLaughlin. Security and privacy challenges in the smart grid. *IEEE Security and Privacy*, 7(3):75–77, 2009.

57. Dept. of Energy United States. Data access and privacy issues related to smart grid technologies. Technical report, 2010.

58. R. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

59. Y. Lindell and B. Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1):59–98, 2009.

60. C. Orlandi. Is multiparty computation any good in practice? *IEEE Int. Conf. Acoustic Speech Signal Processing*, pages 5848–5851, 2011.

61. R. Rajagopal and M. J. Wainwright. Network-based consensus averaging with general noisy channels. *IEEE Transactions on Signal Processing*, 59(1):373–385, 2011.

62. L. Xiao, S. Boyd, and S. J. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46, 2007.

63. T. Tao. Topics in random matrix theory. *Graduate Studies in Mathematics, American Mathematical Society*, 132, 2012.

64. Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *Siam Review*, 41(4):637–676, 1999.

65. W. C. Yueh. Eigenvalues of several tridiagonal matrices. *Applied Mathematics E-Notes*, 5:66–74, 2005.

This document was prepared & typeset with pdfLaTeX*, and formatted with* NDdiss2$_\varepsilon$ *classfile (v3.2013[2013/04/16]) provided by Sameer Vijay and updated by Megan Patnott.*