# Networked Sensing, Estimation and Control Systems

Vijay Gupta
University of Notre Dame

Richard M. Murray
California Institute of Technology

Ling Shi
Hong Kong University of Science and Technology

Bruno Sinopoli
Carnegie Mellon University

DRAFT v1.1a, 20 December 2009

# Contents

# Preface

The area of "Networked Control Systems" has emerged over the past decade as a subdiscipline in control theory in which the flow of information in a system takes place across a communication network. Unlike traditional control systems, where computation and communications are usually ignored, the approaches that have been developed for networked control systems explicitly take into account various aspects of the communication channels that interconnect different parts of the overall system and the nature of the distributed computation that follows from this structure. This leads to a new set of tools and techniques for analysis and design of networked control systems that builds on the rich frameworks of communication theory, computer science and control theory.

This book is based on a series of courses that the authors have developed over the past several years, starting with a joint course taught at Caltech in Spring 2006. These courses were typically taken by students who have a good grounding in the basic techniques of control systems but may not have a strong background in computer science or some aspects of communication theory. While the level of mathematical detail in the book should allow it to be accessible to juniors or seniors in engineering, the treatment is tuned for first and second year graduate students in engineering or computer science. Some tutorial material on estimation theory is included, as well as a brief review of key concepts in graph theory that are needed primarily in the second half of the text.

The book is intended for researchers who are interested in the analysis and design of sensing, estimation and control systems in a networked setting. We focus primary on the effects of the network on the stability and performance of the system, including the effects of packet loss, time delay† and distributed computation. We have attempted to provide a broad view of the field, in the hope that the text will be useful to a wide crossection of researchers. Most of the results are presented in the discrete time setting, with references to the literature for the continuous time analogs. We have also attempted to include a review of the current literature at the end of each chapter, with an emphasis on papers that are frequently referenced by others, along with some directions for future research, when appropriate. To keep the material focused, we have chosen to only touch on material on optimization-based control (e.g., receding horizon control) or protocols for distributed systems, although these are often an integral part of complex

**RMM**: Check to make sure this is correct in final version networked control systems. References to the literature are given for readers interested in these important topics.

**RMM**: Currently written as two parts. Extend to three if we add back in the application chapters.

The book is organized into two main parts: a set of background chapters and the core material. Chapter 1 gives an introduction to the topic of networked control systems, including some driving application examples. Chapters 2–4 cover a collection of topics that are used throughout the remainder of the text. We assume familiarity with standard topics in estimation and control theory, including random processes, Kalman filtering and linear state space control theory, and provide only a quick review of this material in Chapter 2 to define the notation we will use and present some of the basic definitions and formulas. Chapters 3 and 4 complete the background chapters by giving concise overviews of the relevant results in information theory and Markovian jump linear systems, on which many of the later results of the book are built. These background chapters can be reviewed quickly for students and researchers already familiar with this material.

The core material on networked control systems is presented in Chapters 5 through 9. We begin by looking at the case of sensing, estimation and control of a single process across a communication channel, beging with the effects of rate limits in the channel in Chapter 5 and then the effects of packet loss in Chapter 6. Both of these chapters considers the cases where the communication channel affects on the measurements received from the sensor and where the channel affects both the measurements and the actuation commands. In Chapter 7 we begin to look at the problem of control over a graph, starting with an introduction to graph theory and the problem of consensus. Chapters 8 and 9 then go on to consider the distributed estimation and control problems, where one can have multiple processes, sensors, actuators, estimators and controllers distributed over a communications network. In each of these chapters on the core material we have attempted to present a unified view of many of the most recent and relevant results in network control, with the goal of establishing a foundation on which more specialized results of interest to specific groups can be covered.

The topics in the text have been taught by the authors and our colleagues in a variety of formats. In a semester-long, graduate course, it should be possible to cover most of the material in the book, assuming the students have good working knowledge of random processes, estimation theory and linear control systems. We have also used the material in the text for week-long short courses for masters and PhD students, where we cover the results in the background chapters in four 90 minute lectures, then spend 1–2 lectures on each of the remaining chapters. The material is fairly modular, so that the order of teaching the material can be varied according to the tastes of the instructor. The dependencies of the chapters are shown in Figure 1.

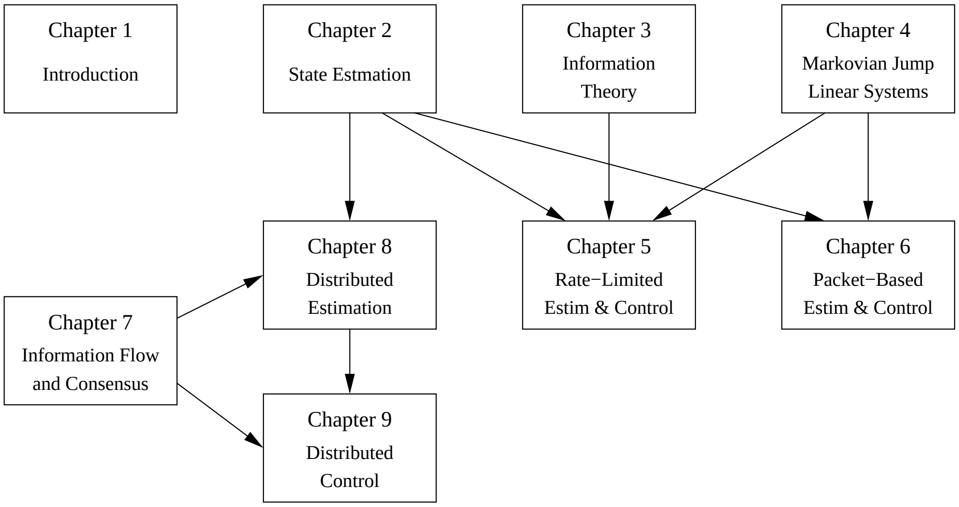


**Figure 1:** Dependencies of the chapters in the text.

Supplement

# Notation

This is an internal chapter that is intended for use by the authors in fixing the notation that is used throughout the text. In the first pass of the book we are anticipating several conflicts in notation and the notes here may be useful to early users of the text.

### General mathematics

- Use $*$ for expressions that are not given explicity
- Matrix transpose: $A^T$

### System dynamics

We focus on linear discrete time systems

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + v_k. \end{aligned}$$

The system is described by the state $x \in \mathbb{R}^n$, inputs $u \in \mathbb{R}^p$ and outputs $y \in \mathbb{R}^m$. Disturbances are represented by the random process $w_k$, which we typically take to be zero mean, white Gaussian noise with covariance matrix $\Sigma_W \geq 0$. Measurement noise is represented by the Gaussian random process $v_k$ with covariance matrix $\Sigma_V > 0$. For systems with multiple sensors, we use the notation $y^j$ to represent the $j$th output and use corresponding superscripts for the other relevant quantities.

In the few instances that we use continuous time dyanmics, these are written as

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu + w \\ y &= Cx + v \end{aligned}$$

Note that for both the continuous and discrete dynamics we leave out the direct term $(Du)$. We should point out the first time these equations come up in a chapter whether it is easy to include the direct term or whether not everything extends directly.

We currently have a discrepency in the notation described above. Some co-authors use $x_k$, others use $x(k)$. Need to resolve at some point in the near future. **RMM**

### Random variables and processes

- Expectation: $\mathbb{E}[X]$, $\mathbb{E}_Y[X]$ for the expectation of $X$ over $Y$.

- Mean: $\mu$, $\mu_X$ for a random variable/vector $X$

- Variance: $\sigma^2$, $\sigma^2_X$ for a (scalar) random variable $X$; $\Sigma$, $\Sigma_X$ for a random vector $X$; $\Sigma_{XY}$ for the cross-covariance

### Additional mathematical notation

- Lists and sets: A index set is can be written inline as $\{X_i : i = \text{min}, \ldots, \text{max}\}$ or as a displayed equation:

$$\{X_i\}_{i=\text{min}}^{\text{max}}.$$

### Observer dynamics

**All** We need to discuss this notation and think through what will work the best for what we want. The notation here might get cumbersome.

**RMM**: Notation for the contribution for the disturbance covariance is awkward. Since we usually don't include the time dependence, should be OK

We write $P > 0$ for the covariance of the estimation error. The observer for a discrete time linear system is written as a prediction step,

$$\begin{aligned}\hat{x}_{k|k-1} &= A\hat{x}_{k-1|k-1} + Bu_{k-1},\\ P_{k|k-1} &= AP_{k-1|k-1}A^T + \Sigma_{W\,k-1},\end{aligned}$$

followed by a correction step,

$$\begin{aligned}\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}),\\ P_{k|k} &= P_{k|k-1} - P_{k|k-1}C^T(CP_{k|k-1}C^T + R_k)^{-1}CP_{k|k-1}.\end{aligned}$$

**RMM**: Discuss at next telecon

The gain matrix for the estimator is given by $K$ (for Kalman). The gain matrix for a state space controller can either by $L$ or possibly $F$ (?).†

### Macros

Several macros have been defined to help enforce the notation described above.

| Macro | Symbol | Comments |
|---|---|---|
| `\reals` | $\mathbb{R}$ | Real numbers |
| `k\mns1` | $k{-}1$ | Compressed spacing |
| `k\pls1` | $k{+}1$ | Compressed spacing |
| `\Prob(A)` | $P(A)$ | Probability of an event |
| `\ctrlgain` | $L$ | State space control gain |
| `\obsvgain` | $K$ | Observer/Kalman gain |
| `\dvar x[k]` | $x_k$ | Discrete-time variable |
| `\cvar x[t]` | $x(t)$ | Continuous-time variable |
| `\pdf{}, \pdf[X]` | $p, p_X$ | Probability density function |
| `\expect{X}, \expect[Y]{X}` | $\mathbb{E}[X], \mathbb{E}_Y[X]$ | Expectation |
| `\avg{X}` | $\langle X \rangle$ | Average (not used) |
| `\mean{}, \mean[X]` | $\mu, \mu_X$ | Mean |
| `\stddev{}, \stddev[X]` | $\sigma, \sigma_X$ | Standard deviation |
| `\varnce{}, \varnce[X]` | $\sigma^2, \sigma^2_X$ | Variance |
| `\covar{}, \covar[X]` | $\Sigma, \Sigma_X$ | Covariance matrix |

Examples of how to use these expressions, especially in combination, can be found in `notation.tex`.

### Action items, notes and supplemental text

There are a macros available for marking up the authors version of the text (`ncsauthor.tex`):

- Action items† mark places in the text where changes need to be made. The owner of the action item should be the person who is expected to make the change.

**RMM**: like this

- Notes† can be placed in the text to leave information about a change or a decision that was made on what to include. The note should include the author of the note and a date.

Note: like this [RMM, 21 Dec 09]

- Supplemental information (in green, like this entire section) can be mark material that might be included at later points in time or that will be integrated into other parts of the book.

All of these markups are turned off in the version of the book that will be distributed to others (`ncsbook.tex`).

*Action items.* The following commands can be used to insert action items into the text:

```
\action[vshift]{Owner}{Action}    % action item with mark in text
\action*[vshift]{Owner}{Action}   % generate action item with no mark
\actionpar[vshift]{Owner}{Action} % generate an action item paragraph
```

*Notes.* The following commands can be used to insert notes into the text:

```
\actnote[Title]{Owner}{Note}     % generate a note in the text
\actnote*[Title]{Owner}{Note}    % generate a note with no mark
\actnotepar[Title]{Owner}{Note}  % generate a note paragraph
```

*Supplemental information.* Use the `supplement` environment to include supplemental information:

```
\begin{supplement}
Supplemental text...
\end{supplement}
```

# Chapter 1
## Introduction

Networked control is an emerging area of control theory driven by the increasing design, implementation and operation of control systems that makes use of communication networks to send information between the sensors, actuators and computational elements that make up a control system. In this chapter we provide an introduction to networked control systems (NCS), including a description of what is different about networked control versus traditional control design, some of the applications that are driving networked control systems research and engineering, and a list of some of the key subproblems in networked control that are the focus on the material in this text.

## 1.1 Overview of Networked Control Systems

Exand the paragraph below to talk more about the “standard” control problem that is solved, perhaps including an equation like $\dot{x} = f(x, u)$ to make things concrete. Mention sensing, estimation and control, so that these can be revisited for networked control systems. RMM

Modern control theory is largely based on the abstraction that information (“signals”) are transmitted along perfect communication channels and that computation is either instantaneous (continuous time) or periodic (discrete time). This abstraction has served the field well for 50 years and has led to many success stories in a wide variety of applications.

Future applications of control will be much more information-rich than those of the past and will involve networked communications, distributed computing, and higher levels of logic and decision-making (see [Mur03] for a recent analysis of future directions in this area). New theory, algorithms, and demonstrations must be developed in which the basic input/output signals are data packets that may arrive at variable times, not necessarily in order, and sometimes not at all. Networks between sensors, actuation, and computation must be taken into account, and algorithms must address the tradeoff between accuracy and computation time. Progress will require significantly more interaction between information theory, computer science, and control than ever before.

An emerging framework for networked control systems is shown in Figure 1.1. This architecture separates the traditional elements of sensing,

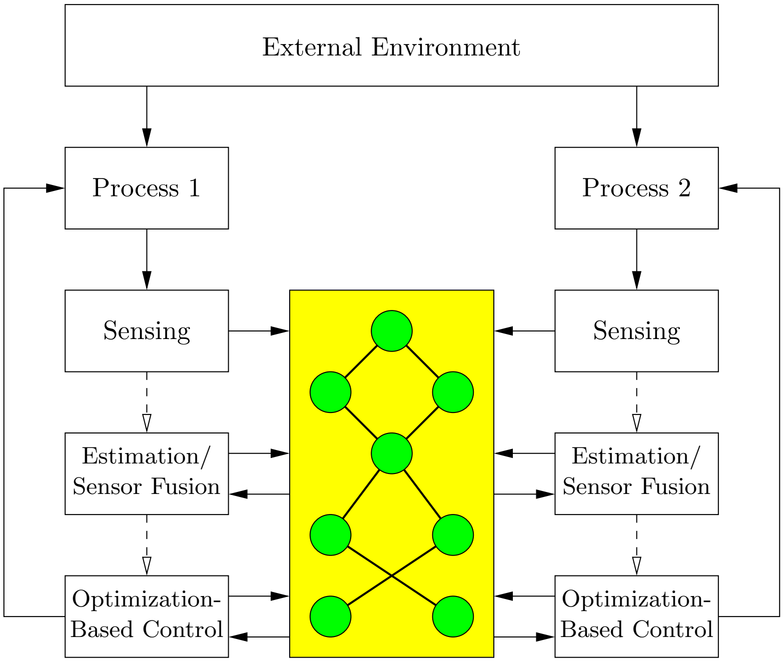

External Environment
Process 1
Process 2
Sensing
Sensing
Estimation/ Sensor Fusion
Estimation/ Sensor Fusion
Optimization-Based Control
Optimization-Based Control

**Figure 1.1:** Emerging framework for networked control systems. Signals between control system modules for multiple processes are transmitted through a communication network.

estimation, control, and actuation for a given system across a network and also allows sharing of information between systems. As we will see in the examples below, careful decisions need to be made on how the individual components in this architecture are implemented and how the communications across the networked elements is managed. This architecture can be used to model either a single system (using either half of the diagram) or multiple systems that interact through the network.

**RMM** Add material on "network effects", including packets, synchronization and distributed data. Talk also about emerging network protocols, such as time-triggered, event-triggered and rate-constrained protocols, plus multi-hop (wireless) networks. Probably a paragraph or two on each topic, with appropriate pictures and diagrams.

**RMM** As a final paragraph or two, talk about some of the problems that we *won't* convert in the book. This would include things like asynchronous execution, protocol-based control systems (ala CCL) and higher-level decision making (learning, goal management, fault recovery).

## 1.2 Application Examples

### Embedded Systems

One example of the use of this architecture is autonomous operations for sensor-rich systems, such as unmanned, autonomous vehicles. As part of the 2004 and 2005 DARPA Grand Challenges, Caltech has developed two such vehicles ("Bob" and "Alice") that each make use of a networked control systems architecture. Alice, the 2005 vehicle, has six cameras, 4 LADAR units, an inertial meaurement unit (IMU), a GPS navigation system, and numerous internal temperature and vibration sensors. The raw data rate for Alice is approximately 1–3 Gb/s, which must be processed and acted upon at rates of up to 100 Hz in order to insure safe operation at high driving speeds.

The control system for Alice makes use of the architecture depicted in Figure 1.1, with distributed data fusion to determine elevation maps (for the height of the terrain in front of the vehicle), multiple optimization-based controllers to plan possible routes for the vehicle, and online modeling, fault management, and decision making to provide reliable and reconfigurable operation. Eight onboard computers distribute the computational load, sharing information at mixed rates across a 1 Gb/s switched network. System specifications call for reliable operation in the presence of up to 1 computer failure and 2 sensor failures, requiring careful coordination between computational elements.

A major challenge in Alice is determining how to send information between nodes. Because of the high data rates and computational loads on the CPUs, packets sent across the network are not always received and the system must be robust to various networking effects. The choice of protocols and design of the overall messaging system is currently informal and based on trial and error. As an example of the issues that must be resolved, certain packets of data are very important, such as packets containing raw sensor information from a portion of the terrain that is scanned only once. Other data can be dropped if needed, such as commanded trajectories (the old trajectory can be used for several sampling periods). Data from the inertial measurement unit must be received with minimum latency, while other data (a change in the temperature of the vehicle) is much less time critical. Substantial effort has been put into trying to make sure that the computations and network protocols complement each other and that loss of data and data latency does not degrade the performance of the system.

The material below should be shortened substantially since it is not a major focus of the book. The main reason to include it would be to talk about some of the "higher level" functions that set on top of an NCS architecture. **RMM**

A more detailed architecture for a networked control system is shown in Figure 1.2. At the top of the figure, the standard elements for a control

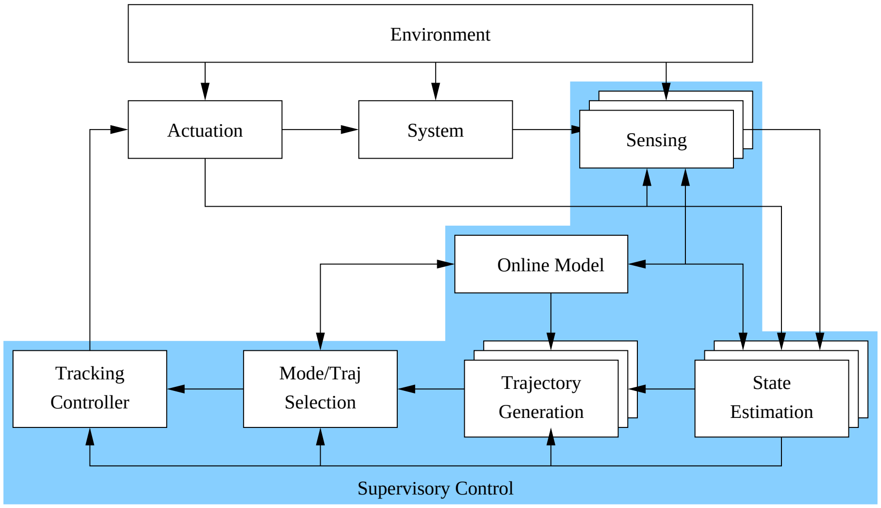


**Figure 1.2:** A detailed architecture for a networked control system, based on the control system for Alice [**?**].

system are present: actuation, system dynamics, sensing and environmental disturbances and noise. For many networked control systems, the amount of sensory information available is very large, requiring care in how this information is transmitted. Alice, for example, had between 1 and 3 gigabits/second (Gb/s) raw data rate, depending on the sensor suite taht was used. Another difference with traditional control systems is that the actuation subsystems are themselves embedded systems, capable of some amount of computation and local memory storage.

The primary control loop in a networked control system consists of state estimation, trajectory generation, and trajectory tracking. These elements can all represent relatively substantial computations (depending on the application) and are linked to each other through a number of network ports. In Alice, for example, the state estimation modules included a traditional inertial state estimator (combining GPS data with gryos and accelerometer measurements) as well as four computers that were estimating terrain information and computing a fused "speed map" that described the maximum allowable velocity that could be used in a given area of the terrain in front of it (more details on the software for Alice is given in Appendix **??**).

The information from the state estimators is used by trajectory generation algorithms that compute the desired state and inputs for the system to accomplish a task or minimize a cost function. The trajectory generation algorithms are responsible for taking into account actuator and state constraints on the sytem, as well as the nonlinear nature of the underlying process dynamics. A typical approach for these algorithms is to perform optimization-based control, in which one attempts to minimize a cost function subject to satisfying the constraints and dynamics. With the advances in computational power, it is often possible to run these optimization-based planners quickly enough that they can recompute the path from the current location in a "receding horizon" fashion, allowing feedback at the planner level. This is particularly useful to manage uncertainty in the cost function, for example when the cost is determined in real-time (as in the case of Alice, where the cost is based no the terrain that is being traversed).

As in the case of state estimation, networked control systems often use more than one trajectory generation algorithm running simultaneously. Since the physical system can only track one trajectory, some level of mode management and trajectory selection is required. This mode or trajectory selection logic is often under the control of higher levels of decision making (supervisory control).

The last element of the primary control loop is the trajectory tracking module, which is responsible for high frequency disturbance rejection and tracking. This module is itself a feedback system, using the state estimate and the desired trajectory to compute the actuation commands. In the context of a networked control system, the primary difference with traditional trajectory tracking algorithms is the need to run in a asynchronous execution environment, where reference trajectories and sensory measurements may come in at varying rates, including short periods where no inputs may arrive (due to network delays, computational delays or fault handling in one of the other modules).

In addition to the elements of the primary control loop, networked control systems can also contain a number of modules responsible for higher levels of decision making. We loosely refer to these modules as "supervisory control": they are responsible for implementing various control system functions that involve choosing parameters used by the primary control loop (such as cost functions and communication rates), dealing with failures of hardware and software components, maintaining an online model of the system dynamics, and adapting the performance of the system based on observed behaviors and memory. While these elements are critical for the operation of a networked control systems, in this text we focus on the primary control loop, where the network effects are most directly relevant.

### Sensor networks

Pull some of the material from the old sensor networks applications chapter here. We should limit the material to 1–2 pages, including some pictures. **LS**

### Process Control

This section will discuss some of the emerging applications of NCS in process control and manufacturing. Kalle Johansson has some very nice examples **VG**

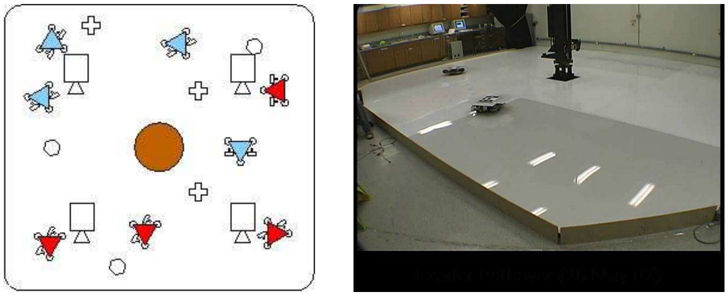


**Figure 1.3:** The Caltech Multi-Vehicle Wireless Testbed. The left figure shows the layout of the testbed area, including overhead cameras and fixed communication nodes (crosses and hexagons). The right picture is the current laboratory, with two vehicles shown.

that we can probably use here. We should try to write something up during the EECI, perhaps by taking notes on some of Vijay's lectures.

### Cooperative Control

**RMM** Update the material here to talk about two case: decoupled dynamics (multi-vehicle systems) and coupled dynamics (power grid, Internet). Much of the material can come from the previous application chapter on cooperative control. Can probably get rid of the MVWT example per se, since the other examples are better.

Another example of a networked control system is illustrated by the Caltech Multi-Vehicle Wireless Testbed (MVWT, shown in Figure 1.3), which consists of a collection of 8-12 vehicles performing cooperative tasks. The MVWT represents a slightly different instantiation of the architecture in Figure 1.1: each vehicle has a single processor with full access to local sensing and actuation, but information between vehicles must be sent across the network. The wireless commmuncation channels can exhibit significant degredation when multiple vehicles are attempting to communication and packet loss rates of 5-15% are not uncommon.

The issues in desiging a cooperative control policy for the MVWT vehicles faces many of the same challenges as those seen in Alice. Information communicated between vehicles can be dropped, reordered or sent with variable delay. Sensor information required for overall situational awareness can be fused at multiple levels and/or in a distributed fashion. Again, the currently available protocols for network communications are not well tuned to operation in this type of environment. For example, bit errors in packets can result in losing the entire data packet, rather than passing the information to the applications layer where partial (lossy) information could still be used effectively.

## 1.3 NCS Subproblems

This section will try to talk about the different class of NCS problems that will be studied in the book. This includes: **RMM**

- One- and two-block estimation and control
- Distributed sensing and estimation
- Distributed and cooperative control

# Chapter 2
## State Estimation and Sensor Fusion

Add an introductory paragraph describing what is in this chapter, and how it fits with the flow of the book. Since this is the first technical chapter, might want to say a bit about the next three chapters providing the background mathematics that we use throughout the text. **LS**

## 2.1 Review of Probability and Random Process

Shorten this section to just cover the main points and define the notation that we will use. I have put this in a separate file `estim/random.tex` so that we can later move it to an appendix if we decide to do so. **LS**

Other things to clean up in this chapter:

- Get rid of subsubsections
- Remove references to sigma fields; we won't need these
- Add material on stability of stochastic systems (`stability.tex`)

We assume the readers have some exposure to the theory of probability and random process. The material presented in this section only serves as a quick review of some basic concepts and tools from probability and random process that will be helpful to understand and derive some important results in subsequent sections and chapters. Good introductory books on probability and random process are:

provide some references here. **LS**

### Random Variables

*Probability Space*

Consider an experiment with many (possibly infinite) outcomes. All these outcomes form the *sample space* $\Omega$. A subset $A \subset \Omega$ is called an *event*. Two events $A_1, A_2$ are called *mutually disjoint* if $A_1 \cap A_2 = \emptyset$. The *complement* of an event $A$ is defined as $\bar{A} = \Omega \setminus A$.

A collection $\mathcal{F}$ of subsets of $\Omega$ is called a $\sigma-$**field** if it satisfies the following conditions:

1. $\emptyset \in \mathcal{F}$.
2. if $A_i \in \mathcal{F}, i = 1, 2, \ldots$, then $\cup_{i=1}^{\infty} A_i \in \mathcal{F}$.
3. if $A \in \mathcal{F}$, then $\bar{A} \in \mathcal{F}$.

A *probability measure* $P(\cdot)$ is a mapping from a $\sigma-$**field** $\mathcal{F}$ into the interval $[0, 1]$ such that the following axioms of probability are satisfied:

1. $P(A) \geq 0$ for all $A \subset \Omega$.
2. $P(\Omega) = 1$.
3. If $\{A_i, i = 1, 2, \ldots\}$ is a collection of disjoint members of $\mathcal{F}$, i.e., $A_i \cap A_j = \emptyset$ for all $i, j$, then $P(\cup A_i) = \sum_i P(A_i)$.

The triple $(\Omega, \mathcal{F}, P)$ is called a *probability space*. From the axioms of probability, it follows that

$$P(A) \leq 1, \quad P(\emptyset) = 0, \quad P(\bar{A}) = 1 - P(A), \quad P(\cup A_i) \leq \sum_i P(A_i).$$

### *Conditional Probability, Independence, and Bayes' Rule*

The joint probability of two events $A$ and $B$ is $P(A \cap B)$ which is often written as $P(AB)$ for simplicity. The conditional probability of $A$ given $B$ i.e., the probability that $A$ occurs if $B$ occurs in an experiment is

$$P(A|B) = \frac{P(AB)}{P(B)}, \quad \text{assuming } P(B) \neq 0.$$

$A$ and $B$ are *mutually independent* if

$$P(AB) = P(A)P(B).$$

If $P(B) \neq 0$, the conditional probability $P(A|B)$ can be calculated from *Bayes' Rule* as

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

If $A_i, i = 1, 2, \ldots$ are mutually disjoint and $\cup A_i = \Omega$, then

$$P(B) = \sum_i P(B|A_i)P(A_i)$$

and

$$P(A_j|B) = \frac{P(B|A_j)P(A_j)}{\sum_i P(B|A_i)P(A_i)}.$$

### *Random Variable*

A *random variable* is a function $X : \Omega \to \mathbb{R}$ with the property that

$$\{\omega \in \Omega : X(\omega) \leq x\} \in \mathcal{F} \text{ for each } x \in \mathbb{R}.$$

The *cumulative distribution function* of a random variable $X$ is a function $F_X : R \to [0,1]$ given by

$$F_X(x) = P(X \leq x).$$

The cumulative distribution function $F$ has the following properties

1. $\lim_{x\to-\infty} F_X(x) = 0$ and $\lim_{x\to\infty} F_X(x) = 1$.
2. If $x \leq y$, then $F_X(x) \leq F_X(y)$.
3. $F_X$ is right-continuous.

When $F_X$ is differentiable, we can define the associated *probability density function* $p_X(x)$ as

$$p_X(x) = \frac{dF_X(x)}{dx}.$$

The *joint cumulative distribution function* of two random variables $X$ and $Y$, denoted as $F_{XY}(x,y)$, is given by

$$F_{XY}(x,y) = P(X \leq x) \cap P(Y \leq y).$$

If its derivative exists, the associated joint probability density function is given by

$$p_{XY}(x,y) = \frac{\partial^2}{\partial x \partial y} F_{XY}(x,y).$$

The definition and results extend trivially to three or more random variables.

Given $F_{XY}(x,y)$, the *marginal distribution functions* of $X$ and $Y$ can be calculated as

$$F_X(x) = P(X \leq x) = F_{XY}(x,\infty),\ F_Y(y) = P(Y \leq y) = F_{XY}(\infty,y).$$

It follows that the *marginal density functions* of $X$ and $Y$ are

$$p_X(x) = \int_{-\infty}^{\infty} F_{XY}(x,y)dy,\ p_Y(y) = \int_{-\infty}^{\infty} F_{XY}(x,y)dx.$$

The *conditional density function* of $X$ given $Y$ is given by

$$p_{X|Y}(x|y) = \frac{p_{XY}(x,y)}{p_Y(y)}.$$

The density function of $X$ can also be calculated as

$$p_X(x) = \int_{-\infty}^{\infty} p_{X|Y}(x|y) p_Y(y) dy.$$

If $X$ and $Y$ are independent random variables, then the following statements holds and are equivalent to each other:

1. $F_{XY}(x, y) = F_X(x)F_Y(y)$.
2. $p_{XY}(x, y) = p_X(x)p_Y(y)$.
3. $p_{X|Y}(x|y) = p_X(x)$.

### *Statistical Properties of a Random Variable*

A random variable $X$ is completely specified by its distribution function $F_X(x)$ or density function $p_X(x)$. In many situations, $F_X(x)$ or $p_X(x)$ are difficult to obtain. It turns out the *mean* $\mu_X$ and *variance* $\sigma_X^2$ may provide us enough (useful) information about $X$. The mean and variance of a random variable $X$ are defined as follows:

$$\begin{aligned}\mu_X &= \mathbb{E}[X] = \int_{-\infty}^{\infty} x p_X(x)dx,\\ \sigma_X^2 &= \mathbb{E}\big[(X - \mathbb{E}[X])^2\big] = \int_{-\infty}^{\infty} (X - \mathbb{E}[X])^2 p_X(x)dx.\end{aligned}$$

We denote $\mathbb{E}[\cdot]$ as the *expectation operator*. Since $\mathbb{E}[\cdot]$ is a linear operator, $\sigma_X^2$ can also be calculated as

$$\sigma_X^2 = \mathbb{E}[X^2] - \big(\mathbb{E}[X]\big)^2.$$

If $X$ is a zero-mean random variable, i.e., $\mathbb{E}[X] = 0$, then $\sigma_X = \mathbb{E}[X^2]$. The $k$th *moment* of $X$ is $m_k = \mathbb{E}[X^k]$ and the $k$th *central moment* is $\mu_k = \mathbb{E}\big[(X - \mathbb{E}[X])^k\big]$.

The *covariance* of two random variables $X$ and $Y$ is defined as $\mathbb{E}\big[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])\big]$. $X$ and $Y$ are *uncorrelated* if $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$. If $X$ and $Y$ are uncorrelated, it is easy to verify that the covariance of $X$ and $Y$ is equal to zero. Clearly if $X$ and $Y$ are independent, then they are uncorrelated. However the converse does not hold in general.

**LS** Introduce conditional distribution function first.

The conditional expectation of $X$ given $Y = y$ is

$$\mathbb{E}[X|Y = y] = \int_{-\infty}^{\infty} x p_{X|Y}(x|y)dx$$

which is a number that depends on the value of $y$. Similarly, the conditional expectation of $X$ given $Y$ is

$$\mathbb{E}[X|Y] = \int_{-\infty}^{\infty} x p_{X|Y}(x|Y)dx$$

which is also a random variable that depends on $Y$, i.e., it is a *function of the random variable* $Y$. The following property is very important and has great practical value in evaluating $\mathbb{E}[X]$:

$$\mathbb{E}[X] = \mathbb{E}_Y\big[\mathbb{E}_X[X|Y]\big],$$

i.e., we first find the conditional expectation of $X$ (conditioned on $Y$), and then remove the condition by taking the expectation with respect to $Y$. From this property, one can easily verify that if $X$ and $Y$ are independent, then

$$\mathbb{E}[X|Y] = \mathbb{E}[X].$$

Furthermore if $X$ and $Y$ are jointly independently of $Z$, then

$$\mathbb{E}[XY|Z] = \mathbb{E}[X|Z]\mathbb{E}[Y|Z].$$

### Random Processes

A random process $X(t)$ is a generalization of a random variable. For a random variable, each experiment leads to a number (or a vector), while for a random process, each experiment leads to a function. For a fixed outcome $\omega \in \Omega$, one obtains the function $X(t, \omega)$, which is also called the *sample path* or *sample function* of the process. For a fixed $t$, $X(t, \omega)$ is a random variable with the underlying probability space $\Omega$. The *mean process* of $X(t)$ is the time function $\mathbb{E}[X(t)]$. The autocorrelation of $X(t)$ is $\mathbb{E}[X(t_1)X(t_2)^T]$ and the autocovariance of $X(t)$ is $\mathbb{E}\big[\big(X(t_1) - m(t_1)\big)\big(X(t_2) - m(t_2)\big)^T\big]$.

#### *Gaussian Random Variable and Random Process*

A random process $X(t)$ is called a *Gaussian random process* if for any finite set $\{t_1, t_2, \ldots, t_N\}$, the random variables $\{X(t_1), X(t_2), \ldots, X(t_N)\}$ have a joint Gaussian distribution, i.e., their joint probability density function is given by

$$p_X(x) = \frac{1}{(2\pi)^{N/2}\sqrt{\det[\mathcal{C}_X]}}\exp\left[-\frac{1}{2}(x - m_X)^T \mathcal{C}_X^{-1}(x - m_X)\right] \tag{2.1}$$

where $m_X = [m_X(t_1)\ m_X(t_2)\ \ldots\ m_X(t_N)]^T$ is the mean vector and $\mathcal{C}_X = \big[\mathrm{cov}\big(X(t_i), X(t_j)\big)\big]$ is the covariance matrix. Gaussian processes have the following properties.

**Theorem 2.1.** *Let $X(t)$ be a Gaussian process. Then*

1. *$X(t)$ is completely determined by $m_X$ and $\mathcal{C}_X$.*

**Theorem 2.2.** *Let $X$ and $Y$ have a joint Gaussian distribution with mean and covariance given by*

$$\mu = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} \text{ and } \Sigma = \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{bmatrix}.$$

*Then $X$ conditioned on $Y = y$ is Gaussian with mean and covariance given by*

$$\mu_{X|Y=y} = \bar{x} + \Sigma_{xy}\Sigma_y^{-1}(y - \bar{y}) \text{ and } \Sigma_{X|Y=y}\Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{yx}.$$

*In other words,*

$$\mathbb{E}[X|Y = y] = \bar{x} + \Sigma_{xy}\Sigma_y^{-1}(y - \bar{y}). \tag{2.2}$$

The proof can be found in AndersonMoore1979.

### Stability of stochastic systems

**LS** This subsection should describe the different types of stability for a stochastic system (mean square, almost surely, convergence in distribution). I have put this in a separate file, `stability.tex`, just in case it gets large and needs to turn into its own section. We should include the definition of the Ricatti operator here (if it hasn't come up already) and also make sure to include Jensen's inequality.

### Sampling of a Continuous-time System

Note: This text is now in a separate file, `sampling.tex`, so that we can pull it out more easily if we decide we don't want to include it. [RMM, 21 Dec 09]

A wide variety of physical systems are modeled in the continuous-time domain. In this book, we focus on continuous-time systems with dynamics of the form

$$\frac{dx}{dt} = A_c x + B_c u + w, \quad y = C_c + v, \tag{2.3}$$

where $x(t) \in \mathbb{R}^n$ is the state vector with unknown initial value $x(0)$, $u(t) \in \mathbb{R}^p$ is the input vector, $y(t) \in \mathbb{R}^m$ is the observation vector, and $w(t)$ and $v(t)$ are process disturbance and measurement noise. We assume $w(t)$ and $v(t)$ are mutually uncorrelated zero-mean Gaussian processes with autocovariances

$$\mathbb{E}[w(s)w(t)^T] = \delta_{st}\Sigma_{wc}, \quad \mathbb{E}[v(s)v(t)^T] = \delta_{st}\Sigma_{vc},$$

where $\delta_{st} = 1$ if $s = t$ and $\delta_{st} = 0$ otherwise.

As more and more controllers are implemented digitally, we need a procedure to convert the continuous-time system (2.3) into an equivalent discrete-time system. This procedure is called *sampling* or *discretization*. A frequently seen approach to implement the control law on a digital computer is to use a digital to analogue converter that holds the analog signal until the next time step, called *zero-order-hold* control.

Consider the following periodic sampling scheme: we sample the system (2.3) at time instances $t = k\tau,\ k = 0, 1, \ldots$, where $\tau > 0$ is the *sampling period*. It can be shown (see Astrom-Wittenmark) that the equivalent discrete-time system of (2.3) is given by

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad y_k = Cx_k + v_k, \tag{2.4}$$

where $x_k$ and $y_k$ correspond to $x(t)$ and $y(t)$ at time $t = k\tau$, and $A, B$ and $C$ are given by

$$A = e^{A_c \tau}, \quad B = \int_0^\tau e^{A_c t} dt B_c, \quad C = C_c. \tag{2.5}$$

In the discrete-time setting, the process and measurement noises are also uncorrelated zero-mean Gaussian random processes with covariance

$$\mathbb{E}[w_s w_k] = \delta_{sk} \Sigma_w, \quad \mathbb{E}[v_s v_k^T] = \delta_{sk} \Sigma_v,$$

where

$$\Sigma_w = \int_0^\tau e^{A_c t} \Sigma_{wc} e^{A_c^T t} dt, \quad \Sigma_v = \Sigma_{vc}.$$

The following method from wikipedia for computing $\Sigma_w$ needs to be verified. **LS**

Computing $\Sigma_w$ directly from the above formula is sometimes difficult due to the integral of matrix exponentials. An easier approach to compute it is given as follows. Define $M$ and $N$ as

$$M = \begin{bmatrix} -A_c & \Sigma_{wc} \\ 0 & A_c^T \end{bmatrix} \tau, \quad N = e^M.$$

Then it is straightforward to show that

$$N = \begin{bmatrix} * & X^{-1} \Sigma_w \\ 0 & X^T \end{bmatrix}.$$

Therefore $\Sigma_w$ can be computed from

$$\Sigma_w = (X^T)^T X^{-1} \Sigma_w,$$

i.e., $\Sigma_w$ is obtained by multiplying the transpose of the lower-right submatrix of $N$ with the upper-right submatrix of $N$.

Most of the results developed in this book also extend straightforward to cases where the sensor measurement $y_k$ involves a direct input term, i.e.,

$$y_k = C x_k + D u_k + v_k. \tag{2.6}$$

For simplicity, we shall use the system model as described by (2.4) for the remainder of the book unless otherwise explicitly stated.

### Markov Chains

Write up this subsection, which should include the relevant results that we will need in later chapters. I have create this as a separate file, `markov.tex`, in case we want to move it around later. **VG**

## 2.2 Optimal Estimation

LS This section needs to be updated to reflect the new structure of the chapter. I have put the former MMSE and Kalman sections as subsections, and added a few additional subsections relevant to optimal estimation as well.

### Minimum Mean Square Error Estimator

Suppose we wish to know some quantity $X$, and we are not able to make a direct and accurate measurement of $X$. However we can make some indirect measurement $Y$ that is related to $X$. Our task is to get an "optimal" estimate of $X$ from $Y$.

One question that immediately arises before we attempt to solve the estimation problem is: what is a good estimate and when an estimate is "optimal"?

Intuitively a "good" estimate should make the estimation error $\hat{X} - X$ "small" since we wish to reconstruct $X$ as perfectly as possible. An "optimal" estimate should make $\hat{X} - X$ the "smallest" among all other estimates. Many metrics can be used to define the size of the error $\hat{X} - X$ (hence we are able to say if it is "small" or not). Since $\hat{X} - X$ is typically a random variable, the metric that we shall use throughout the book is the following *mean squared error* (MSE)

$$\mathbb{E}[(\hat{X} - X)^T(\hat{X} - X)].$$

Therefore given $Y = y$ (i.e., the measurement that we take), our task is to construct the optimal estimate $\hat{X}$ that minimizes

$$\mathbb{E}[(\hat{X} - X)^T(\hat{X} - X)|Y = y].$$

It turns out that the optimal $\hat{X}$ has a very simple form, given in the following theorem.

**Theorem 2.3.** *The optimal estimate $\hat{X}^*$ that minimizes*

$$\mathbb{E}[(\hat{X} - X)^T(\hat{X} - X)|Y = y]$$

*is given by the following conditional expectation of $X$*

$$\hat{X}^* = \mathbb{E}[X|Y = y].$$

*Proof.* We can rewrite $\mathbb{E}[(\hat{X} - X)^T(\hat{X} - X)|Y = y]$ as follows

$$\begin{aligned}
&\mathbb{E}[(\hat{X} - X)^T(\hat{X} - X)|Y = y] \\
=&\mathbb{E}[X^TX|Y = y] - 2\hat{X}^T\mathbb{E}[X|Y = y] + \hat{X}^T\hat{X} \\
=&\big(\hat{X} - \mathbb{E}[X|Y = y]\big)^T\big(\hat{X} - \mathbb{E}[X|Y = y]\big) + \mathbb{E}\big[X^TX - \mathbb{E}[X]^T\mathbb{E}[X^T]|Y = y\big].
\end{aligned}$$

Since $\mathbb{E}\big[X^T X - \mathbb{E}[X]^T \mathbb{E}[X^T] | Y = y\big]$ is independent of $\hat{X}$, we conclude that

$$\hat{X}^* = \mathbb{E}[X|Y = y].$$

□

$\hat{X}^* = \mathbb{E}[X|Y = y]$ is also called the *minimum mean squared error* (MMSE) estimate of $X$.

**Example 2.1 Estimate a Gaussian random variable**
Consider the following equation

$$Y = X + N \tag{2.7}$$

where $X$ and $N$ are both scalar zero-mean Gaussian random variables with covariances $\sigma_x$ and $\sigma_n$ respectively. Further assume $X$ and $N$ are uncorrelated. Suppose we make a measurement of $X$ and get $y$. The MMSE estimate of $X$ is then given by

$$\hat{X} = \mathbb{E}[X|Y = y] = \frac{\sigma_x}{\sigma_x + \sigma_n} y.$$

∇

### Kalman Filtering

Consider the following discrete-time linear time-invariant system

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad y_k = Cx_k + v_k, \tag{2.8}$$

where $x_k \in \mathbb{R}^n$ is the state vector with unknown initial value $x_0$, $u_k \in \mathbb{R}^p$ is the input vector, $y_k \in \mathbb{R}^m$ is the observation vector, and $w_k$ and $v_k$ are process and measurement noises (or disturbances).

Clearly nothing can be said on any estimator without defining a structure on $w_k$ and $v_k$. In this book, we are particularly interested in $w_k$ and $v_k$ that have the following properties:

- $w_k$ and $v_k$ are zero-mean Gaussian random vectors;
- $\mathbb{E}[w_k w_j^T] = \delta_{kj}\Sigma_w$ with $\Sigma_w \geq 0$;
- $\mathbb{E}[v_k v_j^T] = \delta_{kj}\Sigma_v$ with $\Sigma_v > 0$;
- $\mathbb{E}[w_k v_j^T] = 0 \; \forall j, k$,

where $\delta_{kj} = 0$ if $k \neq j$ and $\delta_{kj} = 1$ otherwise. We also assume the initial value $x_0$ of system (2.8) is a zero-mean Gaussian random vector that is uncorrelated with $w_k$ and $v_k$ for all $k \geq 0$. The covariance of $x_0$ is given by $\Pi_0 \geq 0$. Furthermore we assume $(A, \sqrt{Q})$ is stabilizable.

Let $Y_k = \{y_0, y_1, \ldots, y_k\}$ be the measurements available at time $k$ and $U_k = \{u_0, u_1, \ldots, u_k\}$ be the input applied to the system up to time $k$. We are interested in looking for the MMSE $\hat{x}_k$ of $x_k$ at each time $k \geq 0$ given $Y_k$ and $U_{k-1}$. From Theorem 2.3, we know that $\hat{x}_k$ is given by

$$\hat{x}_k = \mathbb{E}[x_k | Y_k, U_{k-1}], \tag{2.9}$$

and the corresponding error covariance $P_k$ is given by

$$P_k = \mathbb{E}[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T | Y_k, U_{k-1}]. \tag{2.10}$$

Calculating $\hat{x}_k$ and $P_k$ according to equation (2.9) and (2.10) is not trivial and is computationally intensive as $k$ increases. The celebrated Kalman filter provides a simple and elegant way to compute $\hat{x}_k$ and $P_k$ recursively.

**LS** put some introductory materials here, e.g., origin of KF, applications of KF, etc.

Assume that $\hat{x}_{k-1}$ and $P_{k-1}$ defined as in equation (2.9) and (2.10) are available. Consider the one-step state prediction $\hat{x}_{k|k-1}$ (also called the *a priori state estimate*) given by

$$\hat{x}_{k|k-1} = \mathbb{E}[x_k | Y_{k-1}, U_{k-1}]$$

and the associated estimation error covariance (also called the *a priori error covariance*) $P_{k|k-1}$ given by

$$P_{k|k-1} = \mathbb{E}[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^T | Y_{k-1}, U_{k-1}].$$

From (2.8), we have

$$\begin{aligned} \hat{x}_{k|k-1} &= \mathbb{E}[x_k | Y_{k-1}, U_{k-1}] \\ &= \mathbb{E}[A x_{k-1} + B u_{k-1} + w_{k-1} | Y_{k-1}, U_{k-1}] \\ &= A \hat{x}_{k-1} + B u_{k-1}, \end{aligned} \tag{2.11}$$

where we use the fact that $w_{k-1}$ is independent of any $y_t$ $(t \leq k-1)$ and the expectation operator is linear. Consequently,

$$P_{k|k-1} = A P_{k-1} A^T + \Sigma_w. \tag{2.12}$$

Now consider $y_k$ conditioned on $Y_{k-1}$ and $U_{k-1}$ which has mean

$$\mathbb{E}[y_k | Y_{k-1}, U_{k-1}] = \mathbb{E}[C x_k + v_k | Y_{k-1}, U_{k-1}] = C \hat{x}_{k|k-1}$$

and covariance

$$\mathbb{E}\big[\big(y_k - \mathbb{E}[y_k]\big)\big(y_k - \mathbb{E}[y_k]\big)^T | Y_{k-1}, U_{k-1}\big] = C P_{k|k-1} C^T + \Sigma_v,$$

where we have used the fact that $v_k$ is independent of $Y_{k-1}$. The cross covariance of $x_k$ and $y_k$ conditioned on $Y_{k-1}$ and $U_{k-1}$ is given by

$$\mathbb{E}\big[\big(x_k - \mathbb{E}[x_k]\big)\big(y_k - \mathbb{E}[y_k]\big)^T | Y_{k-1}, U_{k-1}\big] = P_{k|k-1} C^T.$$

From the above analysis, we see that the random vector $[x_k' \quad y_k']'$ conditioned on $Y_{k-1}$ and $U_{k-1}$ is Gaussian with mean and covariance

$$\begin{bmatrix} \hat{x}_{k|k-1} \\ C\hat{x}_{k|k-1} \end{bmatrix} \text{ and } \begin{bmatrix} P_{k|k-1} & P_{k|k-1}C^T \\ CP_{k|k-1} & CP_{k|k-1}C^T + \Sigma_v \end{bmatrix}.$$

Therefore from Theorem 2.2, $x_k$ conditioned on $y_k$ (and on $Y_{k-1}$ and $U_{k-1}$, i.e., conditioned on $Y_k$ and $U_{k-1}$) has mean

$$\mathbb{E}[x_k|Y_k, U_{k-1}] = \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1})$$

and covariance

$$(I - K_kC)P_{k|k-1}$$

where $K_k = P_{k|k-1}C^T[CP_{k|k-1}C^T + \Sigma_v]^{-1}$ is the so-called *Kalman gain*.

Let us summarize what we have said so far. Given the system (2.8), the MMSE estimate $\hat{x}_k$ of $x_k$ is given by $\hat{x}_k = \mathbb{E}[x_k|Y_k, U_{k-1}]$, which can be computed recursively as follows

1. *time update:*

$$\begin{aligned} \hat{x}_{k|k-1} &= A\hat{x}_{k-1} + Bu_{k-1}, \\ P_{k|k-1} &= AP_{k-1}A^T + \Sigma_w. \end{aligned}$$

2. *measurement update:*

$$\begin{aligned} K_k &= P_{k|k-1}C^T[CP_{k|k-1}C^T + \Sigma_v]^{-1}, \\ \hat{x}_k &= \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}), \\ P_k &= (I - K_kC)P_{k|k-1}. \end{aligned}$$

The initial values of the recursion are set as $\hat{x}_0 = 0$ and $P_0 = \Pi_0$. The Kalman filter essentially consists of the above two update steps.

put some discussions on Kalman filter here, e.g., properties of the filter, applications, etc. **LS**

**Lemma 2.1.** *The Kalman gain $K_k$ and the error covariance $P_k$ satisfy*

$$K_k = P_kC^T\Sigma_v^{-1}. \tag{2.13}$$

*Proof.* Since $P_k = (I - K_kC)P_{k|k-1}$, it suffices to show

$$(I - K_kC)P_{k|k-1}C^T\Sigma_v^{-1} = K_k$$

which is equivalent to

$$\begin{aligned} & P_{k|k-1}C^T\Sigma_v^{-1} = K_k(I + CP_{k|k-1}C^T\Sigma_v^{-1}) \\ \Longleftrightarrow\ & P_{k|k-1}C^T\Sigma_v^{-1} = P_{k|k-1}C^T[CP_{k|k-1}C^T + \Sigma_v]^{-1}(I + CP_{k|k-1}C^T\Sigma_v^{-1}) \\ \Longleftarrow\ & \Sigma_v = (I + CP_{k|k-1}C^T\Sigma_v^{-1})^{-1}(CP_{k|k-1}C^T + \Sigma_v) \\ \Longleftrightarrow\ & \Sigma_v = \Sigma_v(\Sigma_v + CP_{k|k-1}C^T)^{-1}(CP_{k|k-1}C^T + \Sigma_v) \end{aligned}$$

where the last equation holds trivially. □

Note: Alternative version of the proof (from my OBC notes) [RMM, 21 Dec 09]

*Alternate proof.* $K_k$ is defined as

$$K_k = P_{k|k-1}C^T(\Sigma_V + CP_{k|k-1}C^T)^{-1}.$$

Multiplying through by the inverse term on the right and expanding, we have

$$\begin{aligned} K_k(\Sigma_V + CP_{k|k-1}C^T) &= P_{k|k-1}C^T, \\ K_k\Sigma_V + K_kCP_{k|k-1}C^T &= P_{k|k-1}C^T, \end{aligned}$$

and hence

$$\begin{aligned} K_k\Sigma_V &= P_{k|k-1}C^T - K_kCP_{k|k-1}C^T, \\ &= (I - K_kC)P_{k|k-1}C^T = P_{k|k}C^T. \end{aligned}$$

The desired results follows by multiplying on the right by ${\Sigma_V}^{-1}$. □

To simplify the notations, let us define $h : \mathbb{S}_+^n \to \mathbb{S}_+^n$ as

$$h(X) \triangleq AXA^T + \Sigma_w, \tag{2.14}$$

and $\tilde{g} : \mathbb{S}_+^n \to \mathbb{S}_+^n$ as

$$\tilde{g}(X) \triangleq X - XC^T[CXC^T + \Sigma_v]^{-1}CX, \tag{2.15}$$

where $\mathbb{S}_+^n$ is the set of $n$ by $n$ positive semi-definite matrices. Further define $g : \mathbb{S}_+^n \to \mathbb{S}_+^n$ as

$$g(X) \triangleq h \circ \tilde{g} = AXA^T + \Sigma_w - AXC^T[CXC^T + \Sigma_v]^{-1}CXA. \tag{2.16}$$

For functions $f, f_1, f_2 : \mathbb{S}_+^n \to \mathbb{S}_+^n$, $f_1 \circ f_2$ is defined as

$$f_1 \circ f_2(X) \triangleq f_1\big(f_2(X)\big), \tag{2.17}$$

and $f^t$ is defined as

$$f^t(X) \triangleq \underbrace{f \circ f \circ \cdots \circ f}_{t \text{ times}}(X). \tag{2.18}$$

With these definitions, it is straightforward to verify that $P_{k+1|k}$ and $P_{k+1}$ satisfy

$$\begin{aligned} P_{k+1|k} &= h(P_k), \\ P_{k+1|k} &= g(P_{k|k-1}), \\ P_{k+1} &= \tilde{g}(P_{k+1|k}), \\ P_{k+1} &= \tilde{g} \circ h(P_k). \end{aligned}$$

The equation $g(X) = X$, i.e.,

$$AXA^T + \Sigma_w - AXC^T[CXC^T + \Sigma_v]^{-1}CXA = X \tag{2.19}$$

is called the *Discrete-time Algebraic Riccati Equation* (DARE).

### Properties of the Kalman filter

Pull the various lemmas that are currently in the packet-based estimation and control chapter but aren't specific to packet-based implementation and put them here. **LS**

## 2.3 Optimal Control

Write up this section, which should include a summary of the linear quadratic regulator problem, a statement of the separation principle (and proof, if it is useful for later material), and possibly a summary of dynamic programming. **VG**

## 2.4 Further Reading

### Exercises

**RMM**: This doesn't appear to be showing up in the exercises. Fix.

**2.1** Show $\mathbb{E}[X|Y=y] = \frac{\sigma_x}{\sigma_x+\sigma_n} y$ in Example 2.1.

# Chapter 3
## Information Theory

To be written: **VG**

- Motivation
- Basic definitions
- Relations
- Bode's formula for arbitrary feedback
- Performance bounds with feedback across a communication channel
- Information patterns
- Witsenhausen counterexample

# Chapter 4

# Markovian Jump Linear Systems

In this chapter, we present a short overview of Markovian jump linear systems. A more thorough and complete treatment is given in books such as [**?**]. As in other chapters, our focus will be on the Linear Quadratic Gaussian (LQG) control of such systems. As we shall see, even though such systems are non-linear, they can be analyzed using tools that are similar to those used in linear system analysis.

## 4.1 Introduction to Markovian Jump Linear Systems

A useful category of system models are those in which the system operates in multiple modes. Although each of the individual modes in linear, the switching between these modes introduces non-linearity into the overall system description. A general theory of such systems is developed in the hybrid systems community. However, much tighter results can be developed if a further assumptions holds, that the *mode switches are governed by a stochastic process that is statistically independent from the state values.* In the case when the stochastic process can be described by a Markov chain, the system is called a Markovian jump linear system. Although the individual modes of such systems may be continuous or discrete, we will concentrate on the latter case here.

More formally, consider a discrete time discrete state Markov process with state $r(k) \in \{1, 2, \cdots, m\}$ at time $k$. Denote the transition probability $\text{Prob}(r(k+1) = j | r(k) = i)$ by $q_{ij}$, and the resultant transition probability matrix by $Q$. We will assume that the Markov chain is irreducible and recurrent. Also denote

$$\text{Prob}(r(k) = j) = \pi_j(k),$$

with $\pi_j(0)$ as given. The evolution of a Markovian jump linear system (MJLS), denoted by $\mathcal{S}_1$ for future reference, can be described by the following equations

$$\begin{aligned} x(k+1) &= A_{r(k)}x(k) + B_{r(k)}u(k) + F_{r(k)}w(k) \qquad (4.1) \\ y(k) &= C_{r(k)}x(k) + G_{r(k)}v(k), \end{aligned}$$

where $w(k)$ is zero mean white Gaussian noise with covariance $R_w$, $v(k)$ is zero mean white Gaussian noise with covariance $R_v$ and the notation

$X_{r(k)}$ implies that the matrix $X \in \{X_1, X_2, \cdots, X_m\}$ with the matrix $X_i$ being chosen when $r(k) = i$. The initial state $x(0)$ is assumed to be a zero mean Gaussian random variable with variance $\Pi(0)$. For simplicity, we will consider $F_{r(k)} = G_{r(k)} \equiv I$ for all values of $r(k)$ in the sequel. We also assume that $x(0)$, $\{w(k)\}$, $\{v(k)\}$ and $\{r(k)\}$ are mutually independent. The particular case when $q_{ij} = q_j, \forall i, j$ (i.e., the random process governing the switching of the modes is a Bernoulli process) is sometimes referred to as a Bernoulli jump linear system.

Such systems have been studied for a long time in the fault isolation community, and have received new impetus with the advent of networked control systems. We now consider some examples of applicability of Markovian jump linear systems.

**Example 4.1**
Consider the following example of a failure prone production system, which is the discrete time equivalent of the model presented in [AK86]. Consider a manufacturing system producing a single commodity. There is a constant demand rate $d$ for the commodity, and the goal of the manufacturing system is to try to meet this demand. The manufacturing system is, however, subject to occasional breakdowns and so at any time $k$, the system can be in one of two states: a functional ($r(k) = 1$) state and a breakdown ($r(k) = 2$) state. The transitions between these two states are usually modeled to occur as a Markov chain with given mean time between failures and mean repair time. When the manufacturing system is in the breakdown state it cannot produce the commodity, while if it is in the functional state it can produce at any rate $u$ up to a maximum production rate $r > d > 0$. Let $x(k)$ be the inventory of the commodity at time $k$, i.e., $x(k) =$ (total production up to time $k$) - (total demand up to time $k$). Then the system is a Markovian jump linear system that evolves as

$$x(k+1) = \begin{cases} x(k) + u(k) - d & r(k) = 1 \\ x(k) - d & r(k) = 2, \end{cases}$$

where $u(k)$ is the controlled production rate. A negative $x(k)$ denotes backlog, and $u(k)$ satisfies a saturation constraint. $\nabla$

**Example 4.2**
Consider a linear process evolving as

$$x(k+1) = Ax(k) + Bu(k) + w(k),$$

and being observed by a sensor of the form

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

The measurements from the sensor are transmitted to an estimator across an analog erasure link. At any time $k$, the estimator receives measurement $y(k)$ with probability $1-p$, and with a probability $p$ no measurement is received.

As discussed in another chapter, this is a common model for a dynamic process being estimated across an analog erasure channel. This is a Bernoulli jump linear system with two modes $r(k) \in \{0, 1\}$. For both the modes, the system matrices $A_0 = A_1 = A$ and $B_0 = B_1 = B$. Mode 0 corresponds to no measurement being received and for this case $C_0 = 0$. Mode 1 corresponds to measurement being received, and for this case $C_1 = C$. $\nabla$

## 4.2 Stability of Markovian jump linear systems

In this section, we discuss the stability of autonomous Markovian jump linear systems. We will see that the necessary and sufficient condition for stability can be presented an algebraic condition in terms of the spectral radius of a suitable matrix. We will also present an equivalent condition in terms of a linear matrix inequality.

Since an Markovian jump linear systems is a stochastically varying system, numerous notions of stability may be defined. We will primarily be interested in mean square stability. Thus, define the state covariance $C(k) = E[x(k)x^T(k)]$, where the expectation is taken with respect to the initial state, process and measurement noise, and the discrete modes till time $k$. The system is stable if the steady state covariance is bounded, i.e., if $\lim_{k\to\infty} C(k) < C^\star$, where $C^\star$ is a constant matrix, and the inequality is understood in the positive definite sense.

The stability condition for Markovian jump linear systems is given by the following result.

**Theorem 4.1.** *Consider the system $\mathcal{S}_1$ with the control input $u(k) = 0$. The system is stable if and only if the condition*

$$\rho\left((Q^T \otimes I)\,diag(A_i \otimes A_i)\right) < 1$$

*holds, where $\rho(M)$ is the spectral radius of matrix $M$, $Q$ is the transition probability matrix of the Markov chain governing the mode switches of the system, $\otimes$ denotes the Kronecker product, $I$ is the identity matrix of suitable dimensions, and $diag(A_i \otimes A_i)$ denotes a block diagonal matrix formed by using the matrices $A_i \otimes A_i$ for various mode values $i$.*

*Proof.* Consider the term

$$C^i(k) = E[x(k)x^T(k)|r(k) = i]\pi_i(k),$$

so that the covariance is given by

$$C(k) = \sum_{i=1}^{m} C^i(k).$$

We will study the evolution of terms $C^i(k)$. Conditioning on the state value at time $k-1$ yields

$$\begin{aligned} C^i(k) &= \sum_{j=1}^{m} \text{Prob}(r(k-1)=j|r(k)=i)\pi_i(k)E[x(k)x^T(k)|r(k)=i,r(k-1)=j] \\ &= \sum_{j=1}^{m} \text{Prob}(r(k)=i|r(k-1)=j)\pi_j(k-1)E[x(k)x^T(k)|r(k)=i,r(k-1)=j] \\ &= \sum_{j=1}^{m} q_{ji}\pi_j(k-1)E[x(k)x^T(k)|r(k-1)=j], \end{aligned}$$

where in the second line we have used the Bayes law, and in the third line we have used the fact that given the Markov mode at time $k-1$, $x(k)$ is conditionally independent of the Markov mode at time $k$. Now given the Markov mode at time $k-1$, the covariance of the state at time $k$ can be related to the covariance at time $k-1$. Thus, we obtain

$$\begin{aligned} C^i(k) &= \sum_{j=1}^{m} q_{ji}\pi_j(k-1)\Big(A_j E[x(k-1)x^T(k-1)|r(k-1)=j]A_j^T + R_w\Big) \\ &= \sum_{j=1}^{m} q_{ji}A_jC^j(k-1)A_j^T + \sum_{j=1}^{m} q_{ji}\pi_j(k-1)R_w. \end{aligned}$$

We can vectorize this equation and use the identity

$$\text{vec}(ABC) = (C^T \otimes A)\text{vec}(B)$$

to obtain

$$\text{vec}(C^i(k)) = \sum_{j=1}^{m} q_{ji}(A_j \otimes A_j)\text{vec}(C^j(k-1)) + \pi_i(k)\text{vec}(R_w). \tag{4.2}$$

For values of $i = 1, \cdots, m$, these coupled linear equations define the stability of $C(k)$. We can stack the vectors $\text{vec}(C^i(k))$ for various values of $i$, and obtain that the dynamical system recursion is governed by the matrix $\big((Q^T \otimes I)\text{diag}(A_i \otimes A_i)\big)$ . Thus, we need to consider the spectral radius of this matrix. □

For a Bernoulli jump linear system, the condition reduces to the following simple form.

**Theorem 4.2.** *Consider the system $\mathcal{S}_1$ with the control input $u(k) = 0$ and the additional assumption that the Markov transition probability matrix is such that for all states $i$ and $j$, $q_{ij} = q_i$. The system is stable if and only if the condition*

$$\rho\left(E[A_i \otimes A_i]\right) < 1$$

*holds, where the expectation is taken over the probabilities $\{q_i\}$.*

*Proof.* In this case, we have $q_{ij} = q_j, \forall i$. Moreover, $r(k)$ and $x(k)$ are independent, so that $C^i(k) = C(k)\pi_i(k) = C(k)q_i(k)$. Thus, (4.2) yields

$$\begin{aligned}\mathrm{vec}(C(k)) &= \sum_{j=1}^{m}(A_j \otimes A_j)\mathrm{vec}(C(k-1))q_j(k) + \mathrm{vec}(R_w)\\ &= E[A_i \otimes A_i]\mathrm{vec}(C(k-1)) + \mathrm{vec}(R_w),\end{aligned}$$

which yields the desired stability condition. □

Even though the above conditions are simple to write, the calculation of the spectral value may grow computationally expensive as the number of Markov states increases. We can present an alternate condition in terms of a linear matrix inequality as follows.

**Theorem 4.3.**

*Proof.* □

## 4.3 LQG control

We will develop the LQG controller of Markovian jump linear systems in three steps. We will begin by considering the optimal linear quadratic regulator. We will then consider the optimal estimation problem for Markovian jump linear systems in the minimum mean squared error (MMSE). Finally, we will present a separation principle that will allow us to solve the LQG problem as a combination of the above filters.

### Optimal Linear Quadratic Regulator

The Linear Quadratic Regulator (LQR) problem for the system $\mathcal{S}_1$ is posed by assuming that the noises $w(k)$ and $v(k)$ are not present. Moreover, the matrix $C_{r(k)} \equiv I$ for all choices of the mode $r(k)$. The problem aims at designing the control input $u(k)$ to minimize the finite horizon cost function

$$\begin{aligned}J_{LQR}(K) = \sum_{k=1}^{K}\left(E_{\{r(j)\}_{j=k+1}^{K}}\left[x^T(k)Qx(k) + u^T(k)Ru(k)\right]\right)\\ + x^T(K+1)P(K+1)x(K+1),\end{aligned}$$

where the expectation at time $k$ is taken with respect to the future values of the Markov state realization, and $P(K+1)$, $Q$ and $R$ are all assumed to be positive definite. The controller at time $k$ has access to control inputs $\{u(j)\}_{j=0}^{k-1}$, state values $\{x(j)\}_{j=0}^{k}$ and the Markov state values $\{r(j)\}_{j=0}^{k}$. Finally, the system is said to be stabilizable if the infinite horizon cost function $J_\infty \overset{def}{=} \lim_{K\to\infty}\frac{J_{LQR}}{K}$ is finite.

The solution to this problem can readily be obtained through dynamic programming arguments. The optimal control is given by the following result.

**Theorem 4.4.** *Consider the LQR problem posed above for the system $\mathcal{S}_1$.*

1. *At time $k$, if $r(k) = i$, then the optimal control input is given by*

$$u(k) = -\left(R + B_i^T P_i(k+1) B_i\right)^{-1} B_i^T P_i(k+1) A_i x(k),$$

*where for $j = 1, 2, \cdots, m$,*

$$\begin{aligned} P_j(k) = \sum_{t=1}^{m} q_{tj} \Big( & Q + A_t^T P_t(k+1) A_t \\ & - A_t^T P_t(k+1) B_t \left(R + B_t^T P_t(k+1) B_t\right)^{-1} B_t^T P_t(k+1) A_t \Big), \end{aligned} \tag{4.3}$$

*and $P_j(K+1) = P(K+1), \forall j = 1, 2, \cdots, m$.*

2. *Assume that the Markov states reach a stationary probability distribution. A necessary and sufficient condition for stabilizability of the system is that there exist $m$ positive definite matrices $X_1$, $X_2$, $\cdots$, $X_m$ and $m^2$ matrices $K_{1,1}$, $K_{1,2}$, $\cdots$, $K_{1,m}$, $K_{2,1}$, $\cdots$, $K_{m,m}$ such that for all $j = 1, 2, \cdots, m$,*

$$X_j > \sum_{i=1}^{m} q_{ij} \left( (A_i^T + K_{i,j} B_i^T) X_i (A_i^T + K_{i,j} B_i^T)^T + Q + K_{ij} R K_{ij}^T \right).$$

3. *A necessary condition for stabilizability is that*

$$q_{i,i} \rho(A_i)^2 < 1, \qquad \forall i = 1, 2, \cdots, m,$$

*where $\rho(A_i)$ is the spectral radius of the matrix $A_i$ that governs the dynamics of unstabilizable modes of the process in the $i$-th mode.*

*Proof.* The proof follows by standard dynamic programming arguments. We begin by rewriting the cost function $J_{LQR}$ to identify terms in the cost that depend on $x(K)$ and $u(K)$:

$$J_{LQR}(K) = \sum_{k=1}^{K-1} \left( E_{\{r(j)\}_{j=k+1}^{K}} \left[ x^T(k) Q x(k) + u^T(k) R u(k) \right] \right) + T(K)$$

$$T(K) = E_{r(K)} \left[ x^T(K) Q x(K) + u^T(K) R u(K) \right] + x^T(K+1) P(K+1) x(K+1).$$

We rewrite $T(K)$ by explicitly conditioning it on the value of $r(K)$.

$$T(K) = \sum_{i=1}^{m} \pi_i(K) \left( x^T(K) Q x(K) + u^T(K) R u(K) + x^T(K+1) P_i(K+1) x(K+1) | r(K) = i \right),$$

where $P_i(K+1) = P(K+1), \forall i$. At the time of calculation of $u(K)$, the mode $r(K)$ is known. To choose the control input for any value of the mode, we complete the square of each of the terms in the summation. For the $i$-th term we obtain

$$\begin{aligned}
&\big(x^T(K)Qx(K) + u^T(K)Ru(K) + x^T(K+1)P_i(K+1)x(K+1)|r(K)=i\big)\\
&\quad = x^T(K)Qx(K) + u^T(K)Ru(K) + (A_ix(K)+B_iu(K))^T P_i(K+1)(A_ix(K)+B_iu(K))\\
&\quad = x^T(K)M_i(K)x(K) + (u(K) + S_i^{-1}(K)B_i^T P_i(K+1)A_ix(K))^T S_i(K)(u(K) + S_i^{-1}(K)B_i^T P_i(K+1)
\end{aligned}$$

where

$$\begin{aligned}
S_i(K) &= R + B_i^T P_i(K+1)B_i\\
M_i(K) &= Q + A_i^T P_i(K+1)A_i - A_i^T P_i(K+1)B_iS_i^{-1}(K)B_i^T P_i(K+1)A_i.
\end{aligned}$$

Thus, the optimal choice of $u(K)$ for the case $r(K) = i$ is

$$u(K) = -S_i^{-1}(K)B_i^T P_i(K+1)x(K).$$

With the optimal choice of $u(K)$ for all values of $i = 1, \cdots, m$, the term $T(K)$ reduces to

$$\begin{aligned}
T(K) &= \sum_{i=1}^m \pi_i(K)\left(x^T(K)M_i(K)x(K)|r(K)=i\right)\\
&= \sum_{i=1}^m \pi_i(K)\sum_{j=1}^m q_{ji}\left(x^T(K)M_i(K)x(K)|r(K)=i, r(K-1)=j\right)\\
&= \sum_{j=1}^m\sum_{i=1}^m \pi_i(K)q_{ji}\left(x^T(K)M_i(K)x(K)|r(K-1)=j\right)\\
&= \sum_{j=1}^m \left(x^T(K)(\sum_{i=1}^m \pi_i(K)q_{ji}M_i(K))x(K)|r(K-1)=j\right)\\
&= \sum_{j=1}^m \left(x^T(K)\pi_j(K-1)P_j(K)x(K)|r(K-1)=j\right)\\
&= E_{r(K-1)}\left[x^T(K)P_j(K)x(K)\right],
\end{aligned}$$

where

$$\pi_j(K-1)P_j(K) = \sum_{i=1}^m \pi_i(K)q_{ji}M_i(K).$$

Thus, the cost function $J_{LQR}$ can be rewritten as

$$\begin{aligned}
J_{LQR}(K) &= \sum_{k=1}^{K-2}\left(E_{\{r(j)\}_{j=k+1}^{K-1}}\left[x^T(k)Qx(k) + u^T(k)Ru(k)\right]\right) + T(K-1)\\
T(K-1) &= E_{r(K-1)}\left[x^T(K-1)Qx(K-1) + u^T(K-1)Ru(K-1) + x^T(K)P_i(K)x(K)\right].
\end{aligned}$$ ■

If we rewrite $T(K)$ by explicitly conditioning it on the value of $r(K-1)$,

$$\begin{aligned} T(K-1) = \sum_{i=1}^{m} \pi_i(K-1)\Big(& x^T(K-1)Qx(K-1) + u^T(K-1)Ru(K-1) \\ & + x^T(K)P_i(K)x(K)|r(K-1)=i\Big), \end{aligned}$$

we see that the problem of choosing $u(K-1)$ is formally identical to the problem that we solved above for choosing $u(K)$. Thus, the same argument can be repeated at any time step recursively. At a general time $k$, the control input $u(k)$ given $r(k)=i$ is given by

$$u(k) = -S_i^{-1}(k)B_i^T P_i(K+1)x(K),$$

where

$$\begin{aligned} S_i(k) &= R + B_i^T P_i(k+1)B_i \\ \pi_j(K-1)P_j(K) &= \sum_{i=1}^{m} \pi_i(K)q_{ji}M_i(K) \\ M_i(k) &= Q + A_i^T P_i(k+1)A_i - A_i^T P_i(k+1)B_i S_i^{-1}(k)B_i^T P_i(k+1)A_i, \end{aligned}$$

with boundary value $P_i(K+1) = P(K+1)\forall i$. This proves the first part of the theorem.

To prove the second and third parts, we need to study the stability of the terms $P_i(0)$ as the horizon $K \to \infty$. □

The sufficient condition for stabilizability can also be cast in alternate forms as linear matrix inequalities, that can be efficiently solved, as follows.

**Theorem 4.5.**

*Proof.* □

The above conditions reduce to simpler form for Bernoulli jump linear systems. For this case, the LQR and stabilizability problems can be solved to yield the following result.

**Theorem 4.6.** *Consider system $\mathcal{S}_1$ with the additional assumption that the Markov transition probability matrix is such that for all states $i$ and $j$, $q_{ij} = q_i$ (in other words, the states are chosen independently and identically distributed from one time step to the next). Consider the LQR problem posed above for the system $\mathcal{S}_1$.*

1. *At time $k$, if $r(k)=i$, then the optimal control input is given by*

$$u(k) = -\left(R + B_i^T P(k+1)B_i\right)^{-1} B_i^T P(k+1)A_i x(k),$$

*where*

$$P(k) = \sum_{t=1}^{m} q_t \Big( Q + A_t^T P(k+1) A_t \\ - A_t^T P(k+1) B_t \left( R + B_t^T P(k+1) B_t \right)^{-1} B_t^T P(k+1) A_t \Big).$$

2. *Assume that the Markov states reach a stationary probability distribution. A sufficient condition for stabilizability of the system is that there exists a positive definite matrix $X$, and $m$ matrices $K_1$, $K_2$, $\cdots$, $K_m$ such that*

$$X > \sum_{i=1}^{m} q_i \left( (A_i^T + K_i B_i^T) X (A_i^T + K_i B_i^T)^T + Q + K_i R K_i^T \right).$$

3. *A necessary condition for stabilizability is that*

$$q_i \rho(A_i)^2 < 1, \qquad \forall i = 1, 2, \cdots, m,$$

*where $\rho(A_i)$ is the spectral radius of the matrix $A_i$ that governs the dynamics of unstabilizable modes of the process in the $i$-th mode.*

*Proof.* The result follows readily from the LQR solution of Markovian jump linear systems. Specifically, if we substitute $q_{tj} = q_j \forall t$ in (4.3), we see that all matrices $P_j(k)$ are identical for $j = 1, \cdots, m$. If we denote this value by $P(k)$, we obtain the desired form of the LQR control law. Similarly the stability conditions in the theorem also follow from those for Markovian jump linear systems in Theorem 4.4. □

### Optimal Minimum Mean Squared Error Estimator

The minimum mean squared error estimate problem for the system $\mathcal{S}_1$ is posed by assuming that the control $u_{r(k)}$ is identically zero. The objective is to identify at every time step $k$, an estimate $\hat{x}(k+1)$ of the state $x(k+1)$ that minimizes the mean squared error covariance

$$\Pi(k+1) = E_{\{w(j)\},\{v(j)\},x(0)} \left[ (x(k+1) - \hat{x}(k+1))(x(k+1) - \hat{x}(k+1))^T \right],$$

where the expectation is taken with respect to the process and measurement noises, and the initial state value (but not the Markov state realization). The estimator at time $k$ has access to observations $\{y(j)\}_{j=0}^{k}$ and the Markov state values $\{r(j)\}_{j=0}^{k}$. Moreover, the error covariance is said to be stable if the expected steady state error covariance $\lim_{k\to\infty} E_{\{r(j)\}_{j=0}^{k-1}}[\Pi(k)]$ is bounded, where the expectation is taken with respect to the Markov process.

Since the estimator has access to the Markov state values till time $k$, the optimal estimate can be calculated through a time-varying Kalman filter.

Thus, if at time $k$, $r_k = i$, the estimate evolves as

$$\hat{x}(k+1) = A_i\hat{x}(k) + K(k)\left(y(k) - C_i\hat{x}(k)\right),$$

where

$$K(k) = A_i\Pi(k)C_i^T\left(C_i\Pi(k)C_i^T + R_v\right)^{-1}$$
$$\Pi(k+1) = A_i\Pi(k)A_i^T + R_w - A_i\Pi(k)C_i^T\left(C_i\Pi(k)C_i^T + R_v\right)^{-1}C_i\Pi(k)A_i^T.$$

The error covariance $\Pi(k)$ is available through the above calculations. However, calculating $E_{\{r(j)\}_{j=0}^{k-1}}[\Pi(k)]$ seems to be intractable. Instead, the normal approach is to consider an upper bound to this quantity[1] that will also help in obtaining sufficient conditions for the error covariance to be stable.

The intuition behind obtaining the upper bound is simple. The optimal estimator presented above optimally utilizes the information about the Markov states till time $k$. Consider an alternate estimator that at every time step $k$, averages over the values of the Markov states $r_0, \cdots, r_{k-1}$. Such an estimator is sub-optimal and the error covariance for this estimator forms an upper bound for $E_{\{r(j)\}_{j=0}^{k-1}}[\Pi(k)]$. A more formal derivation for the upper bound is presented below.

**Theorem 4.7.** *The term $E_{\{r(j)\}_{j=0}^{k-1}}[\Pi(k)]$ obtained from the optimal estimator is upper bounded by $M(k) = \sum_{j=1}^{m} M_j(k)$ where*

$$\begin{aligned} M_j(k) = \sum_{t=1}^{m} q_{tj}\Big(&R_w + A_tM_t(k-1)A_t^T \\ &- A_tM_t(k-1)C_t^T\left(R_v + C_tM_t(k-1)C_t^T\right)^{-1}C_tM_t(k-1)A_t^T\Big), \end{aligned}$$

*with $M_j(0) = \Pi(0)$ $\forall j = 1, 2, \cdots, m$. Moreover, assume that the Markov states reach a stationary probability distribution. A sufficient condition for stabilizability of the system is that there exist $m$ positive definite matrices $X_1, X_2, \cdots, X_m$ and $m^2$ matrices $K_{1,1}, K_{1,2}, \cdots, K_{1,m}, K_{2,1}, \cdots, K_{m,m}$ such that for all $j = 1, 2, \cdots, m$,*

$$X_j > \sum_{i=1}^{m} q_{ij}\left((A_i + K_{i,j}C_i)X_i(A_i + K_{i,j}C_i)^T + R_w + K_{ij}R_vK_{ij}^T\right).$$

*Finally, a necessary condition for stabilizability is that*

$$q_{i,i}\rho(A_i)^2 < 1, \qquad \forall i = 1, 2, \cdots, m,$$

*where $\rho(A_i)$ is the spectral radius of the matrix $A_i$ that governs the dynamics of unobservable modes of the process in the $i$-th mode.*

*Proof.* We begin by defining

$$M_j(k) = \pi_j(k-1)E\left[\Pi(k)|r(k-1) = j\right],$$

[1] We say that $A$ is upperbounded by $B$ if $B - A$ is positive semi-definite.

so that

$$E\left[\Pi(k)\right] = \sum_{i=1}^{m} M_j(k).$$

Now we can bound each term $M_j(k)$ as follows.

$$\begin{aligned} M_j(k+1) &= \pi_j(k) \sum_{i=1}^{m} E\left[\Pi(k+1) | r(k)=j, r(k-1)=i\right] \text{Prob}(r(k-1)=i|r(k)=j) \\ &= \sum_{i=1}^{m} E\big[A_j\Pi(k)A_j^T + R_w - A_j\Pi(k)C_j^T(C_j\Pi(k)C_j^T + R_v)^{-1}C_j\Pi(k)A_j^T | r(k-1)=i\big]\, q_{ij}\pi_i(k - \end{aligned}$$

since given $r(k-1)$, $\Pi(k)$ and $r(k)$ are independent. Further, note that the Riccati operator

$$f_j(M) = A_jMA_j^T + R_w - A_jMC_j^T(C_jMC_j^T + R_v)^{-1}C_jMA_j^T$$

is both concave and increasing. Since it is concave, Jensen's inequality yields

$$M_j(k{+}1) \leq \sum_{i=1}^{m} \big(A_jE[\Pi(k)|r(k-1)=i]A_j^T + R_w - A_jE[\Pi(k)|r(k-1)=i]C_j^T(C_jE[\Pi(k)|r(k-1)=i]C$$

Now from the definition of $M_i(k-1)$ and the fact that $f_j(.)$ is an increasing operator, we obtain the required bound.

For the stability proof, □

The special case of a Bernoulli jump linear systems can be obtained from the above result by substituting $q_{ij} = q_j \forall i$. We state the result below.

**Theorem 4.8.** *Consider the estimation problem posed above for the system $\mathcal{S}_1$ with the additional assumption that the Markov transition probability matrix is such that for all states $i$ and $j$, $q_{ij} = q_i$ (in other words, the states are chosen independently and identically distributed from one time step to the next). The term $E_{\{r(j)\}_{j=0}^{k-1}}[\Pi(k)]$ obtained from the optimal estimator is upper bounded by $M(k)$ where*

$$\begin{aligned} M(k) = \sum_{t=1}^{m} q_t \Big( & R_w + A_tM(k-1)A_t^T \\ & - A_tM(k-1)C_t^T\left(R_v + C_tM(k-1)C_t^T\right)^{-1}C_tM(k-1)A_t^T\Big), \end{aligned}$$

*with $M(0) = \Pi(0)$. Further, a sufficient condition for stabilizability of the system is that there exists a positive definite matrix $X$, and $m$ matrices $K_1$, $K_2$, $\cdots$, $K_m$ such that*

$$X > \sum_{i=1}^{m} q_i\left((A_i + K_iC_i)X(A_i + K_iC_i)^T + R_w + K_iR_vK_i^T\right).$$

*Finally, a necessary condition for stabilizability is that*

$$q_i \rho(A_i)^2 < 1, \qquad \forall i = 1, 2, \cdots, m,$$

*where $\rho(A_i)$ is the spectral radius of the matrix $A_i$ that governs the dynamics of unobservable modes of the process in the $i$-th mode.*

### Linear Quadratic Gaussian Control

Given the optimal linear quadratic regulator and minimum mean squared error estimator, the solution of the linear quadratic Gaussian control problem can be solved by utilizing a separation principle. The Linear Quadratic Gaussian (LQG) problem for the system $\mathcal{S}_1$ aims at designing the control input $u(k)$ to minimize the finite horizon cost function

$$J_{LQG} = E\left[\sum_{k=1}^{K}\left(x^T(k)Qx(k) + u^T(k)Ru(k)\right) + x^T(K+1)P(K+1)x(K+1)\right],$$

where the expectation at time $k$ is taken with respect to the future values of the Markov state realization, the measurement and process noises, and the initial state. Further, the matrices $P(K+1)$, $Q$ and $R$ are all assumed to be positive definite. The controller at time $k$ has access to control inputs $\{u(j)\}_{j=0}^{k-1}$, measurements $\{y(j)\}_{j=0}^{k}$ and the Markov state values $\{r(j)\}_{j=0}^{k}$. The system is said to be stabilizable if the infinite horizon cost function $J_\infty \overset{def}{=} \lim_{K\to\infty} \frac{J_{LQG}}{K}$ is finite.

The solution to this problem is provided by Theorems 4.4 and 4.7 because of the following separation principle.

**Theorem 4.9.** *Consider the LQG problem for the system $\mathcal{S}_1$. At time $k$, if $r(k) = i$, then the optimal control input is given by*

$$u(k) = -\left(R + B_i^T P_i(k+1)B_i\right)^{-1} B_i^T P_i(k+1)A_i\hat{x}(k),$$

*where for $P_i(k)$ is calculated as in Theorem 4.4 and $\hat{x}(k)$ is calculated using a time-varying Kalman filter.*

*Proof.* □

Given this separation principle, the stabilizability conditions provided in Theorems 4.4 and 4.7 can then be combined to yield the stabilizability conditions for the LQG case as well. Finally, we note that a similar separation principle also holds for Bernoulli jump linear systems. Thus, the LQG problem can be solved for this case as well.

## 4.4 $H_\infty$ Control

Include?

## 4.5 Further Resources

# Chapter 5
## Rate-Limited Estimation and Control

Some references we should cover: RMM

- P. Minero, M. Franceschetti, S. Dey and G. N. Nair, “Data rate theorem for Stabilization over time-varying fedback channels”, IEEE TAC, 54(2):243–255, 2008.

*Vijay Comment: The above reference can be covered both in analog erasure link chapter and this chapter. My suggestion is to cover it in the chapter that comes later in the book. Also Nuno Martins claims that a similar result exists in one of his papers. We should check that, and if true, cite both.*

In this chapter, we consider the class of networked control systems in which the communication channel can be described by a digital noiseless channel. Such a channel imposes a limit on the number of bits that can be transmitted across it as a function of time; however, the transmission is perfect. As we shall see, there is a minimum bit rate required for the existence of encoders and decoders so that the plant can be stabilized across such a channel. In that sense, this problem is an analog of the source coding problem in information theory. However, the results from information theory are not directly applicable to the control scenario because of their reliance on large delays for the block codes to work. Nevertheless, concepts and insights from information theory will be used in the following discussion.

The chapter is organized as follows. We begin by describing the channel model in the next section.

### 5.1 Channel Model

By a digital noiseless channel, we will mean the following model. Consider a finite alphabet $S$ of cardinality $M \geq 1$. At every time $k$, the channel accepts as input one symbol $s(k) \in S$. With a delay of $d$ time steps, the channel outputs the symbol $r(k+d) = s(k)$. We will nominally consider the delay to be 0; however, we mention how the results can be extended to any finite value of the delay. Since the encoder for such a channel maps a continuous variable (e.g., the state value or the measurement) to a discrete variable (the input of the channel), it is often referred to as a quantizer.

An alternate viewpoint is to consider a channel that operates with a binary alphabet; however, at every time step, it can support a data rate

$R = \log_2 M$ bits per sample. From this perspective, the channel model is that of a bit rate limited channel. We can also distinguish between channels that support a rate $R$ at every time step, and those that support an average rate $R = \lim_{N\to\infty} \sum_{k=0}^{N} \frac{R(k)}{N}$, where $R(k)$ refers to the instantaneous rate (or number of bits supported by the channel) at time $k$.

## 5.2 Single Block Design

Consider a process with state $x(k) \in \mathbf{R}^n$ that evolves as

$$x(k+1) = Ax(k) + Bu(k) + w(k),$$

where $w(k)$ is process noise modeled as white and in a bounded region $\mathcal{W}$. The initial state $x(0)$ is also assumed to lie in a bounded region $\mathcal{X}$. For simplicity, we assume that the sensor can observe the state $x(k)$. The sensor transmits data to a controller across a digital noiseless channel with rate $2^M$ bits at every time step. The single block design problem refers to a situation in which the sensor quantizes the state space using $M$ bits and transmits them to the controller. The controller aims to calculate a control input $u(k)$ to minimize the quadratic cost

$$J_T = \sum_{k=0}^{T} E[x^T(k)Qx(k) + u^T(k)Ru(k)] + x^T(T+1)P(T+1)x(T+1).$$

If the infinite horizon cost $\lim_{T\to\infty} \frac{J_T}{T}$ is bounded, we say that the process has been stabilized. Notice that in the single block design paradigm for this channel implies that the quantizer is given and the system designer specifies the decoder/controller. However, the quantizer can be of many different types as long as it satisfies the rate constraint. Some popular choices for quantizers are uniform or logarithmic with given range and step sizes.

The presence of a digital noiseless channel significantly complicates the analysis and design of control loops even for the LQG problem. For one, quantization is inherently a non-linear process and thus converts the problem to a non-linear control problem. Thus, there are only a limited number of results about optimal controller design. Another reason is that the quantization error introduced at any time step impacts the state value, and hence the quantization error, at all future time steps. This relation can become very complicated for arbitrary quantizers, possibly even leading to the control having a dual effect. For the cases when process noise is present, the possibility of state value becoming large enough to fall outside the quantizer range (termed quantizer overflow) is an additional complication.

The chief approach in single block design is to make a white noise approximation for the quantization error. Under this approximation, the possibility of quantizer overflow is ignored and each of the $n$ elements in the state vector are assumed to be quantized independently using a uniform quantizer with step size $\delta$, where $\delta$ is such that the total number of bits transmitted by the quantizer is $M$. Moreover, the quantization error $q(k)$ is assumed to be white and independent of $x(k)$. Since the quantization error for a uniform quantizer with step size $\delta$ has mean 0 and variance $\delta^2/12$, the effect of the above assumptions is to replace the quantizer with a sensor of the form

$$y(k) = x(k) + v(k),$$

where $v(k)$ is sensor noise modeled as bounded and white with mean zero and variance $\delta^2/12$. The controller design problem thus reduces to the design of a stabilizing controller for a linear system, which can be readily solved. Since the noises are not Gaussian, the performance optimal controller is harder to design.

Some initial results when the assumption of quantization error being either white or independent of the state value is not made are provided for uniform and logarithmic quantizers in [**?**]. The chief technical tool is the following *high rate approximation* result from source coding theory [**?**].

**Theorem 5.1.** *Given a scalar quantizer with mean squared error based distortion measure $d(x,y) = \|x-y\|^2$, the expected distortion of the random variable $X$ that is being quantized can be bounded as*

$$\bar{d} \geq d_L = \frac{1}{12N^2} E[\lambda(X)^{-2}],$$

*where $\lambda(X)$ is the asymptotic quantizer density normalized to unit integral, while $N$ refers to the number of quantization levels. Further, the lower bound becomes tighter as the rate of the quantizer becomes high.*

Thus, consider the quadratic cost $J_T$ for a scalar plant

$$x(k+1) = ax(k) + u(k) + w(k),$$

where the noise $w(k)$ and the initial state $x(0)$ are both bounded. Assume that there is no quantizer overflow, and that the control input is given by $u(k) = f\hat{x}(k)$, where $\hat{x}(k)$ is the estimate of the state at the decoder. Then, for a midpoint based uniform quantizer, $d_L = \frac{\delta^2}{12}$ where $\delta$ is the quantizer step size. Moreover, as Marco and Neuhoff [**?**] proved, for a high rate uniform quantizer,[1] $E[x(k)\delta(k)] \ll E[\delta^2(k)]$ and can thus be approximated by zero. Thus, at high rates the cost $J_T$ evaluates to

$$J_T = (Q+Rf^2)E[x^2(0)]\sum_{k=0}^{T}(a+f)^{2k} + Rf^2(T+1)\frac{\delta^2}{12} + \frac{Q+Rf^2}{1-(a+f)^2}\left(\frac{\delta^2 f^2}{12} + \Sigma_w^2\right)\sum_{k=0}^{T}\left(1-(a+f)^{2k}\right).$$

The optimal controller can now be evaluated numerically. On the other hand, if a logarithmic quantizer with ratio $g$ operating over the union of

[1] There are some additional technical conditions required, which hold in this case.

regions $[-a, \epsilon]$ and $[\epsilon, a]$ is used, the distortion can be evaluated to be

$$d_L = \frac{(\ln g)^2}{12} E[x^2(k)].$$

Using the Cauchy-Schwarz inequality

$$-\sqrt{E[\delta^2(k)]E[x^2(k)]} \leq E[\delta(k)x(k)] \leq \sqrt{E[\delta^2(k)]E[x^2(k)]},$$

we can then obtain that

$$h_1 E[x^2(0)] \frac{1-g_1^{T+1}}{1-g_1} + \frac{h_1 \sigma^2 (T - 1 + g_1^{T+1})}{1-g_1} \leq J_T \leq h_2 E[x^2(0)] \frac{1-g_2^{T+1}}{1-g_2} + \frac{h_2 \sigma^2 (T - 1 + g_2^{T+1})}{1-g_2},$$

where

$$\begin{aligned} g_1 &= (a+f)^2 + cf^2 - 2 \mid f(a+f) \mid \sqrt{c} \\ g_2 &= (a+f)^2 + cf^2 + 2 \mid f(a+f) \mid \sqrt{c} \\ h_1 &= Q + Rf^2 + Rcf^2 - 2Rf^2\sqrt{c} \\ h_2 &= Q + Rf^2 + Rcf^2 + 2Rf^2\sqrt{c} \end{aligned}$$

and $c = \frac{(\ln g)^2}{12}$.

For a quantizer with large enough rate, either of the above approaches yield reasonably accurate results. However, analytically, the problem is largely unsolved since the framework with the above approximations fail to capture some crucial features of the solution. For one, the white noise approximation implies that the system can be stabilized by a suitable control law with any non-zero rate supported by the digital noiseless channel (provided that the pair $(A, B)$ is stabilizable). However, as we shall see in the next section, the data rate theorem implies that there is a minimum data rate that needs to be supported by the channel, otherwise the system cannot be stabilized even in the two block design paradigm. Moreover, the assumptions mentioned in this section fail to predict the chaotic nature of the state space trajectory that was identified by Delchamps.

**Vijay** Expand on this point. If somebody else has read the paper and better understands it, please fill in a few lines here.

## 5.3 Two Block Design

The two block design paradigm involves designing both an encoder at the input of the channel and a decoder at the output of the channel. As we shall see, for the digital noiseless channel, encoders and decoders that achieve stability with the minimum possible bit rate have been identified for a variety of stability notions and conditions on the encoder structure. However, designs that minimize a performance cost are largely unknown.

We begin by considering the plant structure with state $x(k) \in \mathbf{R}^n$ that evolves as

$$x(k+1) = Ax(k) + Bu(k) + w(k), \tag{5.1}$$

where $u(k) \in \mathbf{R}^m$ is the control input. The state is observed by a sensor that generates measurements $y(k) \in \mathbf{R}^p$ of the form

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

For different notions of stability, we will make different assumptions on the noises $w(k)$ and $v(k)$, and the initial state $x(0)$. We assume that the pair $(A, B)$ is controllable and the pair $(A, C)$ is observable.

The encoder at the input of the channel transmits a symbol $s(k)$ from the alphabet $S$ (equivalently, $M$ bits) at every time step. The message that is transmitted is a function of past transmissions and all measurements till time $k$, i.e.,

$$s(k) = \gamma(k, y(0), y(1), \cdots, y(k), s(0), s(1), \cdots, s(k-1)).$$

The channel transmits the symbol $s(k)$ without distortion, but with a constant delay of $d$ time steps. The decoder generates a control input of the form

$$u(k) = \delta(k, s(0), s(1), \cdots, s(k-d)).$$

We begin by considering stability in the sense of constraining the state value to lie within a bounded set. To this end, assume that the noises $w(k)$ and $v(k)$ are deterministic but unknown sequences constrained to lie in bounded sets $\mathcal{W}$ and $\mathcal{V}$ respectively. Moreover assume that the initial condition $x(0)$ lies in the bounded set $\mathcal{X}_0$. Then, we consider the system to be stable if the worst case cost

$$J = \lim_{k\to\infty} \sup\{\|x(k)\| : x(0) \in \mathcal{X}_0, w(j) \in \mathcal{W}, v(j) \in \mathcal{V}, j = 0, 1, \cdots\}$$

is bounded.

### Date Rate Theorem

The basic result in stability across digital noiseless channels is called the data rate theorem and is stated in terms of the intrinsic entropy of a system. The intrinsic entropy of a system is a measure of instability of a system and for the linear process in equation (5.1) is defined by the relation $H = \sum_i \log_2(\max(|\lambda_i(A)|, 1))$ where $\lambda_i(A)$ is the $i$-th eigenvalue of the matrix $A$. Since any mode of the process whose evolution is governed by an eigenvalue with magnitude less than 1 is stable and decays to zero even without any control input, for stabilization purpose, without loss of generality we can consider $A$ to have all eigenvalues with magnitude strictly larger than 1.

**Theorem 5.2** (Date Rate Theorem:)**.** *Consider the two block design formulation with a causal encoder and decoder structure as defined above with* *the channel supporting a rate $R$.*

1. *If $R \leq H$ and the process noise has non-zero support, then $J \to \infty$ with any encoder and decoder design.*

2. *If $R > H$ then*
$$J > \frac{\beta^{-1/n}\lambda(\mathcal{W})^{1/n}}{1 - 2^{-(R-H)/n}},$$
*where $\beta$ is the volume of an $n$-dimensional sphere with unit radius, and $\lambda(\mathcal{W})$ is the measure of $\mathcal{W}$.*

The proof of this theorem relies on considering the rate of increase in the volume of the set that the norm of the state value $x(k)$ can be in. The volume increases at every step because of the unstable eigenvalues, and decreases because of the information passed by the encoder. Note that the control value simply shifts this set, and cannot alter the volume since all previous control values are known to the controller. By balancing the rate of increase and decrease, the two conditions in the data rate theorem are obtained. To focus on the basic idea of the proof, we prove the data rate theorem for the special case when the process state is a scalar ($n = 1$). For this special case, the result implies:

1. If $R \leq \log_2(A)$ and the process noise has non-zero support, then $J \to \infty$ with any encoder and decoder design.

2. If $R > \log_2(A)$ then
$$J > \frac{0.5\lambda(\mathcal{W})}{1 - 2^{-(R-H)}},$$
for any encoder and decoder design, where $\lambda(\mathcal{W})$ is the measure of $\mathcal{W}$.

*Proof.* Define by $\lambda(x(t))$ the length of the possibly disconnected region defined by the set of values that the state value can achieve at time $t$ for various values of control inputs, $x(0)$ and the noise till time $t$. Also for given values of the signals transmitted by the encoder, define the region $\lambda(x(t) : \{s(j)\}_{j=0}^{t-1} = \{c(j)\}_{j=0}^{t-1})$ similarly. Finally define

$$m(t) = \max_{\{c(j)\}_{j=0}^{t-1} \in S} \lambda(x(t) : \{s(j)\}_{j=0}^{t-1} = \{c(j)\}_{j=0}^{t-1}).$$

We wish to study the evolution of $m(t)$. By definition,

$$\begin{aligned} m(t+1) &= \max_{\{c(j)\}_{j=0}^{t} \in S} \lambda(x(t+1) : \{s(j)\}_{j=0}^{t} = \{c(j)\}_{j=0}^{t}) \\ &= \max_{\{c(j)\}_{j=0}^{t} \in S} \lambda(Ax(t) + B\delta(t, \{c(j)\}_{j=0}^{t-d}) + w(t) : \{s(j)\}_{j=0}^{t} = \{c(j)\}_{j=0}^{t}). \end{aligned}$$

Now for given symbols $\{c(j)\}_{j=0}^{t-d}$, the control value is a constant and hence cannot affect the measure of the set. Moreover, the Brunn-Minkowski inequality implies that $\lambda^{1/n}(a+b) \geq \lambda^{1/n}(a) + \lambda^{1/n}(b)$, for any sets $a$ and $b$ in

$n$-dimensions. Utilizing these two facts, we obtain

$$m(t+1) \geq A \max_{\{c(j)\}_{j=0}^{t} \in S} \lambda(x(t) : \{s(j)\}_{j=0}^{t} = \{c(j)\}_{j=0}^{t}) + \lambda(w(t)). \quad (5.2)$$

Now, notice that

$$\{x(t) : \{s(j)\}_{j=0}^{t-1} = \{c(j)\}_{j=0}^{t-1}\} = \bigcup_{\text{all possible values of } c(t)} \{x(t) : \{s(j)\}_{j=0}^{t} = \{c(j)\}_{j=0}^{t}\}.$$

Thus, Brunn-Minkowski inequality yields

$$\lambda(x(t) : \{s(j)\}_{j=0}^{t-1} = \{c(j)\}_{j=0}^{t-1}) \leq \sum_{\text{all possible values of } c(t)} \lambda(x(t) : \{s(j)\}_{j=0}^{t} = \{c(j)\}_{j=0}^{t})$$

$$M \max_{c(t) \in S} \lambda(x(t) : \{s(j)\}_{j=0}^{t} = \{c(j)\}_{j=0}^{t}),$$

where $M$ is the number of symbols in the alphabet $S$. Using this in (5.2) yields

$$\begin{aligned} m(t+1) &\geq A \max_{\{c(j)\}_{j=0}^{t-1} \in S} \frac{1}{M} \lambda(x(t) : \{s(j)\}_{j=0}^{t-1} = \{c(j)\}_{j=0}^{t-1}) + \lambda(w(t)) \\ &= \frac{A}{M} m(t) + \lambda(w(t)) \\ &= 2^{-(R-H)} m(t) + \lambda(w(t), \end{aligned}$$

since $\frac{A}{M} = \frac{2^H}{2^R}$. If $R \leq H$, then as $t \to \infty$, $m(t)$ grows without bound and $J \to \infty$. If $R > H$, then we can solve for $m(t)$ explicitly and achieve the bound stated in the theorem as $t \to \infty$. This, the data rate theorem is proven for the scalar case. □

The proof for the $n$-dimensional state space is along similar lines by considering the evolution of the volume of $|x(t)|$. The technical changes required are:

- to relate the volume of the set to $\sup |x(t)|$ by using the expression
  $$\lambda(\mathcal{T}) \leq \beta(\sup |\tau| : \tau \in \mathcal{T})^{1/n},$$
  for any set $\mathcal{T}$, where $\beta$ is the volume of the $n$-dimensional sphere with unit radius.

- to define $m(t)$ as the $1/n$-th root of volume so that Brunn-Minkowski inequality is applicable.

- to use the relation $\lambda(Ax) = \det(A)\lambda(x)$.

The proof of the theorem can be easily modified to consider the case when the rate $R$ is time-varying. By considering the evolution of $m(t)$ in terms of $m(0)$, we see that the data rate theorem holds if we define $R$ to be the long term average data rate.

### Tightness of Bounds

There are two questions that one can consider regarding the tightness of the bounds:

1. Is it possible to achieve stability with a data rate $R = H + \epsilon$ for any non-zero $\epsilon$?

2. Is it possible to achieve the lower bound on the state value when $R > H$?

The first question can be answered in affirmative. We construct an encoder and decoder for the scalar case that achieves stability with rate $H + \epsilon$ and indicate how it can be extended to more general cases. Consider at time $k$, the state $x(k)$ to be in a region with support $l(k)$. The encoder uniformly quantizes the region using $M$ bits and transmits the symbol related to the midpoint of the quantization cell containing the state value. The decoder knows the region being quantized since there is no stochasticity in the system. Thus, it knows exactly the midpoint of the quantization cell as transmitted by the encoder. It applies the control that translates the midpoint to the origin. Using this encoder decoder pair, the length $l(k)$ evolves as follows. It increases by a factor of $A$ due to the state dynamics, and decreases by a factor of $2^M$ due to the quantizer. Thus,

$$l(k+1) = \frac{1}{M}(Al(k) + l_w(k)),$$

where $l_w(k)$ is the support of the noise $w(k)$. Thus, the length is bounded as $k \to \infty$ (in other words stability is achieved) if $\frac{A}{M} < 1$ or $R = H + \epsilon$ for any non-zero $\epsilon$. For a vector plant, a similar encoder-decoder pair can be used for each individual mode as identified by a Jordan decomposition. By allotting bit rates suitably for all unstable modes, stability can be achieved for any rate $R > H$.

Regarding the second question, we notice that the lower bound on the norm of the state in case of $R > H$ is independent of the delay $d$. Thus, it can be expected that the bound is quite loose in general. While the presence of a finite delay cannot affect the stability condition, it does affect the performance in terms of the achievable norm of the state. One can modify the above proof by considering the evolution of $m(t)$ in terms of $m(t-d)$ to more accurately capture the effects of the delay. However, the effect of the rate $R$ and the delay $d$ do not separate out in a simple manner.

Even if $d = 0$, in general the bound is not tight for vector plants. For scalar plants, the encoder-decoder proposed above will achieve the bound with equality.

### Other Notions of Stability

Now we briefly discuss how to consider alternate notions of stability. If the noises $w(k)$ and $v(k)$ are random variables, then the state $x(k)$ evolves stochastically. In that case, we might be interested in boundedness of a particular moment of the state. The most popular notion is that of mean square stability, i.e., we define the system to be stable if $E[x(k)x^T(k)]$ is bounded as $k \to \infty$. This stability notion can be analyzed using very similar tools as employed in the data rate theorem. Instead of volume of the set in which the state can lie, we consider the evolution of the entropy power of the state. By using the entropy power inequality instead of the Brunn-Minkowski inequality, we can follow the same proof as that of the data rate theorem. Rather surprisingly, the condition on the minimum bit rate required for stability turns out to be identical to the deterministic case considered earlier.

However, unlike the deterministic case, a finite memory encoder-decoder pair is no longer sufficient to provide moment stability if the noise has infinite support. This result is related to the area over which the quantizer needs to operate. If the noise has a compact support, then given the region in which $x(k)$ can lie, the region that needs to be quantized for $x(k+1)$ is bounded. The encoder can thus vary its range at every time step and achieve stability. However, if the noise has unbounded support, then there is always a finite chance that the state at time $k+1$ falls outside the range of the quantizer. Such quantizer overflow leads to controller saturation. If the plant is unstable, the difference between where the state is, and the maximum value that can be handled by the controller exacerbates and quantizer overflow happens with increasing probability, ultimately leading to instability.

Stability with noises that have infinite support requires an encoder that adapts its range to allow the control signal to catch up. Moreover, the adaptation parameter can depend on the entire history of actions and have infinite set of values. A typical example of such a quantizer is the zoom in / zoom out quantizer.

Introduce description of zoom in and out quantizer **vijay**

.

If there is no noise in the system, one can also aim for asymptotic stability. The first result in this direction is the insufficiency of a finite memory encoder / decoder pair to achieve asymptotic stability. This is because given any finite memory encoder / decoder, and a finite data rate, at any time $t$, the controller can only distinguish between finitely many state values. In other words, at any time $t$, there are a countable number of values for the initial state $x(0)$, that can be mapped to the origin. For any other value of the initial state, let, if possible, the system achieve asymptotic stability. This implies that if there exists a time $T$, such that for all $t > T$, the state value starting from this initial state satisfies $|x(t)| < \epsilon$, then there must exist some time $\tau > T$ such that $u(\tau)$ is nonzero. This is simply because the state value $x(t)$ is non-zero and hence needs to be driven to the origin by a suitable control input. Now, of the (countably many) distinct possible values possible for the control input at time $\tau$, let $m = \min |u(\tau)|$. Moreover, choose $\epsilon = \frac{m}{1+|a|}$, where for simplicity we have choosen a scalar system with system parameter $a$. Then we can obtain a contradiction by noting that

$$m \leq |u(\tau)| = |x(\tau+1) - ax(\tau)| \leq |x(\tau+1)| + |ax(\tau)| \leq (1+|a|)\epsilon < m.$$

Thus, for any finite memory of the encoder and decoder, there are uncountably many initial values of the state such that the system trajectories starting from them do not achieve asymptotic stability. There are two chief research directions that have emerged in light of this negative result:

1. Loosening the constraint of asymptotic stability to practical stability: In practical stability, the system satisfies the constraint $|x(t)| < \epsilon$ for any given $\epsilon$ for times in the range $T_1 < t < T_2$.

2. Considering encoders / decoders with infinite memory: The zoom in and out quantizers discussed above can achieve asymptotic stability by varying the region of state space that is quantized. As the state value moves closer to the origin, the range of the quantizer decreases. The achievement of increasingly finer quantization levels leads to asymptotic stability.

**vijay** Should we discuss either of these directions in more detail?

.

## 5.4 Extensions and Open Questions

The above discussion provides a sketch of the type of problems that have been analyzed and the results that are available for estimation and control across a digital noiseless channel. Since this is a research area that continues to see intense activity, there are a number of aligned problems that have also been looked at. We provide a discussion on some of these problems and outline a few open research questions.

*Performance.* Most of the material presented this far focussed on various notions of stability. Results on the design of control inputs to minimize a cost metric are more limited. As an example, consider the case when the noises are stochastic and Gaussian. One can consider the LQG problem for this case. The problem is difficult because of the non-linearity introduced by the quantizer. However, it can be proven that for the class of encoders that subtract the effect of previous control inputs (thus transmitting an innovation-like quantity), the certainty equivalence principle holds. Thus, for this class of encoders, the optimal control law is linear and its form is obtained by assuming that the controller has access to state information. However, instead of the state value, an estimate of the state value is used in the law to obtain the control input value. Moreover, there is no loss of optimality by restricting attention to this class of encoders.

However, a complete separation principle does not exist because of the non-linearity introduced by the quantizer. In other words, the estimate at the controller needs to be calculated for a distortion metric that depends on the input matrix. Thus, the estimator depends on the control value being calculated and is hard to characterize analytically. In fact, the LQG problem, and identifying the optimal controller for general cost functions, is still largely open. In fact, even the problems of identifying the optimal encoder for a given controller and identifying the optimal controller for a given encoder from a performance perspective do not yet have general solutions.

*Noisy Digital Channels.* If the channel is not noiseless, the problem becomes much harder. The easiest extension to consider is when the bits are erased by the channel with a certain probability at every time step. Stability conditions for such channels are obtained by extending the results from this chapter and that of analog erasure channels. Consider the case when the sensor data for a scalar process with process matrix $A$ is transmitted to the controller over a channel such that the channel supports a rate of $R$ bits at every time step, and the data packet is erased with a probability $p$ in an independent and identically distributed manner at every time step. Then, a necessary and sufficient condition for existence of causal encoders and decoders that achieve second moment stability for the plant state is that $pA^2 + (1-p)\frac{A^2}{2^{2R}} < 1$. Note that in the limit $R \to \infty$, the condition reduces to $pA^2 < 1$ that is the stability condition for two block design with analog erasure channels. Similarly in the limit $p = 0$, we regain the condition for digital noiseless channel. The extension of the condition for vector plants is obtained using Jordan decomposition of the system matrix and considering each unstable mode separately.

If the noise in the channel can yield bit errors (rather than erasures), then a binary symmetric channel is more accurate. However, only very limited results are available even for stability over such channels.

*Finite $L_p$ Gain / Nonlinear systems.*

Read paper and include discussion **Vijay**

*Distributed Control.* Since performance optimal distributed controllers are not available for arbitrary connection topologies even for the case of no limitations in terms of communication channels, it is not surprising that the problem is open if various components transmit data over digital noiseless channels. However, the stability problem has been looked at by many researchers and conditions are available in many different but equivalent forms.

**vijay** Include a condition? Requires a lot of new notation.

## 5.5 Conclusions

In this chapter, we looked at control across a digital noiseless link. Stability conditions were identified in the two block framework. Some extensions and open problems were also looked at.

# Chapter 6
## Packet-Based Estimation and Control

Outline:

- Problem setup and useful lemmas
- Expected value of covariance
- Probabilistic bounds
- Markov models for packet loss (JLMS)
- Multi-channel and/or multi-sensor?

### 6.1 Introduction

This goal of this chapter is to analyze the problem of state estimation in the case where observations have to travel through a network to reach the estimator and may get lost in the process.

Consider the problem of navigating a vehicle based on the sensor web's estimate of its current position and velocity. The measurements underlying this estimate can be lost or delayed due to the unreliability of the wireless links. The question is, then, what is the amount of data loss that the control loop can tolerate to reliably perform the navigation task? And, can communication protocols be designed to satisfy this constraint? Answering these questions requires a generalization of classical control techniques that explicitly take into account the stochastic nature of the communication channel.

In this setting, the sensor network provides observed data that is used to estimate the state of a controlled system, and this estimate is then in turn used for control purposes. This chapter and the next one study the effect of data loss due to the unreliability of the network links.

The current chapter generalizes the most ubiquitous recursive estimation technique in control—the discrete Kalman filter [**?**]—modeling the arrival of an observation as a random process whose parameters are related to the characteristics of the communication channel, see Figure 6.8. In this setting the statistical convergence of the expected estimation error covariance is characterized and analyzed.

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

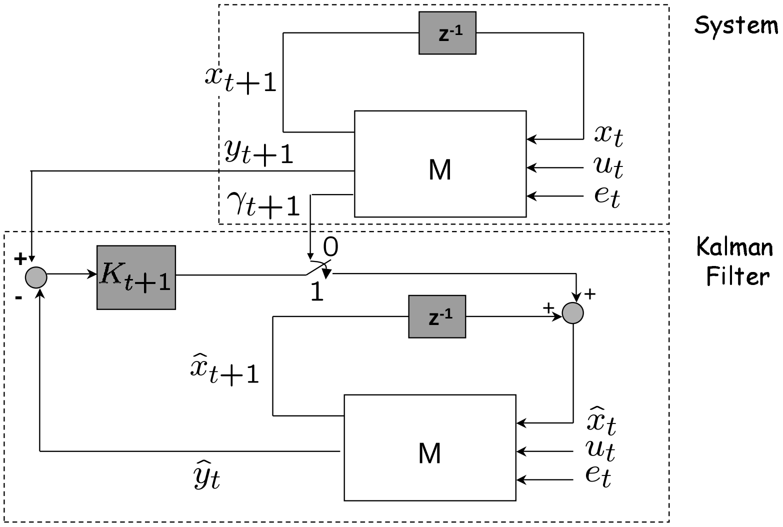


**Figure 6.1: Overview of the system.** The goal is to study the statistical convergence of the expected estimation error covariance of the discrete-time Kalman filter, where the observation, travelling over an unreliable communication channel, can be lost at each time step with probability $1-\bar{\gamma}$.

dynamical system:

$$\begin{aligned} x_{t+1} &= Ax_t + w_t \\ y_t &= Cx_t + v_t, \end{aligned} \tag{6.1}$$

where $x_t \in \mathbb{R}^n$ is the state vector, $y_t \in \mathbb{R}^m$ the output vector, $w_t \in \mathbb{R}^p$ and $v_t \in \mathbb{R}^m$ are Gaussian random vectors with zero mean and covariance matrices $Q \geq 0$ and $R > 0$, respectively. $w_t$ is independent of $w_s$ for $s < t$. Assume that the initial state, $x_0$, is also a Gaussian vector of zero mean and covariance $\Sigma_0$. Under the hypothesis of stabilizability of the pair $(A, Q)$ and detectability of the pair $(A, C)$, the estimation error covariance of the Kalman filter converges to a unique value from any initial condition [**?**].

The assumptions of the Kalman Filter have been relaxed in various ways. Extended Kalman filtering [**?**] attempts to cope with nonlinearities in the model; particle filtering [**?**] is also appropriate for nonlinear models and additionally does not require the noise model to be Gaussian. Recently, more general observation processes have been studied. In particular, in [**?**, **?**] the case in which observations are randomly spaced in time according to a Poisson process has been studied, where the underlying dynamics evolve in continuous time. These authors showed the existence of a lower bound on the arrival rate of the observations below which it is possible to maintain the estimation error covariance below a fixed value, with high probability. However, the results were restricted to scalar SISO systems.

A similar approach is taken in this chapter. While the analysis falls within the framework of discrete time, it provides results for general $n$-dimensional MIMO systems. In particular, it considers a discrete-time system in which the arrival of an observation is a Bernoulli process with parameter $0 < \gamma \bar{<} 1$, and, rather than asking for the estimation error covariance to be bounded with high probability, the study focuses on the asymptotic behavior (in time) of its average. The main contribution is to show that, depending on the eigenvalues of the matrix $A$, and on the structure of the matrix $C$, there exists a critical value $\gamma_c$, such that if the probability of arrival of an observation at time $t$ is $\bar{\gamma} > \gamma_c$, then the expectation of the estimation error covariance is always finite (provided that the usual stabilizability and detectability hypotheses are satisfied). If $\bar{\gamma} \leq \gamma_c$, then the expectation of the estimation error covariance is unbounded. The following analysis provides explicit upper and lower bounds on $\gamma_c$, and shows that they are tight in some special cases.

Philosophically this result can be seen as another manifestation of the well known *uncertainty threshold principle* [**?**, **?**]. This principle states that optimum long-range control of a dynamical system with uncertainty parameters is possible if and only if the uncertainty does not exceed a given threshold. The uncertainty is modeled as white noise scalar sequences acting on the system and control matrices. In our case, the result is for optimal estimation, rather than optimal control, and the uncertainty is due to the random arrival of the observation, with the randomness arising from losses in the network.

## 6.2 Related Work

Studies on filtering with intermittent observations can be tracked back to Nahi [**?**] and Hadidi [**?**]. More recently, this problem has been studied using Jump Linear Systems (JLS) [**?**]. JLS are stochastic hybrid systems characterized by linear dynamics and discrete regime transitions modeled as Markov chains. In the work of Costa et al. [**?**] and Nilsson et al. [**?**, **?**] the Kalman filter with missing observations is modeled as a JLS switching between two discrete regimes: an open loop configuration and a closed loop configuration. Following this approach, these authors obtain convergence criteria for the expected estimation error covariance. However, they restrict their formulation to the steady state case, where the Kalman gain is constant, and they do not assume to know the switching sequence. The resulting process is wide sense stationary [?], and this makes the exact computation of the transition probability and state error covariance possible. Other work on optimal, constant gain filtering can be found in the work of Wang et al. [?], who included the presence of system parameters uncertainty besides missing observations, and Smith et al. [?], who considered the fusion of multiple filters. Instead, we consider the general case of *time varying* Kalman gain. In the presence of missing observations, this filter has a smaller linear minimum mean square error (LMMSE) than its static counterpart, as it is detailed in Section 6.3.

The general case of time-varying Kalman filter with intermittent observations was also studied by Fortmann et al. [?], who derived stochastic equations for the state covariance error. However, they do not statistically characterize its convergence and provide only numerical evidence of the transition to instability, leaving a formal characterization of this as an open problem, which is addressed in this chapter. A somewhat different formulation was considered in [?], where the observations arrival have a bounded delay.

Finally, we point out that our analysis can also be viewed as an instance of Expectation-Maximization (EM) theory. EM is a general framework for doing Maximum Likelihood estimation in missing-data models [?]. Lauritzen [?] shows how EM can be used for general graphical models. In our case, however, the graph structure is a function of the missing data, as there is one graph for each pattern of missing data.

The chapter is organized as follows. In section 6.3 the problem of Kalman filtering with intermittent observations is formally defined. In section 6.4 upper and lower bounds on the expected estimation error covariance of the Kalman filter are provided, along with conditions on the observation arrival probability $\bar{\gamma}$ for which the upper bound converges to a fixed point, and for which the lower bound diverges. Section 6.5 describes some special cases and gives an intuitive understanding of the results. Section 6.6 compares the current approach to previous ones [?] based on jump linear systems.

## 6.3 Problem Formulation

Consider the canonical state estimation problem. The arrival of the observation at time $t$ is modeled as a binary random variable $\gamma_t$, with probability distribution $p_{\gamma_t}(1) = \bar{\gamma}$, and with $\gamma_t$ independent of $\gamma_s$ if $t \neq s$. The output noise $v_t$ is defined in the following way:

$$p(v_t|\gamma_t) = \begin{cases} \mathcal{N}(0, R) & : \quad \gamma_t = 1 \\ \mathcal{N}(0, \sigma^2 I) & : \quad \gamma_t = 0, \end{cases}$$

for some $\sigma^2$ . Therefore, the variance of the observation at time $t$ is $R$ if $\gamma_t$ is 1, and $\sigma^2 I$ otherwise. In reality the absence of observation corresponds to the limiting case of $\sigma \to \infty$. Following this approach the Kalman filter equations are re-derived using a "dummy" observation with a given variance when the real observation does not arrive, and then take the limit as $\sigma \to \infty$.

First define:

$$\hat{x}_{t|t} \overset{\Delta}{=} \mathbb{E}[x_t|\mathbf{y_t}, \gamma_\mathbf{t}] \tag{6.2}$$

$$P_{t|t} \overset{\Delta}{=} \mathbb{E}[(x_t - \hat{x}_t)(x_t - \hat{x}_t)'|\mathbf{y_t}, \gamma_\mathbf{t}] \tag{6.3}$$

$$\hat{x}_{t+1|t} \overset{\Delta}{=} \mathbb{E}[x_{t+1}|\mathbf{y_t}, \gamma_\mathbf{t}] \tag{6.4}$$

$$P_{t+1|t} \overset{\Delta}{=} \mathbb{E}[(x_{t+1} - \hat{x}_{t+1})(x_{t+1} - \hat{x}_{t+1})'|\mathbf{y_t}, \gamma_\mathbf{t}] \tag{6.5}$$

$$\hat{y}_{t+1|t} \overset{\Delta}{=} \mathbb{E}[y_{t+1}|\mathbf{y_t}, \gamma_\mathbf{t}], \tag{6.6}$$

where the vectors $\mathbf{y_t}$ and $\gamma_\mathbf{t}$ are defined as: $\mathbf{y_t} \overset{\Delta}{=} [y_0, \ldots, y_t]'$ and $\gamma_\mathbf{t} \overset{\Delta}{=} [\gamma_0, \ldots, \gamma_t]'$. It is easy to see that:

$$\mathbb{E}[(y_{t+1} - \hat{y}_{t+1|t})(x_{t+1} - \hat{x}_{t+1|t})'|\mathbf{y_t}, \gamma_\mathbf{t+1}] = CP_{t+1|t} \tag{6.7}$$

$$\mathbb{E}[(y_{t+1} - \hat{y}_{t+1|t})(y_{t+1} - \hat{y}_{t+1|t})'|\mathbf{y_t}, \gamma_\mathbf{t+1}] = CP_{t+1|t}C' + \gamma_{t+1}R + (1 - \gamma_{t+1})\sigma^2 I, \tag{6.8}$$

and it follows that the random variables $x_{t+1}$ and $y_{t+1}$, conditioned on the output $\mathbf{y_t}$ and on the arrivals $\gamma_\mathbf{t+1}$, are jointly gaussian with mean

$$\mathbb{E}[x_{t+1}, y_{t+1}|\mathbf{y_t}, \gamma_\mathbf{t+1}] = \begin{pmatrix} \hat{x}_{t+1|t} \\ C\hat{x}_{t+1|t} \end{pmatrix},$$

and covariance

$$COV(x_{t+1}, y_{t+1}|\mathbf{y_t}, \gamma_\mathbf{t+1}) =$$
$$= \begin{pmatrix} P_{t+1|t} & P_{t+1|t}C' \\ CP_{t+1|t} & CP_{t+1|t}C' + \gamma_{t+1}R + (1 - \gamma_{t+1})\sigma^2 I \end{pmatrix}.$$

Hence, the Kalman filter equations are modified as follows:

$$\hat{x}_{t+1|t} = A\hat{x}_{t|t} \tag{6.9}$$

$$P_{t+1|t} = AP_{t|t}A' + Q \tag{6.10}$$

$$\hat{x}_{t+1|t+1} = \hat{x}_{t+1|t} + P_{t+1|t}C'(CP_{t+1|t}C' + \gamma_{t+1}R + (1 - \gamma_{t+1})\sigma^2 I)^{-1}(y_{t+1} - C\hat{x}_{t+1|t}) \tag{6.11}$$

$$P_{t+1|t+1} = P_{t+1|t} - P_{t+1|t}C'(CP_{t+1|t}C' + \gamma_{t+1}R + (1 - \gamma_{t+1})\sigma^2 I)^{-1}CP_{t+1|t}. \tag{6.12}$$

Taking the limit as $\sigma \to \infty$, the update equations (6.11) and (6.12) can be rewritten as follows:

$$\hat{x}_{t+1|t+1} = \hat{x}_{t+1|t} + \gamma_{t+1}K_{t+1}(y_{t+1} - C\hat{x}_{t+1|t}) \tag{6.13}$$

$$P_{t+1|t+1} = P_{t+1|t} - \gamma_{t+1}K_{t+1}CP_{t+1|t}, \tag{6.14}$$

where $K_{t+1} = P_{t+1|t}C'(CP_{t+1|t}C' + R)^{-1}$ is the Kalman gain matrix for the standard ARE. Note that performing this limit corresponds *exactly* to propagating the previous state when there is no observation update available at time t. It is important to point out the main difference from the standard Kalman filter formulation: both $\hat{x}_{t+1|t+1}$ and $P_{t+1|t+1}$ are now random variables, being a function of $\gamma_{t+1}$, which is itself random.

Equations (6.13)-(6.66) give the minimum state error variance filter given the observations $\{y_t\}$ and their arrival sequence$\{\gamma_t\}$, i.e. $\hat{x}_t^{tm} = \mathbb{E}[x_t|y_t, \ldots, y_1, \gamma_t, \ldots, \gamma_1]$. As a consequence, the filter proposed in this paper is necessarily time-varying and stochastic since it depends on the arrival sequence. The filters that have been proposed so far using JLS theory [**?**, **?**] give the minimum state error variance filters assuming that only the observations $\{y_t\}$ and the knowledge on the last arrival $\gamma_t$ are available, i.e. $\hat{x}_t^{JLS} = \mathbb{E}[x_t|y_t, \ldots, y_1, \gamma_t]$. Therefore, the filter given by Equations (6.13)-(6.66) gives a better performance than its JLS counterparts, since it exploits additional information regarding the arrival sequence.

Given the new formulation, we now study the Riccati equation of the state error covariance matrix in the specific case when the arrival process of the observation is time-independent, i.e. $\bar{\gamma}_t = \bar{\gamma}$ for all time. This will allow us to provide deterministic upper and lower bounds on its expectation. We then characterize the convergence of these upper and lower bounds, as a function of the arrival probability $\bar{\gamma}$ of the observation.

## 6.4 Convergence conditions and transition to instability

It is easy to verify that the modified Kalman filter formulation in Equations (6.10) and (6.66) can be rewritten as follows:

$$P_{t+1} = AP_tA' + Q - \gamma_t\, AP_tC'(CP_tC' + R)^{-1}CP_tA', \tag{6.15}$$

where we use the simplified notation $P_t = P_{t|t-1}$. Since the sequence $\{\gamma_t\}_0^\infty$ is random, the modified Kalman filter iteration is stochastic and cannot be determined off-line. Therefore, only statistical properties can be deduced.

In this section we show the existence of a critical value $\gamma_c$ for the arrival probability of the observation update, such that for $\bar{\gamma} > \gamma_c$ the mean state covariance $\mathbb{E}[P_t]$ is bounded for all initial conditions, and for $\bar{\gamma} \leq \gamma_c$ the mean state covariance diverges for some initial condition. We also find a lower bound $\gamma_{min}$, and upper bound $\gamma_{max}$, for the critical probability $\gamma_c$, i.e., $\gamma_{min} \leq \gamma_c \leq \gamma_{max}$. The lower bound is expressed in closed form; the upper bound is the solution of a linear matrix inequality (LMI). In some special cases the two bounds coincide, giving a tight estimate. Finally, we present numerical algorithms to compute a lower bound $\bar{S}$, and upper bound $\bar{V}$, for $\lim_{t\to\infty} \mathbb{E}[P_t]$, when it is bounded.

First, we define the modified algebraic Riccati equation (MARE) for the Kalman filter with intermittent observations as follows,

$$g_{\bar{\gamma}}(X) = AXA' + Q - \bar{\gamma}\, AXC'(CXC' + R)^{-1}CXA'. \tag{6.16}$$

Our results derive from two principal facts: the first is that concavity of the modified algebraic Riccati equation for our filter with intermittent observations allows use of Jensen's inequality to find an upper bound on the mean state covariance; the second is that all the operators we use to estimate upper and lower bounds are monotonically increasing, therefore if a fixed point exists, it is also stable.

We formally state all main results in form of theorems. Omitted proofs appear in the Appendix. The first theorem expresses convergence properties of the MARE.

**Theorem 6.1.** *Consider the operator* $\phi(K,X) = (1-\bar{\gamma})(AXA' + Q) + \bar{\gamma}(FXF' + V)$, *where* $F = A + KC$, $V = Q + KRK'$. *Suppose there exists a matrix* $\tilde{K}$ *and a positive definite matrix* $\tilde{P}$ *such that*

$$\tilde{P} > 0 \quad \textit{and} \quad \tilde{P} > \phi(\tilde{K}, \tilde{P})$$

*Then,*

- *for any initial condition* $P_0 \geq 0$, *the MARE converges, and the limit is independent of the initial condition:*

$$\lim_{t\to\infty} P_t = \lim_{t\to\infty} g^t_{\bar{\gamma}}(P_0) = \overline{P}$$

- $\overline{P}$ *is the unique positive semidefinite fixed point of the MARE.*

The next theorem states the existence of a sharp transition.

**Theorem 6.2.** *If* $(A, Q^{\frac{1}{2}})$ *is controllable,* $(A, C)$ *is detectable, and* $A$ *is unstable, then there exists a* $\gamma_c \in [0,1)$ *such that*

$$\lim_{t\to\infty} \mathbb{E}[P_t] = +\infty \, \textit{for} \; 0 \leq \bar{\gamma} \leq \gamma_c \quad \textit{and} \quad \exists P_0 \geq 0 \tag{6.17}$$

$$\mathbb{E}[P_t] \leq M_{P_0} \quad \forall t \, \textit{for} \; \gamma_c < \bar{\gamma} \leq 1 \quad \textit{and} \quad \forall P_0 \geq 0 \tag{6.18}$$

*where* $M_{P_0} > 0$ *depends on the initial condition* $P_0 \geq 0$[1].

The next theorem gives upper and lower bounds for the critical probability $\gamma_c$.

**Theorem 6.3.** *Let*

$$\gamma_{min} = inf\,[\bar{\gamma} : \exists \hat{S} \mid \hat{S} = (1-\bar{\gamma})A\hat{S}A' + Q] = 1 - \frac{1}{\alpha^2} \tag{6.19}$$

$$\gamma_{max} = inf\,[\bar{\gamma} : \exists (\hat{K}, \hat{X}) \mid \hat{X} > \phi(\hat{K}, \hat{X})] \tag{6.20}$$

*where* $\alpha = \max_i |\sigma_i|$ *and* $\sigma_i$ *are the eigenvalues of* $A$. *Then*

$$\gamma_{min} \leq \gamma_c \leq \gamma_{max}. \tag{6.21}$$

[1] We use the notation $\lim_{t\to\infty} A_t = +\infty$ when the sequence $A_t \geq 0$ is not bounded; i.e., there is no matrix $M \geq 0$ such that $A_t \leq M, \forall t$.

Finally, the following theorem gives an estimate of the limit of the mean covariance matrix $\mathbb{E}[P_t]$, when this is bounded.

**Theorem 6.4.** *Assume that $(A, Q^{\frac{1}{2}})$ is controllable, $(A, C)$ is detectable and $\bar{\gamma} > \gamma_{max}$, where $\gamma_{max}$ is defined in Theorem 6.3. Then*

$$0 < S_t \leq \mathbb{E}[P_t] \leq V_t \qquad \forall\, \mathbb{E}[P_0] \geq 0 \tag{6.22}$$

*where $\lim_{t\to\infty} S_t = \bar{S}$ and $\lim_{t\to\infty} V_t = \bar{V}$, where $\bar{S}$ and $\bar{V}$ are solution of the respective algebraic equations*
*$\bar{S} = (1-\bar{\gamma})A\bar{S}A' + Q$ and $\bar{V} = g_{\bar{\gamma}}(\bar{V})$.*

The previous theorems give lower and upper bounds for both the critical probability $\gamma_c$ and for the mean error covariance $\mathbb{E}[P_t]$. The lower bound $\gamma_{min}$ is expressed in closed form. We now resort to numerical algorithms for the computation of the remaining bounds $\gamma_{max}, \bar{S}$ and $\bar{V}$.

The computation of the upper bound $\gamma_{max}$ can be reformulated as the iteration of an LMI feasibility problem. To establish this we need the following theorem:

**Theorem 6.5.** *If $(A, Q^{\frac{1}{2}})$ is controllable and $(A, C)$ is detectable, then the following statements are equivalent:*

- $\exists \bar{X}$ *such that* $\bar{X} > g_{\bar{\gamma}}(\bar{X})$
- $\exists \bar{K}, \bar{X} > 0$ *such that* $\bar{X} > \phi(\bar{K}, \bar{X})$
- $\exists \bar{Z}$ *and* $0 < \bar{Y} \leq I$ *such that*

$$\Psi_{\bar{\gamma}}(Y, Z) = \begin{bmatrix} Y & \sqrt{\bar{\gamma}}(YA + ZC) & \sqrt{1-\bar{\gamma}}YA \\ \sqrt{\bar{\gamma}}(A'Y + C'Z') & Y & 0 \\ \sqrt{1-\bar{\gamma}}A'Y & 0 & Y \end{bmatrix} > 0.$$

*Proof.* (a)$\Longrightarrow$(b) If $\bar{X} > g_{\bar{\gamma}}(\bar{X})$ exists, then $\bar{X} > 0$ by Lemma 6.1(g). Let $\bar{K} = K_{\bar{X}}$. Then $\bar{X} > g_{\bar{\gamma}}(\bar{X}) = \phi(\bar{K}, \bar{X})$ which proves the statement.
(b)$\Longrightarrow$(a) Clearly $\bar{X} > \phi(\bar{K}, \bar{X}) \geq g_{\bar{\gamma}}(\bar{X})$ which proves the statement.
(b)$\Longleftrightarrow$(c) Let $F = A + KC$, then:

$$X > (1-\bar{\gamma})AXA' + \bar{\gamma}FXF' + Q + \bar{\gamma}KRK'$$

is equivalent to

$$\begin{bmatrix} X - (1-\bar{\gamma})AXA' & \sqrt{\bar{\gamma}}F \\ \sqrt{\bar{\gamma}}F' & X^{-1} \end{bmatrix} > 0,$$

where we used the Schur complement decomposition and the fact that $X - (1-\bar{\gamma})AXA' \geq \bar{\gamma}FXF' + Q + \bar{\gamma}KRK' \geq Q > 0$. Using one more time the Schur complement decomposition on the first element of the matrix we obtain

$$\Theta = \begin{bmatrix} X & \sqrt{\bar\gamma}F & \sqrt{1-\bar\gamma}A \\ \sqrt{\bar\gamma}F' & X^{-1} & 0 \\ \sqrt{1-\bar\gamma}A' & 0 & X^{-1} \end{bmatrix} > 0.$$

This is equivalent to

$$\begin{aligned} \bar\gamma &= \begin{bmatrix} X^{-1} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \Theta \begin{bmatrix} X^{-1} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} > 0 \\ &= \begin{bmatrix} X^{-1} & \sqrt{\bar\gamma}X^{-1}F & \sqrt{1-\bar\gamma}X^{-1}A \\ \sqrt{\bar\gamma}F'X^{-1} & X^{-1} & 0 \\ \sqrt{1-\bar\gamma}A'X^{-1} & 0 & X^{-1} \end{bmatrix} > 0. \end{aligned}$$

Let us consider the change of variable $Y = X^{-1} > 0$ and $Z = X^{-1}K$, in which case the previous LMI is equivalent to:

$$\Psi(Y,Z) =$$

$$= \begin{bmatrix} Y & \sqrt{\bar\gamma}(YA+ZC) & \sqrt{1-\bar\gamma}YA \\ \sqrt{\bar\gamma}(A'Y+C'Z') & Y & 0 \\ \sqrt{1-\bar\gamma}A'Y & 0 & Y \end{bmatrix} > 0.$$

Since $\Psi(\alpha Y, \alpha K) = \alpha\Psi(Y,K)$, then $Y$ can be restricted to $Y \le I$, which completes the theorem. □

Combining theorems 6.3 and 6.5 we immediately have the following corollary

**Corollary 6.5.1.** *The upper bound $\gamma_{max}$ is given by the solution of the following optimization problem,*

$$\gamma_{max} = \text{argmin}_{\bar\gamma}\Psi_{\bar\gamma}(Y,Z) > 0, \quad 0 \le Y \le I.$$

This is a quasi-convex optimization problem in the variables $(\bar\gamma, Y, Z)$ and the solution can be obtained by iterating LMI feasibility problems and using bisection for the variable $\bar\gamma$, as shown in [**?**].

The lower bound $\bar S$ for the mean covariance matrix can be easily obtained via standard Lyapunov Equation solvers. The upper bound $\bar V$ can be found by iterating the MARE or by solving a semidefinite programming (SDP) problem as shown in the following theorem.

**Theorem 6.6.** *If $\bar\gamma > \gamma_{max}$, then the matrix $\bar V = g_{\bar\gamma}(\bar V)$ is given by:*

1. *$\bar V = \lim_{t\to\infty} V_t$; $V_{t+1} = g_{\bar\gamma}(V_t)$ where $V_0 \ge 0$*

2.

$$\begin{aligned} &\text{argmax}_V \quad Trace(V) \\ &\textit{subject to} \begin{bmatrix} AVA' - V + Q & \sqrt{\bar\gamma}AVC' \\ \sqrt{\bar\gamma}CVA' & CVC' + R \end{bmatrix} \ge 0, \quad V \ge 0 \end{aligned}$$

*Proof.* (a) It follows directly from Theorem 6.1.

(b) It can be obtained by using the Schur complement decomposition on the equation $V \leq g_{\bar{\gamma}}(V)$. Clearly the solution $\bar{V} = g_{\bar{\gamma}}(\bar{V})$ belongs to the feasible set of the optimization problem. We now show that the solution of the optimization problem is the fixed point of the MARE. Suppose it is not, i.e., $\hat{V}$ solves the optimization problem but $\hat{V} \neq g_{\bar{\gamma}}(\hat{V})$. Since $\hat{V}$ is a feasible point of the optimization problem, then $\hat{V} < g_{\bar{\gamma}}(\hat{V}) = \hat{\hat{V}}$. However, this implies that $Trace(\hat{V}) < Trace(\hat{\hat{V}})$, which contradicts the hypothesis of optimality of matrix $\hat{V}$. Therefore $\hat{V} = g_{\bar{\gamma}}(\hat{V})$ and this concludes the theorem. □

## 6.5 Special Cases and Examples

In this section we present some special cases in which upper and lower bounds on the critical value $\gamma_c$ coincide, and give some examples. From Theorem 6.1, it follows that if there exists a $\tilde{K}$ such that $F$ is the zero matrix, then the convergence condition of the MARE is for $\bar{\gamma} > \gamma_c = 1 - 1/\alpha^2$, where $\alpha = \max_i |\sigma_i|$, and $\sigma_i$ are the eigenvalues of $A$.

- **C is invertible**. In this case a choice of $K = -AC^{-1}$ makes $F = 0$. Note that the scalar case also falls under this category. Figure (2) shows a plot of the steady state of the upper and lower bounds versus $\bar{\gamma}$ in the scalar case. The discrete time LTI system used in this simulation has $A = -1.25$, $C = 1$, with $v_t$ and $w_t$ having zero mean and variance $R = 2.5$ and $Q = 1$, respectively. For this system we have $\gamma_c = 0.36$. The transition clearly appears in the figure, where we see that the steady state value of both upper and lower bound tends to infinity as $\bar{\gamma}$ approaches $\gamma_c$. The dashed line shows the lower bound, the solid line the upper bound, and the dash-dot line shows the asymptote.

- **A has a single unstable eigenvalue**. In this case, regardless of the dimension of $C$ (and as long as the pair $(A, C)$ is detectable), we can use Kalman decomposition to bring to zero the unstable part of $F$ and thereby obtain tight bounds. Figure (3) shows a plot for the system $A = \begin{pmatrix} 1.25 & 1 & 0 \\ 0 & 0.9 & 7 \\ 0 & 0 & 0.6 \end{pmatrix}$, $C = \begin{pmatrix} 1 & 0 & 2 \end{pmatrix}$ with $v_t$ and $w_t$ having zero mean and variance $R = 2.5$ and $Q = 20 \cdot I_{3\times 3}$, respectively. This time, the asymptotic value for trace of upper and lower bound is plotted versus $\bar{\gamma}$. Once again $\gamma_c = 0.36$.

In general $F$ cannot always be made zero and we have shown that while a lower bound on $\gamma_c$ can be written in closed form, an upper bound on $\gamma_c$ is the result of a LMI. Figure (4) shows an example where upper and lower bounds

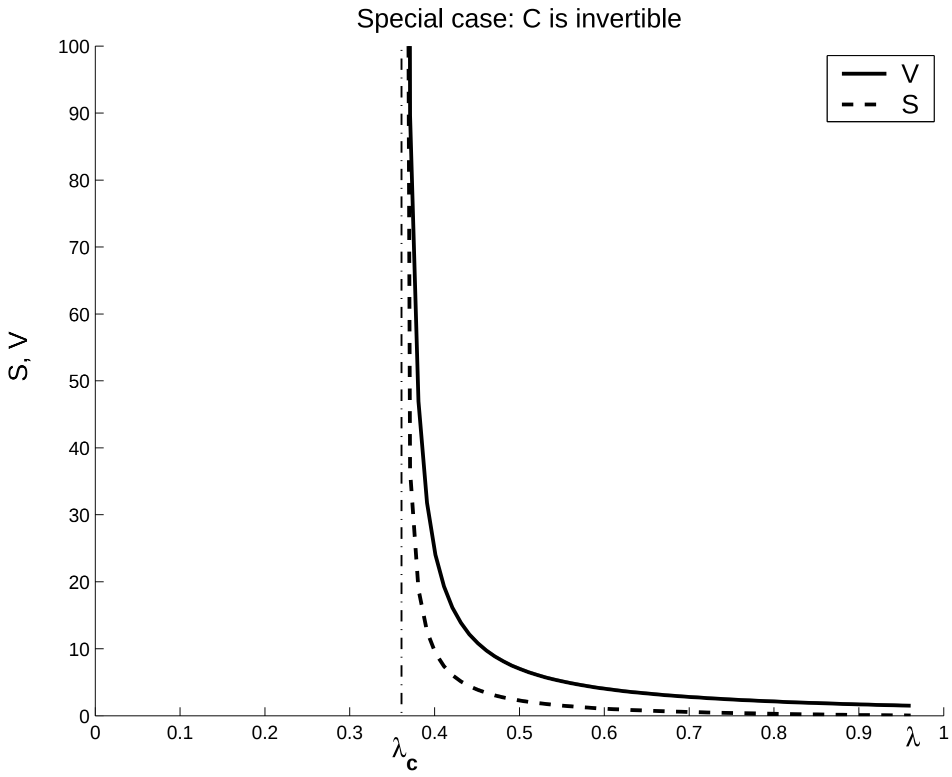


**Figure 6.2:** Example of transition to instability in the scalar case. The dashed line shows the asymptotic value of the lower bound ($\bar{S}$), the solid line the asymptotic value of the upper bound ($\bar{V}$), and the dash-dot line shows the asymptote ($\gamma_c$).

have different convergence conditions. The system used for this simulation is $A = \begin{pmatrix} 1.25 & 0 \\ 1 & 1.1 \end{pmatrix}$, $C = \begin{pmatrix} 1 & 1 \end{pmatrix}$ with $v_t$ and $w_t$ having zero mean and variance $R = 2.5$ and $Q = 20 \cdot I_{2\times 2}$, respectively.

Finally, in Figure (5) we report results of another experiment, plotting the state estimation error for the scalar system used above at two similar values of $\bar{\gamma}$, one being below and one above the critical value. We note a dramatic change in the error at $\gamma_c \approx 0.36$. The figure on the left shows the estimation error with $\bar{\gamma} = 0.3$. The figure on the right shows the estimation error for the same system evolution with $\bar{\gamma} = 0.4$. In the first case the estimation error grows dramatically, making it practically useless for control purposes. In the second case, a small increase in $\bar{\gamma}$ reduces the estimation error by approximately three orders of magnitude.

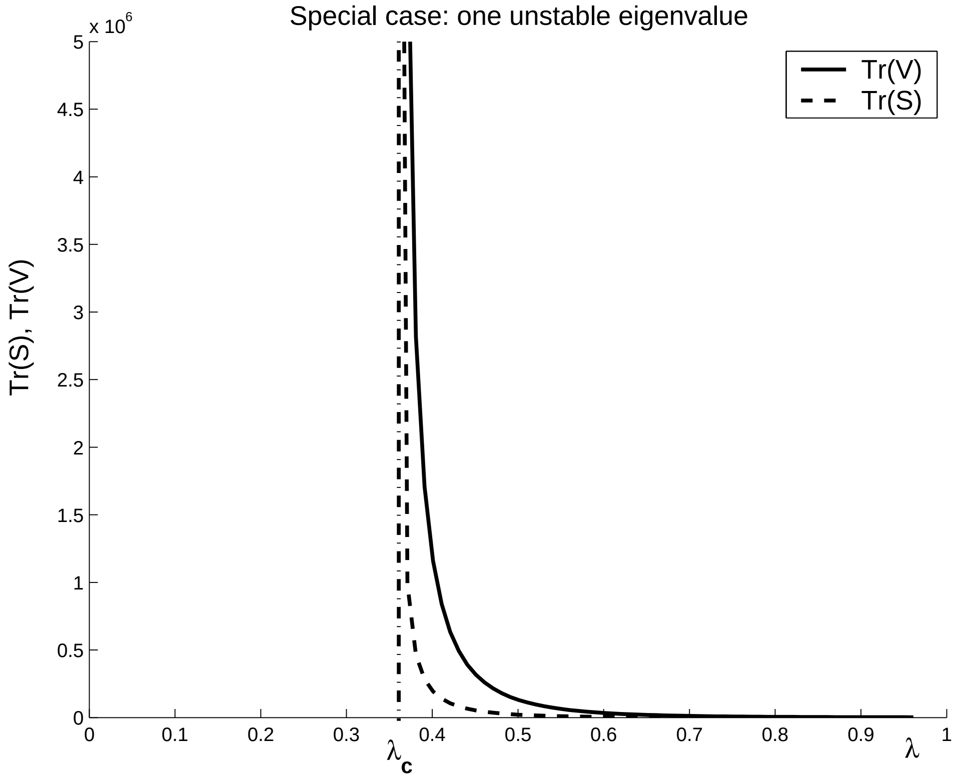


**Figure 6.3:** Example of transition to instability with a single unstable eigenvalue in the MIMO case. The dashed line shows the asymptotic value of the trace of lower bound ($\bar{S}$), the solid line the asymptotic value of trace of the upper bound ($\bar{V}$), and the dash-dot line shows the asymptote ($\gamma_c$).

## 6.6 Static versus dynamic Kalman gain

In this section we compare the performance of filtering with static and dynamic gain for a scalar discrete system. For the static estimator we follow the jump linear system approach of [**?**]. The scalar static estimator case has been also worked out in [**?**].

Consider the dynamic state estimator

$$\begin{aligned}\hat{x}^d_{t+1} &= A\hat{x}^d_t + \gamma_t K^d_t (y_t - \hat{y}_t)\\ K^d_t &= AP_tC'(CP_tC' + R)^{-1}\\ P_{t+1} &= AP_tA' + Q - \gamma_t K^d_t C P_t A' \end{aligned} \tag{6.23}$$

where the Kalman gain $K^d_t$ is time-varying. Also consider the static state estimator

$$\hat{x}^s_{t+1} = A\hat{x}^d_t + \gamma_t K_s (y_t - \hat{y}_t) \tag{6.24}$$

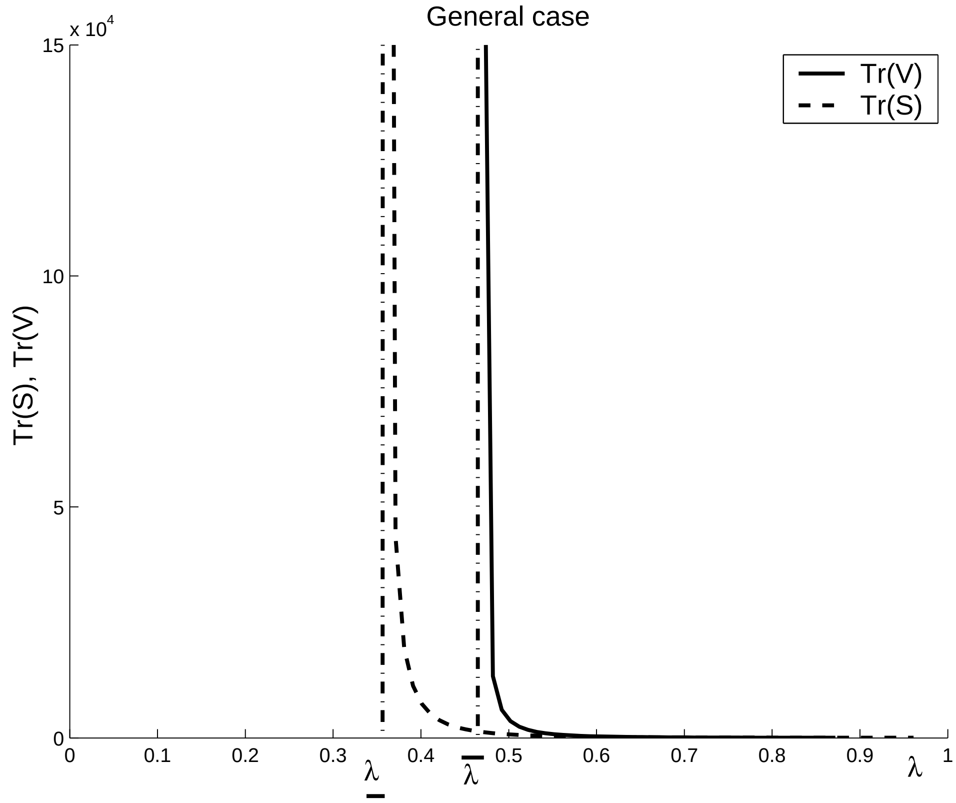


**Figure 6.4:** Transition to instability in the general case, with arbitrary A and C. In this case lower and upper bounds do not have the same asymptote.

where the estimator gain $K_s$ is constant. If no data arrives, i.e. $\gamma_t = 0$, both estimators simply propagate the state estimate of the previous time-step.

The performance of the dynamic state estimator (6.23) has been analyzed in the previous sections. The performance of static state estimator (6.24), instead, can be readily obtained using jump linear system theory [**?**, **?**]. To do so, let us consider the estimator error $e^s_{t+1} \stackrel{\Delta}{=} x_{t+1} - \hat{x}^s_{t+1}$. Substituting Equations (6.1) for $x_{t+1}$ and (6.24) for $\hat{x}^s_{t+1}$, we obtain the dynamics of the estimation error:

$$e^s_{t+1} = (A - \gamma_t K_s C)e^s_t + v_t + \gamma_t K_s w_t. \tag{6.25}$$

Using the same notation of Chapter 6 in Nilsson [**?**], where he considers the general system:

$$z_{k+1} = \Phi(r_k)z_k + \Gamma(r_k)e_k,$$

the system (6.25) can be seen as jump linear system switching between two

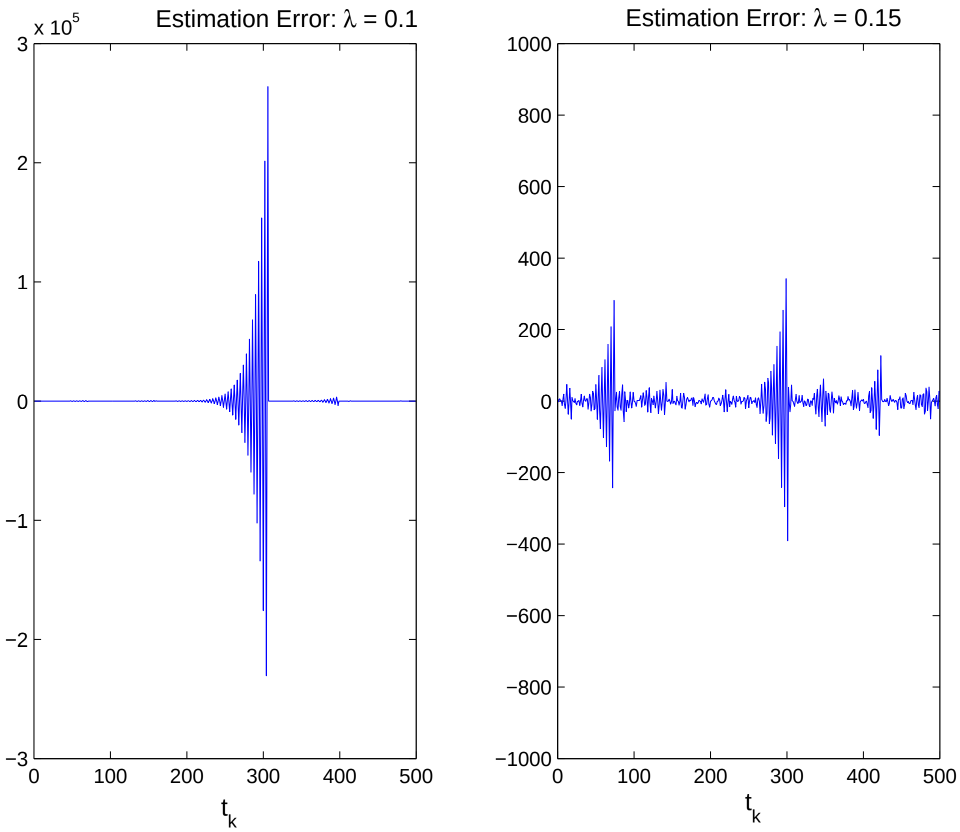


**Figure 6.5:** Estimation error for $\bar{\gamma}$ below (left) and above (right) the critical value

states $r_k \in \{1, 2\}$ given by:

$$\begin{aligned} \Phi(1) &= A - K_s C \qquad & \Gamma(1) &= [1 \; K_s] \\ \Phi(2) &= A & \Gamma(2) &= [1 \; 0], \end{aligned}$$

where the noise covariance $\mathbb{E}[e_k e_k'] = R_e$, the transition probability matrix $Q_\pi$ and the steady state probability distribution $\pi^\infty$ are given by:

$$R_e = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}, \quad Q_\pi = \begin{bmatrix} \bar{\gamma} & 1-\bar{\gamma} \\ \bar{\gamma} & 1-\bar{\gamma} \end{bmatrix}, \quad \pi^\infty = \begin{bmatrix} \bar{\gamma} & 1-\bar{\gamma} \end{bmatrix}.$$

Following the methodology proposed in Nilsson [**?**] is possible to show that the system above is mean square stable, i.e. $\lim_{t\to\infty} \mathbb{E}[(e_t^s)' e_t^s] = 0$ if and only if the transition probability is

$$\bar{\gamma} < \bar{\gamma}_s = \frac{1}{1 - \left(1 - \frac{K_s C}{A}\right)^2} \left(1 - \frac{1}{A^2}\right). \tag{6.26}$$

If the system is mean square stable, the steady state error covariance $P^s_\infty = \lim_{t\to\infty} \mathbb{E}[e^s_t (e^s_t)']$ is given by:

$$P^s_\infty = \frac{Q + K_s^2 R}{1 - \bar{\gamma}(A - K_s C)^2 - (1-\bar{\gamma})A^2}. \tag{6.27}$$

Calculations to obtain Equations (6.26) and (6.27) are tedious but straightforward, therefore they are omitted.

It is immediately evident that the critical transition probability $\bar{\gamma}_s$ of the estimator (6.24) using a static gain is always greater than the critical transition probability $\gamma_c$ of the estimator (6.23) which adopts a dynamic gain, in fact

$$\bar{\gamma}_s = \gamma_c \, \frac{1}{1 - \left(1 - \frac{K_s C}{A}\right)^2}$$

and the two probabilities are equal only when $K_s = \frac{A}{C}$.

A natural choice for the static estimator gain $K_s$ is the steady state Kalman gain $K_{SS}$ of the closed loop system ($r = 1$), which is always different from $\frac{A}{C}$. For the scalar system considered in the previous section, where $A = -1.5$, $C = 1$, $Q = 1$, $R = 2.5$, this is given by $K_{SS} = -0.70$, while the gain for largest mean square stability range is $K_s = \frac{A}{C} = -1.25$. In the special case when the arrival probability is known, another natural choice for the estimator gain $K$ is obtained by substituting the error covariance solution of $\bar{P} = g_{\bar{\gamma}}(\bar{P})$ into the equation for the Kalman filter gain $K_{\bar{\gamma}} = A\bar{P}C'(C\bar{P}C' + R)^{-1}$. For example, assuming $\bar{\gamma} = 0.6$, then $\bar{P} = 7.32$ and $K_{\bar{\gamma}} = -0.93$. Figure 6.6 shows all of these cases, comparing them with the upper bound on the state error covariance $\bar{V}$ of the dynamic estimator (6.23) that can be computed as indicated in Theorem 6. The steady state error covariance of the static predictor for the three different gains is always greater then our upper bound $\bar{V}$. This is not surprising, since the dynamic estimator is optimal over all possible estimators as shown in Section II. Note that the static predictor with static gain $K_{\bar{\gamma}}$ (designed for $\bar{\gamma} = 0.6$) achieves the same state error covariance predicted by our upper bound for the optimal dynamic filter when $\bar{\gamma} = 0.6$. However, the empirical error state covariance is on average better than the static filter, as shown in Figure 6.7. This is to be expected, since the solution of MARE gives only an upper bound of the true expected state covariance of the time-varying filter. Moreover, it is worth stressing that if the arrival probability is different from the one used to design the static gain, the performance of the static filter will degrade considerably, while the time-varying filter will still perform optimally since it does not require knowledge of $\bar{\gamma}$. From this example, it seems that the upper bound for the dynamic estimator $\bar{V}$ gives en estimate of the minimum steady state covariance that can be achieved with a static estimator for any given arrival probability if the static gain $K_s$ is chosen optimally. Then the MARE could be used to find the minimum steady state covariance and

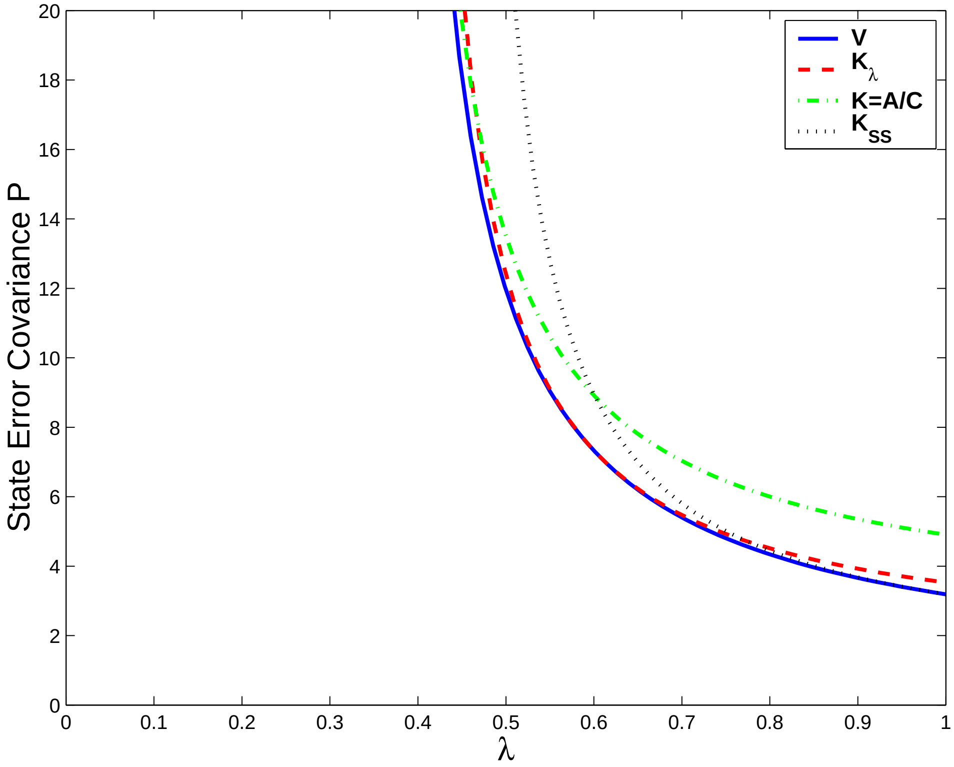


**Figure 6.6:** Error covariance bound $\bar{V}$ for dynamic predictor obtained from our theory and steady state error covariance for three natural static predictors obtained from JLS theory.

then the corresponding steady state modified Kalman gain, thus providing a useful tool for optimal static estimator design. Future work will explore this possibility.

## 6.7 Appendix A

In order to give complete proofs of our main theorems, we need to prove some preliminary lemmas. The first one shows some useful properties of the MARE.

**Lemma 6.1.** *Let the operator*

$$\phi(K,X) = (1-\bar{\gamma})(AXA'+Q) + \bar{\gamma}(FXF'+V) \tag{6.28}$$

*where* $F = A+KC, \quad V = Q+KRK'$*. Assume* $X \in \mathbb{S} = \{S \in \mathbb{R}^{n\times n} | S \geq 0\}$, $R > 0$, $Q \geq 0$, *and* $(A, Q^{\frac{1}{2}})$ *is controllable. Then the following facts are true:*

1. *With* $K_X = -AXC'\left(CXC'+R\right)^{-1}$, $g_{\bar{\gamma}}(X) = \phi(K_X, X)$

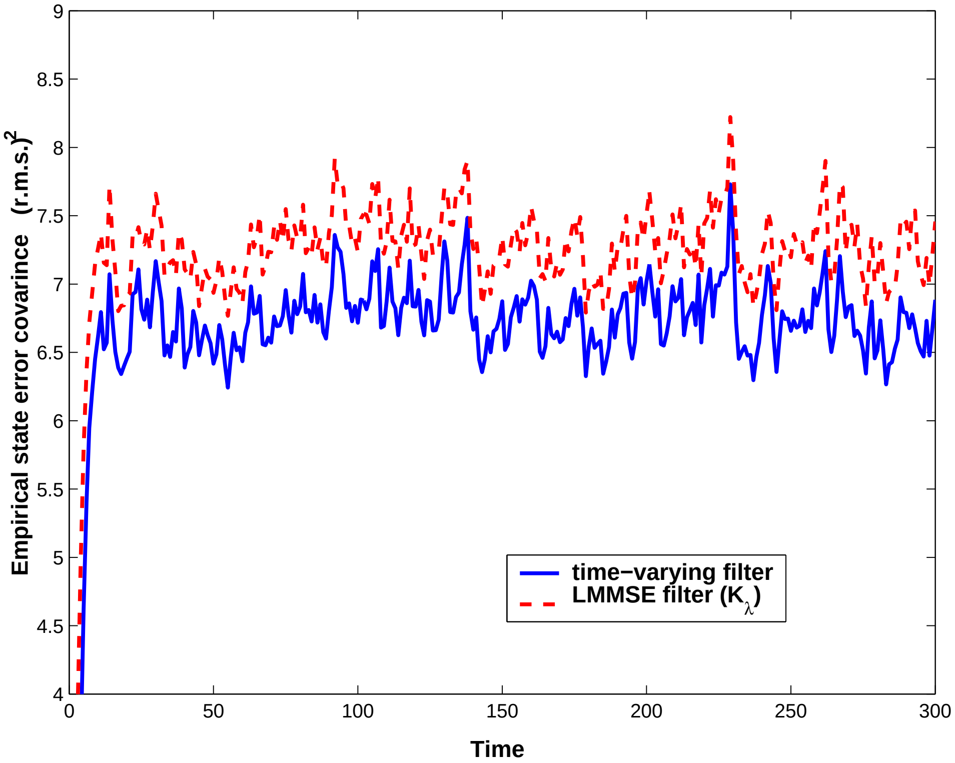


**Figure 6.7:** Empirical state error covariance of our time-varying filter and the linear minimum mean square error estimator (LMMSEE) [**?**] obtained by using the optimal static kalman gain $K_{\bar{\gamma}}$. The curves are obtained by averaging 10000 Monte Carlo simulations for $t = 1, \ldots, 300$, with the values of the input noise $(v_t, w_t)$ and the arrival sequence $\gamma_t$ generated randomly. Both filters were compared under the same conditions.

2. $g_{\bar{\gamma}}(X) = \min_K \phi(K, X) \leq \phi(K, X), \ \forall K$

3. *If* $X \leq Y$, *then* $g_{\bar{\gamma}}(X) \leq g_{\bar{\gamma}}(Y)$

4. *If* $\bar{\gamma}_1 \leq \bar{\gamma}_2$ *then* $g_{\bar{\gamma}_1}(X) \geq g_{\bar{\gamma}_2}(X)$

5. *If* $\alpha \in [0, 1]$, *then*
   $g_{\bar{\gamma}}(\alpha X + (1 - \alpha)Y) \geq \alpha g_{\bar{\gamma}}(X) + (1 - \alpha) g_{\bar{\gamma}}(Y)$

6. $g_{\bar{\gamma}}(X) \geq (1 - \bar{\gamma}) A X A' + Q$

7. *If* $\bar{X} \geq g_{\bar{\gamma}}(\bar{X})$, *then* $\bar{X} > 0$

8. *If* $X$ *is a random variable, then*
   $(1 - \bar{\gamma}) A \mathbb{E}[X] A' + Q \leq \mathbb{E}[g_{\bar{\gamma}}(X)] \leq g_{\bar{\gamma}}(\mathbb{E}[X])$

*Proof.* (a) Define $F_X = A + K_X C$, and observe that

$$\begin{aligned} F_X X C' + K_X R &= (A + K_X C) X C' + K_X R \\ &= AXC' + K_X (CXC' + R) \\ &= 0. \end{aligned}$$

Next, we have

$$\begin{aligned} g_{\bar\gamma}(X) &= (1-\bar\gamma)(AXA' + Q) + \bar\gamma(AXA' + Q - AXC'\left(CXC' + R\right)^{-1} CXA') \\ &= (1-\bar\gamma)(AXA' + Q) + \bar\gamma(AXA' + Q + K_X CXA') \\ &= (1-\bar\gamma)(AXA' + Q) + \bar\gamma(F_X XA' + Q) \\ &= (1-\bar\gamma)(AXA' + Q) + \bar\gamma(F_X XA' + Q) + (F_X XC' + K_X R)K' \\ &= \phi(K_X, X) \end{aligned}$$

(b) Let $\psi(K, X) = (A + KC)X(A + KC)' + KRK' + Q$. Note that

$$\mathrm{argmin}_K \phi(K, X) = \mathrm{argmin}_K FXF' + V = \mathrm{argmin}_K \psi(X, K).$$

Since $X, R \geq 0$, $\phi(K, X)$ is quadratic and convex in the variable $K$, therefore the minimizer can be found by solving $\frac{\partial\psi(K,X)}{\partial K} = 0$, which gives:

$$2(A + KC)XC' + 2KR = 0 \Longrightarrow K = -AXC'\left(CXC' + R\right)^{-1}.$$

Since the minimizer corresponds to $K_X$ defined above, the fact follows from fact (1)

(c) Note that $\phi(K, X)$ is affine in $X$. Suppose $X \leq Y$. Then

$$g_{\bar\gamma}(X) = \phi(K_X, X) \leq \phi(K_Y, X) \leq \phi(K_Y, Y) = g_{\bar\gamma}(Y).$$

This completes the proof.

(d) Note that $AXC'(CXC' + R)^{-1}CXA \geq 0$. Then

$$\begin{aligned} g_{\bar\gamma_1}(X) &= AXA' + Q - \bar\gamma_1\, AXC'(CXC' + R)^{-1}CXA \\ &\geq AXA' + Q - \bar\gamma_2\, AXC'(CXC' + R)^{-1}CXA \\ &= g_{\bar\gamma_2}(X) \end{aligned}$$

(e) Let $Z = \alpha X + (1-\alpha)Y$ where $\alpha \in [0, 1]$. Then we have

$$\begin{aligned} g_{\bar\gamma}(Z) &= \phi(K_Z, Z) \\ &= \alpha(A + K_Z\, C)X(A + K_Z\, C)' + (1-\alpha)(A + K_Z\, C)Y(A + K_Z\, C)' + \\ &+ (\alpha + 1 - \alpha)(K_Z\, R\, K_Z' + Q) \\ &= \alpha\phi(K_Z, X) + (1-\alpha)\phi(K_Z, Y) \\ &\geq \alpha\phi(K_X, X) + (1-\alpha)\phi(K_Y, Y) \\ &= \alpha g_{\bar\gamma}(X) + (1-\alpha) g_{\bar\gamma}(Y). \end{aligned}$$

(f) Note that $F_X X F_X' \geq 0$ and $KRK' \geq 0$ for all $K$ and $X$. Then

$$\begin{aligned} g_{\bar\gamma}(X) &= \phi(K_X, X) = \\ &= (1-\bar\gamma)(AXA'+Q) + \bar\gamma(F_X X F_X' + K_X R K_X' + Q) \\ &\geq (1-\bar\gamma)(AXA'+Q) + \bar\gamma Q = (1-\bar\gamma)AXA' + Q. \end{aligned}$$

(g)First observe that $\bar X \geq g_\lambda(\bar X) \geq 0$. Thus, to prove that $\bar X > 0$, we only need to establish that $\bar X$ is nonsingular. Suppose $0 \neq v \in \mathcal{N}(\bar X)$, i.e. $\bar X v = 0$. Then

$$\begin{aligned} 0 &= v'\bar X v \geq v' g_\lambda(\bar X) v \\ &= (1-\lambda) v'(A\bar X A' + Q)v + \lambda v'(F\bar X F' + Q)v \end{aligned}$$

Positive semi-definiteness of $\bar X$ and $Q$ implies that all the terms in the sum must be zero for the inequality to hold. Consequently we have

$$v'A\bar X A' v = 0 \Longrightarrow \bar X A' v = 0 \Longrightarrow A'v \in \mathcal{N}(\bar X)$$

and

$$v'Qv = 0 \Longrightarrow Qv = 0$$

As a result, the null space $\mathcal{N}(\bar X)$ is $A'$-invariant. Therefore, $\mathcal{N}(\bar X)$ contains an eigenvector of $A'$, i.e. there exists $u \neq 0$ such that $\bar X u = 0$ and $A'u = \sigma u$. As before, we conclude that $Qu$=0. This implies (using the Popov-Belevich-Hautus(PBH) test) that the pair $(A, Q^{1/2})$ is not controllable, contradicting the hypothesis. Thus, $\mathcal{N}(\bar X)$ is empty, proving that $\bar X > 0$.

(h) Using fact (f) and linearity of expectation we have

$$\mathbb{E}[g_{\bar\gamma}(X)] \geq \mathbb{E}[(1-\bar\gamma)AXA' + Q] = (1-\bar\gamma)A\mathbb{E}[X]A' + Q.$$

Fact (e) implies that the operator $g_{\bar\gamma}()$ is concave, therefore by Jensen's Inequality we have:

$$\mathbb{E}[g_{\bar\gamma}(X)] \leq g_{\bar\gamma}(\mathbb{E}[X]).$$

□

**Lemma 6.2.** *Let $X_{t+1} = h(X_t)$ and $Y_{t+1} = h(Y_t)$. If $h(X)$ is a monotonically increasing function then:*

$$\begin{array}{lll} X_1 \geq X_0 & \Rightarrow & X_{t+1} \geq X_t, \quad \forall t \geq 0 \\ X_1 \leq X_0 & \Rightarrow & X_{t+1} \leq X_t, \quad \forall t \geq 0 \\ X_0 \leq Y_0 & \Rightarrow & X_t \leq Y_t, \qquad \forall t \geq 0 \end{array}$$

*Proof.* This lemma can be readily proved by induction. It is true for $t = 0$, since $X_1 \geq X_0$ by definition. Now assume that $X_{t+1} \geq X_t$, then $X_{t+2} = h(X_{t+1}) \geq h(X_t) = X_{t+1}$ because of monotonicity of $h(\cdot)$. The proof for the other two cases is analogous. □

It is important to note that while in the scalar case $X \in \mathbb{R}$ either $h(X) \leq X$ or $h(X) \geq X$; in the matrix case $X \in \mathbb{R}^{n\times n}$, it is not generally true that either $h(X) \geq X$ or $h(X) \leq X$. This is the source of the major technical difficulty for the proof of convergence of sequences in higher dimensions. In this case convergence of a sequence $\{X_t\}_0^\infty$ is obtained by finding two other sequences, $\{Y_t\}_0^\infty, \{Z_t\}_0^\infty$ that bound $X_t$, i.e., $Y_t \leq X_t \leq Z_t, \forall t$, and then by showing that these two sequences converge to the same point.

The next two Lemmas show that when the MARE has a solution $\bar{P}$, this solution is also stable, i.e., every sequence based on the difference Riccati equation $P_{t+1} = g_{\bar{\gamma}}(P_t)$ converges to $\bar{P}$ for all initial positive semidefinite conditions $P_0 \geq 0$.

**Lemma 6.3.** *Define the linear operator*

$$\mathcal{L}(Y) = (1-\bar{\gamma})(AYA') + \bar{\gamma}(FYF')$$

*Suppose there exists* $\overline{Y} > 0$ *such that* $\overline{Y} > \mathcal{L}(\overline{Y})$.

1. *For all* $W \geq 0$,
$$\lim_{k\to\infty} \mathcal{L}^k(W) = 0$$

2. *Let* $U \geq 0$ *and consider the linear system*
$$Y_{k+1} = \mathcal{L}(Y_k) + U \quad \textit{initialized at} \quad Y_0.$$
*Then, the sequence* $Y_k$ *is bounded.*

*Proof.* (a) First observe that $0 \leq \mathcal{L}(Y)$ for all $0 \leq Y$. Also, $X \leq Y$ implies $\mathcal{L}(X) \leq \mathcal{L}(Y)$. Choose $0 \leq r < 1$ such that $\mathcal{L}(\overline{Y}) < r\overline{Y}$. Choose $0 \leq m$ such that $W \leq m\overline{Y}$. Then,

$$0 \leq \mathcal{L}^k(W) \leq m\mathcal{L}^k(\overline{Y}) < mr^k\overline{Y}.$$

The assertion follows when we take the limit $r \to \infty$, on noticing that $0 \leq r < 1$.

(b) The solution of the linear iteration is

$$\begin{aligned} Y_k &= \mathcal{L}^k(Y_0) + \sum_{t=0}^{k-1} \mathcal{L}^t(U) \\ &\leq \left(m_{Y_0} r^k + \sum_{t=0}^{k-1} m_U r^t\right)\overline{Y} \\ &\leq \left(m_{Y_0} r^k + \frac{m_U}{1-r}\right)\overline{Y} \\ &\leq \left(m_{Y_0} + \frac{m_U}{1-r}\right)\overline{Y}, \end{aligned}$$

proving the claim. □

**Lemma 6.4.** *Consider the operator* $\phi(K, X)$ *defined in Equation (6.28). Suppose there exists a matrix* $\overline{K}$ *and a positive definite matrix* $\overline{P}$ *such that*

$$\overline{P} > 0 \quad \textit{and} \quad \overline{P} > \phi(\overline{K}, \overline{P}).$$

*Then, for any $P_0$, the sequence $P_t = g^t_{\bar\gamma}(P_0)$ is bounded, i.e. there exists $M_{P_0} \geq 0$ dependent of $P_0$ such that*

$$P_t \leq M \quad \textit{for all} \quad t.$$

*Proof.* First define the matrices $\overline{F} = A + \overline{K}C$ and consider the linear operator

$$\mathcal{L}(Y) = (1-\bar\gamma)(AYA') + \bar\gamma(\overline{F}Y\overline{F}')$$

Observe that

$$\overline{P} > \phi(\overline{K}, \overline{P}) = \mathcal{L}(\overline{P}) + Q + \bar\gamma\overline{K}R\overline{K}' \geq \mathcal{L}(\overline{P}).$$

Thus, $\mathcal{L}$ meets the condition of Lemma 6.3. Finally, using fact (b) in Lemma 6.1 we have

$$P_{t+1} = g_{\bar\gamma}(P_t) \leq \phi(\overline{K}, P_t) = \mathcal{L}P_t + Q + \bar\gamma\overline{K}R\overline{K}' = \mathcal{L}(P_t) + U.$$

Since $U = \bar\gamma\overline{K}R\overline{K}' + Q \geq 0$, using Lemma 6.3, we conclude that the sequence $P_t$ is bounded. □

We are now ready to give proofs for Theorems 1-4.

### Proof of Theorem 6.1

(a) We first show that the modified Riccati difference equation initialized at $Q_0 = 0$ converges. Let $Q_k = g^k_{\bar\gamma}(0)$. Note that $0 = Q_0 \leq Q_1$. It follows from Lemma 6.1(c) that

$$Q_1 = g_{\bar\gamma}(Q_0) \leq g_{\bar\gamma}(Q_1) = Q_2.$$

A simple inductive argument establishes that

$$0 = Q_0 \leq Q_1 \leq Q_2 \leq \cdots \leq M_{Q_0}.$$

Here, we have used Lemma 6.4 to bound the trajectory. We now have a monotone non-decreasing sequence of matrices bounded above. It is a simple matter to show that the sequence converges, i.e.

$$\lim_{k\to\infty} Q_k = \overline{P}.$$

Also, we see that $\overline{P}$ is a fixed point of the modified Riccati iteration:

$$\overline{P} = g_{\bar\gamma}(\overline{P}),$$

which establishes that it is *a* positive semi-definite solution of the MARE.

Next, we show that the Riccati iteration initialized at $R_0 \geq \overline{P}$ also converges, and to the same limit $\overline{P}$. First define the matrices

$$\overline{K} = -A\overline{P}C'\left(C\overline{P}C' + R\right)^{-1}, \quad \overline{F} = A + \overline{K}C$$

and consider the linear operator

$$\hat{\mathcal{L}}(Y) = (1-\bar\gamma)(AYA') + \bar\gamma(\overline{F}Y\overline{F}').$$

Observe that

$$\overline{P} = g_{\bar{\gamma}}(\overline{P}) = \mathcal{L}(\overline{P}) + Q + \overline{K}R\overline{K}' > \hat{\mathcal{L}}(\overline{P}).$$

Thus, $\hat{\mathcal{L}}$ meets the condition of Lemma 6.3. Consequently, for all $Y \geq 0$,

$$\lim_{k\to\infty} \hat{\mathcal{L}}^k(Y) = 0.$$

Now suppose $R_0 \geq \overline{P}$. Then,

$$R_1 = g_{\bar{\gamma}}(R_0) \geq g_{\bar{\gamma}}(\overline{P}) = \overline{P}.$$

A simple inductive argument establishes that

$$R_k \geq \overline{P} \quad \text{for all } k.$$

Observe that

$$\begin{aligned} 0 \leq (R_{k+1} - \overline{P}) &= g_{\bar{\gamma}}(R_k) - g_{\bar{\gamma}}(\overline{P}) \\ &= \phi(K_{R_k}, R_k) - \phi(K_{\overline{P}}, \overline{P}) \\ &\leq \phi(K_{\overline{P}}, R_k) - \phi(K_{\overline{P}}, \overline{P}) \\ &= (1-\bar{\gamma})A(R_k - \overline{P})A' + \bar{\gamma}F_{\overline{P}}(R_k - \overline{P})F_{\overline{P}}' \\ &= \hat{\mathcal{L}}(R_k - \overline{P}). \end{aligned}$$

Then, $0 \leq \lim_{k\to\infty}(R_{k+1} - \overline{P}) \leq 0$, proving the claim.

We now establish that the Riccati iteration converges to $\overline{P}$ for all initial conditions $P_0 \geq 0$. Define $Q_0 = 0$ and $R_0 = P_0 + \overline{P}$. Consider three Riccati iterations, initialized at $Q_0, P_0$, and $R_0$. Note that

$$Q_0 \leq P_0 \leq R_0.$$

It then follows from Lemma 6.2 that

$$Q_k \leq P_k \leq R_k \quad \text{for all } k.$$

We have already established that the Riccati equations $P_k$ and $R_k$ converge to $\overline{P}$. As a result, we have

$$\overline{P} = \lim_{k\to\infty} P_k \leq \lim_{k\to\infty} Q_k \leq \lim_{k\to\infty} R_k = \overline{P},$$

proving the claim.

(b) Finally, we establish that the MARE has a unique positive semidefinite solution. To this end, consider $\hat{P} = g_{\bar{\gamma}}(\hat{P})$ and the Riccati iteration initialized at $P_0 = \hat{P}$. This yields the constant sequence

$$\hat{P}, \hat{P}, \cdots$$

However, we have shown that every Riccati iteration converges to $\overline{P}$. Thus $\overline{P} = \hat{P}$.

### Proof of Theorem 6.2

First we note that the two cases expressed by the theorem are indeed possible. If $\bar{\gamma} = 1$ the modified Riccati difference equation reduces to the standard Riccati difference equation, which is known to converge to a fixed point, under the theorem's hypotheses. Hence, the covariance matrix is always bounded in this case, for any initial condition $P_0 \geq 0$. If $\bar{\gamma} = 0$ then we reduce to open loop prediction, and if the matrix $A$ is unstable, then the covariance matrix diverges for some initial condition $P_0 \geq 0$. Next, we show the existence of a single point of transition between the two cases. Fix a $0 < \bar{\gamma}_1 \leq 1$ such that $\mathbb{E}_{\bar{\gamma}_1}[P_t]$ is bounded for any initial condition $P_0 \geq 0$. Then, for any $\bar{\gamma}_2 \geq \bar{\gamma}_1$ $\mathbb{E}_{\bar{\gamma}_2}[P_t]$ is also bounded for all $P_0 \geq 0$. In fact we have

$$\begin{aligned}
\mathbb{E}_{\bar{\gamma}_1}[P_{t+1}] &= \mathbb{E}_{\bar{\gamma}_1}[AP_tA' + Q - \gamma_{t+1}AP_tC'(CP_tC' + R)^{-1}CP_tA] \\
&= \mathbb{E}[AP_tA' + Q - \bar{\gamma}_1 AP_tC'(CP_tC' + R)^{-1}CP_tA] \\
&= \mathbb{E}[g_{\bar{\gamma}_1}(P_t)] \\
&\geq \mathbb{E}[g_{\bar{\gamma}_2}(P_t)] \\
&= \mathbb{E}_{\bar{\gamma}_2}[P_{t+1}],
\end{aligned}$$

where we exploited fact (d) of Lemma 6.1 to write the above inequality . We can now choose

$$\gamma_c = \{\inf \bar{\gamma}^* : \bar{\gamma} > \bar{\gamma}^* \Rightarrow \mathbb{E}_{\bar{\gamma}}[P_t] \text{is bounded, for all } P_0 \geq 0\},$$

completing the proof.

### Proof of Theorem 6.3

Define the Lyapunov operator $m(X) = \tilde{A}X\tilde{A}' + Q$ where $\tilde{A} = \sqrt{1-\bar{\gamma}}A$. If $(A, Q^{\frac{1}{2}})$ is controllable, also $(\tilde{A}, Q^{\frac{1}{2}})$ is controllable. Therefore, it is well known that $\hat{S} = m(\hat{S})$ has a unique strictly positive definite solution $\hat{S} > 0$ if and only if $\max_i |\sigma_i(\tilde{A})| < 1$, i.e. $\sqrt{1-\bar{\gamma}}\, \max_i |\sigma_i(A)| < 1$, from which follows $\gamma_m in = 1 - \frac{1}{\alpha^2}$. If $\max_i |\sigma_i(\tilde{A})| \geq 1$ it is also a well known fact that there is no positive semidefinite fixed point to the Lyapunov equation $\hat{S} = m(\hat{S})$, since $(\tilde{A}, Q^{\frac{1}{2}})$ is controllable.

Let us consider the difference equation $S_{t+1} = m(S_t)$, $S_0 = 0$. It is clear that $S_0 = 0 \leq Q = S_1$. Since the operator $m()$ is monotonic increasing, by Lemma 6.2 it follows that the sequence $\{S_t\}_0^\infty$ is monotonically increasing, i.e. $S_{t+1} \geq S_t$ for all $t$. If $\bar{\gamma} < \gamma_m in$ this sequence does not converge to a finite matrix $\bar{S}$, otherwise by continuity of the operator $m$ we would have $\bar{S} = m(\bar{S})$, which is not possible. Since it is easy to show that a monotonically increasing sequence $S_t$ that does not converge is also unbounded, then we have

$$\lim_{t\to\infty} S_t = \infty.$$

Let us consider now the mean covariance matrix $\mathbb{E}[P_t]$ initialized at

$\mathbb{E}[P_0] \geq 0$. Clearly $0 = S_0 \leq \mathbb{E}[P_0]$. Moreover it is also true that $S_t \leq \mathbb{E}[P_t]$ implies:

$$\begin{aligned} S_{t+1} &= (1-\bar{\gamma})AS_tA' + Q \\ &\leq (1-\bar{\gamma})A\mathbb{E}[P_t]A' + Q \\ &\leq \mathbb{E}[g_{\bar{\gamma}}(P_t)] \\ &= \mathbb{E}[P_{t+1}], \end{aligned}$$

where we used fact (h) from Lemma 6.1. By induction, it is easy to show that

$$S_t \leq \mathbb{E}[P_t] \;\; \forall t, \; \forall \mathbb{E}[P_0] \geq 0 \Longrightarrow \lim_{t\to\infty} \mathbb{E}[P_t] \geq \lim_{t\to\infty} S_t = \infty.$$

This implies that for any initial condition $\mathbb{E}[P_t]$ is unbounded for any $\bar{\gamma} < \gamma_m in$, therefore $\gamma_m in \leq \gamma_c$, which proves the first part of the Theorem.

Now consider the sequence $V_{t+1} = g_{\bar{\gamma}}(V_t)$, $V_0 = \mathbb{E}[P_0] \geq 0$. Clearly $\mathbb{E}[P_t] \leq V_t$ implies:

$$\begin{aligned} \mathbb{E}[P_{t+1}] &= \mathbb{E}[g_{\bar{\gamma}}(P_t)] \\ &\leq g_{\bar{\gamma}}(\mathbb{E}[P_t]) \\ &\leq [g_{\bar{\gamma}}(V_t)] \\ &= V_{t+1}, \end{aligned}$$

where we used facts (c) and (h) from Lemma 6.1. Then a simple induction argument shows that $V_t \geq \mathbb{E}[P_t]$ for all $t$. Let us consider the case $\bar{\gamma} > \gamma_m ax$, therefore there exists $\hat{X}$ such that $\hat{X} \geq g_{\bar{\gamma}}(\hat{X})$. By Lemma 6.1(g) $\bar{X} > 0$, therefore all hypotheses of Lemma 6.3 are satisfied, which implies that

$$\mathbb{E}[P_t] \leq V_t \leq M_{V_0} \;\; \forall t.$$

This shows that $\gamma_c \leq \gamma_m ax$ and concludes the proof of the Theorem.

### Proof of Theorem 6.4

Let us consider the sequences $S_{t+1} = (1-\bar{\gamma})AS_tA' + Q$, $S_0 = 0$ and $V_{t+1} = g_{\bar{\gamma}}(V_t)$, $V_0 = \mathbb{E}[P_0] \geq 0$. Using the same induction arguments in Theorem 6.3 it is easy to show that

$$S_t \leq \mathbb{E}[P_t] \leq V_t \;\; \forall t.$$

From Theorem 6.1 it also follows that $\lim_{t\to\infty} V_t = \bar{V}$, where $\bar{V} = g_{\bar{\gamma}}(V)$. As shown before the sequence $S_t$ is monotonically increasing. Also it is bounded since $S_t \leq V_t \leq M$. Therefore $\lim_{t\to\infty} S_t = \bar{S}$, and by continuity $\bar{S} = (1-\bar{\gamma})A\bar{S}A' + Q$, which is a Lyapunov equation. Since $\sqrt{1-\bar{\gamma}}A$ is stable and $(A, Q^{\frac{1}{2}})$ is controllable, then the solution of the Lyapunov equation is strictly positive definite, i.e. $\bar{S} > 0$. Adding all the results together we get

$$0 < \bar{S} = \lim_{t\to\infty} S_t \leq \lim_{t\to\infty} \mathbb{E}[P_t] \leq \lim_{t\to\infty} V_t = \bar{V},$$

which concludes the proof.

The text below is from pbctrl.tex, which used to be in a separate chapter. Integrate as appropriate **Bruno**

## 6.8 Packet-Based Control

Outline:

- Problem setup
- Communication protocols and information patterns
- TCP-Based Control
- UDP-Based Control
- Receding-Horizon networked control and actuation buffers
- Generalization to multi-channel
- Nonlinear estensions

## 6.9 Introduction

Today, an increasing number of applications demand remote control of plants over unreliable networks. The recent development of sensor web technology [**?**] enables the development of wireless sensor networks that can be immediately used for estimation and control of dynamical systems. In these systems, issues of communication delay, data loss, and time-synchronization play critical roles. Communication and control become very tightly coupled and these two issues cannot be addressed independent of one another during the design and analysis stages of development. Consider, for example, the problem of navigating a fleet of vehicles using observations from a sensor web. Wireless nodes collect their sensor measurements and send them to a computing unit. This unit, in turn, generates estimates of the state of each vehicle and computes inputs that are then delivered, using the same wireless channel, to the actuators onboard the vehicles. Due to the unreliability of the wireless channel, both observations underlying the estimate and control packets sent to each vehicle can be lost or delayed while travelling across the network. It needs to be determined exactly how much data loss a control loop can tolerate to reliably perform the navigation task. What's more, specific communication protocols need to be designed to satisfy this constraint. The goal of this chapter is to provide the first steps in meeting these requirements by examining the basic system-theoretic implications of using unreliable networks for control. This requires a generalization of

**Figure 6.8: Overview of the system.** We study the statistical convergence properties of the expected state covariance of the discrete time LQG control system, where both the observation and the control signal, transmitted over an unreliable communication channel, can be lost at each time step with probability $1 - \bar{\gamma}$ and $1 - \bar{\nu}$ respectively.

classical control techniques that explicitly takes into account the stochastic nature of the communication channel.

Packet networks communication channels typically use one of two kinds of protocols: Transmission Control (TCP) or User Datagram (UDP). In the first case there is acknowledgement of received packets, while in the second case no confirmation feedback is provided on the communication link. In this chapter, we study the effect of data losses due to the unreliability of the network links under these two protocols. We generalize the Linear Quadratic Gaussian (LQG) optimal control problem to these problems by modeling the arrival of both observations and control packets as random processes whose parameters are related to the characteristics of the communication channel. Accordingly, two independent Bernoulli processes are considered, with parameters $\overline{\gamma}$ and $\overline{\nu}$, that govern packet losses between the sensors and the estimation-control unit, and between the latter and the actuation points (see Figure 6.8).

In our analysis, the distinction between the two classes of protocols resides exclusively in the availability of packet acknowledgements. Adopting the framework proposed by Imer *et al.* [**?**], we will refer therefore to TCP-like protocols if packet acknowledgements are available and to UDP-like protocols otherwise. We summarize our contributions as follows. For the TCP-like case the classic separation principle holds, and consequently the controller and estimator can be designed independently. Moreover, the optimal controller is a linear function of the state. In sharp contrast, for the UDP-like case, a counter-example demonstrates that the optimal controller is in general non-linear. In the special case when the state is fully observable and the observation noise is zero, the optimal controller is indeed linear. We explicitly note that a similar, but slightly less general special case was previously analyzed in [?], where both observation and process noise are assumed to be zero and the input coefficient matrix to be invertible.

Our final set of results relate to convergence in the infinite horizon. Here, our previous results on estimation with missing observation packets [?] [?] are extended to the control case. We show the existence of a critical domain of values for the parameters of the Bernoulli arrival processes, $\overline{\nu}$ and $\overline{\gamma}$, outside which a transition to instability occurs and the optimal controller fails to stabilize the system. In particular, we show that under TCP-like protocols the critical arrival probabilities for the control and observation channel are independent of each other. This is another consequence of the fact that the separation principle holds for these protocols. In contrast, under UDP-like protocols the critical arrival probabilities for the control and observation channels are coupled. Here, the stability domain and performance of the optimal controller degrade considerably as compared with TCP-like protocols as shown in Figure 6.9.

Finally, we wish to mention some closely related research. The study of the stability of dynamical systems where components are connected asynchronously via communication channels has received considerable attention in the past few years and our contribution can be put in the context of the previous literature. In [?] and [?], the authors proposed to place an estimator, i.e. a Kalman filter, at the sensor side of the link without assuming any statistical model for the data loss process. In [?], Smith *et al.* considered a suboptimal but computationally efficient estimator that can be applied when the arrival process is modeled as a Markov chain, which is more general than a Bernoulli process. Other work includes Nilsson *et al.* [?][?] who present the LQG optimal regulator with bounded delays between sensors and controller, and between the controller and the actuator. In this work, bounds for the critical probability values are not provided. Additionally, there is no analytical solution for the optimal controller. The case where dropped measurements are replaced by zeros is considered by Hadjicostis and Touri [?], but only in the scalar case. Other approaches include using the last received sample for control [?], or designing a dropout compensator [?], which combines estimation and control in a single process. However, the former approach does not consider optimal control and the latter is limited to scalar systems. Yu *et al.* [?] studied the design of an

**Figure 6.9:** Region of stability for UDP-like and TCP-like optimal control relative to measurement packet arrival probability $\gamma$, and the control packet arrival probability $\nu$.

optimal controller with a single control channel and deterministic dropout rates. Seiler *et al.* [**?**] considered Bernoulli packet losses only between the plant and the controller, and posed the controller design as an $H_\infty$ optimization problem. Other authors [**?**] [**?**] [**?**] [**?**] model networked control systems with missing packets as Markovian jump linear systems (MJLSs), however this approach gives suboptimal controllers since the estimators are stationary. Finally, Elia [**?**][**?**] proposed to model the plant and the controller as deterministic time invariant discrete-time systems connected to zero-mean stochastic structured uncertainty. The variance of the stochastic perturbation is a function of the Bernoulli parameters, and the controller design is posed as an optimization problem to maximize mean-square stability of the closed loop system. This approach allows analysis of Multiple Input Multiple Output (MIMO) systems with many different controller and receiver compensation schemes [**?**], however, it does not include process and observation noise and the controller is restricted to be time-invariant, hence suboptimal. There is also extensive literature, inspired by Shannon's results on the maximum bit-rate that an imperfect channel can reliably carry. Here the goal is to determine the minimum bit-rate that is needed to stabilize a system through feedback [?] [?] [?] [?] [?] [?] [?] [?] [?] [?]. This approach is somewhat different from ours since in a packet-based communication network, such as ATMs, Ethernet and Bluetooth, bits are grouped into packets and are considered as a single entity. Nonetheless there are several similarities that are not yet fully explored.

This work considers the alternative approach where the external compensator feeding the controller is the optimal time varying Kalman gain. Moreover, this approach considers the general Multiple Input Multiple Output (MIMO) case, and gives some necessary and sufficient conditions for closed loop stability. The work of [?] is most closely related to ours. However, we consider the more general case when the matrix $C$ is not the identity and there is noise in the observation and in the process. In addition, we also give stronger necessary and sufficient conditions for existence of solution for the infinite horizon LQG.

The remainder of this chapter is organized as follows. Section 2 provides a mathematical formulation of the problems we consider. Section 3 offers some preliminary results. Section 4 illustrates the TCP-like case, while the UDP-like case is studied in section 5. Finally, conclusions and directions for future work are offered in section 6.

## 6.10 Problem formulation

Consider the following linear stochastic system with intermittent observation and control packets:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{6.29}$$

$$u_k^a = \nu_k u_k^c \tag{6.30}$$

$$y_k = \gamma_k C x_k + v_k, \tag{6.31}$$

where $u_k^a$ is the control input to the actuator, $u_k^c$ is the desired control input computed by the controller, $(x_0, w_k, v_k)$ are Gaussian, uncorrelated, white, with mean $(\bar{x}_0, 0, 0)$ and covariance $(P_0, Q, R)$ respectively, and $(\gamma_k, \nu_k)$ are i.i.d. Bernoulli random variables with $P(\gamma_k = 1) = \bar{\gamma}$ and $P(\nu_k = 1) = \bar{\nu}$. The stochastic variable $\nu_k$ models the loss packets between the controller and the actuator: if the packet is correctly delivered then $u_k^a = u_k^c$, otherwise if it is lost then the actuator does nothing, i.e. $u_k^a = 0$. This compensation scheme is summarized by Equation (6.30). This modeling choice is not unique: for example if the control packet $u_k^c$ is lost, then the actuator could use the previous control value, i.e. $u_k^a = u_{k-1}^a$. However, the latter control compensation is slightly more involved to analyze and it is left as future work. The stochastic variable $\gamma_k$ models the packet loss between the sensor and the controller: if the packet is delivered then $y_k = Cx_k + v_k$, otherwise if it is lost then the controller reads pure noise, i.e. $y_k = v_k$. This observation model is summarized by Equation (6.31). A different observation formalism was proposed in [**?**], where the missing observation was modeled as an observation for which the measurement noise had infinite covariance. It is possible to show that both models are equivalent, but the one considered here has the advantage to give rise to simpler analysis. This arises from the fact that when no packet is delivered, then the optimal estimator does not use the observation $y_k$ at all, therefore its value is irrelevant.

Let us define the following information sets:

$$\mathcal{I}_k = \begin{cases} \mathcal{F}_k & \triangleq & \{\mathbf{y}^k, \boldsymbol{\gamma}^k, \boldsymbol{\nu}^{k-1}\}, & \text{TCP-like} \\ \mathcal{G}_k & \triangleq & \{\mathbf{y}^k, \boldsymbol{\gamma}^k\}, & \text{UDP-like} \end{cases} \tag{6.32}$$

where $\mathbf{y}^k = (y_k, y_{k-1}, \ldots, y_1)$, $\boldsymbol{\gamma}^k = (\gamma_k, \gamma_{k-1}, \ldots, \gamma_1)$, and $\boldsymbol{\nu}^k = (\nu_k, \nu_{k-1}, \ldots, \nu_1)$.

Consider also the following cost function:

$$J_N(\mathbf{u}^{N-1}, \bar{x}_0, P_0) = \mathbb{E}\left[ x_N' W_N x_N + \sum_{k=0}^{N-1} (x_k' W_k x_k + \nu_k u_k' U_k u_k) \;\middle|\; \mathbf{u}^{N-1}, \bar{x}_0, P_0 \right] \tag{6.33}$$

where $\mathbf{u}^{N-1} = (u_{N-1}, u_{N-2}, \ldots, u_1)$. Note that we are weighting the input only if it is successfully received at the plant. In fact, if it is not received, the plant applies zero input and therefore there is no energy expenditure.

We now look for a control input sequence $\mathbf{u}^{*N-1}$ as a function of the admissible information set $\mathcal{I}_k$, i.e. $u_k = g_k(\mathcal{I}_k)$, that minimizes the functional defined in Equation (6.33), i.e.

$$J_N^*(\bar{x}_0, P_0) \triangleq \min_{\mathbf{u_k} = \mathbf{g}_k(\mathcal{I}_k)} J_N(\mathbf{u}^{N-1}, \bar{x}_0, P_0), \tag{6.34}$$

where $\mathcal{I}_k = \{\mathcal{F}_k, \mathcal{G}_k\}$ is one of the sets defined in Equation (6.32). The set $\mathcal{F}$ corresponds to the information provided under an acknowledgement-based communication protocols (TCP-like) in which successful or unsuccessful packet delivery at the receiver is acknowledged to the sender within the same sampling time period. The set $\mathcal{G}$ corresponds to the information available at the controller under communication protocols in which the sender receives no feedback about the delivery of the transmitted packet to the receiver (UDP-like). The UDP-like schemes are simpler to implement than the TCP-like schemes from a communication standpoint. Moreover UDP-like protocols includes broadcasting which you cannot do with TCP-like. However the price to pay is a less rich set of information. The goal of this chapter is to design optimal LQG controllers and to estimate their performance for each of these classes of protocols for a general discrete-time linear stochastic system.

## 6.11 Mathematical Preliminaries

Before proceeding, let us define the following variables:

$$\begin{aligned} \hat{x}_{k|k} &\overset{\Delta}{=} \mathbb{E}[x_k \mid \mathcal{I}_k], \\ e_{k|k} &\overset{\Delta}{=} x_k - \hat{x}_{k|k}, \\ P_{k|k} &\overset{\Delta}{=} \mathbb{E}[e_{k|k}e'_{k|k} \mid \mathcal{I}_k]. \end{aligned} \tag{6.35}$$

Derivations below will make use of the following facts:

**Lemma 6.5.** *The following facts are true [**?**]:*

1. $\mathbb{E}\left[(x_k - \hat{x}_k)\hat{x}'_k \mid \mathcal{I}_k\right] = \mathbb{E}\left[e_{k|k}\hat{x}'_k \mid \mathcal{I}_k\right] = 0$

2. $\mathbb{E}\left[x'_k S x_k \mid \mathcal{I}_k\right] = \hat{x}'_k S \hat{x}_k + \mathrm{trace}\left(SP_{k|k}\right) \quad \forall S \geq 0$

3. $\mathbb{E}\left[\mathbb{E}[\; g(x_{k+1}) \; |\mathcal{I}_{k+1}] \mid \mathcal{I}_k\right] = \mathbb{E}\left[g(x_{k+1}) \mid \mathcal{I}_k\right], \forall g(\cdot).$

*Proof.* (a) It follows directly from the definition. In fact:

$$\begin{aligned} \mathbb{E}\left[(x_k - \hat{x}_k)\hat{x}'_k \mid \mathcal{I}_k\right] &= \mathbb{E}\left[x_k\hat{x}'_k - \hat{x}_k\hat{x}'_k \mid \mathcal{I}_k\right] \\ &= \mathbb{E}\left[x_k \mid \mathcal{I}_k\right]\hat{x}'_k - \hat{x}_k\hat{x}'_k \\ &= 0 \end{aligned}$$

(b) Using standard algebraic operations and the previous fact we have:

$$\begin{aligned} \mathbb{E}\left[x'_k S x_k \; |\mathcal{I}_k\right] &= \mathbb{E}\left[(x_k - \hat{x}_k + \hat{x}_k)' S (x_k - \hat{x}_k + \hat{x}_k) \; |\mathcal{I}_k\right] \\ &= \hat{x}'_k S \hat{x}_k + \mathbb{E}\left[(x_k - \hat{x}_k)' S (x_k - \hat{x}_k)\right] + 2\mathbb{E}\left[\hat{x}'_k S (x_k - \hat{x}_k) \mid \mathcal{I}_k\right] \\ &= \hat{x}'_k S \hat{x}_k + 2\mathrm{trace}(S\mathbb{E}[(x_k - \hat{x}_k)\hat{x}'_k \; |\mathcal{I}_k]) + \mathrm{trace}(S\mathbb{E}[(x_k - \hat{x}_k)(x_k - \hat{x}_k)' \; |\mathcal{I}_k]) \\ &= \hat{x}'_k S \hat{x}_k + \mathrm{trace}\{SP_{k|k}\} \end{aligned}$$

(c) Let $g()$ any measurable function, $(X, Y, Z)$ be any random vectors, and $p$ their probability distribution, then

$$\begin{aligned} \mathbb{E}_{Y,Z}\left[g(X,Y,Z) \mid X\right] &= \int_Z \int_Y g(X,Y,Z) p(Y,Z|X) dY dZ \\ &= \int_Z \int_Y g(X,Y,Z) p(Y|Z,X) p(Z|X) dY dZ \\ &= \int_Z \left[\int_Y g(X,Y,Z) p(Y|Z,X) dY\right] p(Z|X) dZ \\ &= \mathbb{E}_Z\left[\; \mathbb{E}_Y\left[g(X,Y,Z) \mid Z, X\right] \; \mid X\right] \end{aligned}$$

where we used the Bayes' Rule. Since by hypothesis $\mathcal{I}_k \subseteq \mathcal{I}_{k+1}$, then fact (c) follows from the above equality by substituting $\mathcal{I}_k = X$ and $\mathcal{I}_{k+1} = (X, Z)$. □

We now make the following computations that will be useful when deriving the equation for the optimal LQG controller.

$$\begin{aligned}
\mathbb{E}[x'_{k+1}Sx_{k+1} \mid \mathcal{I}_k] &= \mathbb{E}[(Ax_k+\nu_k Bu_k+w_k)'S(Ax_k+\nu_k Bu_k+w_k) \mid \mathcal{I}_k]\\
&= \mathbb{E}[x'_kA'SAx_k+\nu_k^2u'_kB'SBu_k+w'_kSw_k+2\nu_k u'_kB'SAx_k+2(Ax_k+\nu_k Bu_k)w_k|\mathcal{I}_k]\\
&= \mathbb{E}[x'_kA'SAx_k|\mathcal{F}_k]+\bar{\nu}u'_kB'SBu_k+2\bar{\nu}u'_kB'SA\,\mathbb{E}[x_k|\mathcal{I}_k]+\mathrm{trace}(S\mathbb{E}[w_kw'_k \mid \mathcal{F}_k])\\
&= \mathbb{E}[x'_kA'SAx_k \mid \mathcal{I}_k]+\bar{\nu}u'_kB'SBu_k+\ 2\bar{\nu}u'_kB'SA\,\hat{x}_{k|k}+\mathrm{trace}(SQ) \qquad (6.36)
\end{aligned}$$

where both the independence of $\nu_k, w_k, x_k$, and the zero-mean property of $w_k$ are exploited. The previous expectation holds true for both the information sets, i.e. $\mathcal{I}_k = \mathcal{F}_k$ or $\mathcal{I}_k = \mathcal{G}_k$. Also

$$\begin{aligned}
\mathbb{E}[e'_{k|k}Te_{k|k} \mid \mathcal{I}_k] &= \mathrm{trace}(T\mathbb{E}[e_{k|k}e'_{k|k} \mid \mathcal{I}_k])\\
&= \mathrm{trace}(TP_{k|k}), \quad \forall T \geq 0.
\end{aligned}$$

## 6.12 LQG control for TCP-like protocols

First, equations for the optimal estimator are derived. They will be needed to solve the LQG controller design problem, as it will be shown later.

### Estimator Design

Equations for optimal estimator are derived using similar arguments used for the standard Kalman filtering equations. The innovation step is given by:

$$\begin{aligned}
\hat{x}_{k+1|k} &\overset{\Delta}{=} \mathbb{E}[x_{k+1}|\nu_k,\mathcal{F}_k] = \mathbb{E}[Ax_k+\nu_kBu_k+w_k|\nu_k,\mathcal{F}_k]\\
&= A\mathbb{E}[x_k|\mathcal{F}_k]+\nu_kBu_k = A\hat{x}_{k|k}+\nu_kBu_k \qquad (6.37)\\
e_{k+1|k} &\overset{\Delta}{=} x_{k+1}-\hat{x}_{k+1|k}\\
&= Ax_k+\nu_kBu_k+w_k-(A\hat{x}+\nu_kBu_k)\\
&= Ae_{k|k}+w_k \qquad (6.38)\\
P_{k+1|k} &\overset{\Delta}{=} \mathbb{E}[e_{k+1|k}e'_{k+1|k}\ |\nu_k,\mathcal{F}_k]\\
&= \mathbb{E}\left[\left(Ae_{k|k}+w_k\right)\left(Ae_{k|k}+w_k\right)'\ |\nu_k,\mathcal{F}_k\right]\\
&= A\mathbb{E}[e_{k|k}e'_{k|k}|\mathcal{F}_k]A'+\mathbb{E}[w_kw'_k]\\
&= AP_{k|k}A'+Q, \qquad (6.39)
\end{aligned}$$

where the independence of $w_k$ and $\mathcal{F}_k$, and the requirement that $u_k$ is a deterministic function of $\mathcal{F}_k$, are used. Since $y_{k+1}, \gamma_{k+1}$, $w_k$ and $\mathcal{F}_k$ are independent, the correction step is given by:

$$
\begin{aligned}
\hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + \gamma_{k+1} K_{k+1} (y_{k+1} - C\hat{x}_{k+1|k}) && (6.40)\\
e_{k+1|k+1} &\overset{\Delta}{=} x_{k+1} - \hat{x}_{k+1|k+1} \\
&= x_{k+1} - \big(\hat{x}_{k+1|k} + \gamma_{k+1} K_{k+1} (C x_{k+1} + v_{k+1} - C\hat{x}_{k+1|k})\big) \\
&= (I - \gamma_{k+1} K_{k+1} C) e_{k+1|k} - \gamma_{k+1} K_{k+1} v_{k+1} && (6.41)\\
P_{k+1|k+1} &= P_{k+1|k} - \gamma_{k+1} K_{k+1} C P_{k+1|k} \\
&= P_{k+1|k} - \gamma_{k+1} P_{k+1|k} C' (C P_{k+1|k} C' + R)^{-1} C P_{k+1|k} && (6.42)\\
K_{k+1} &\overset{\Delta}{=} P_{k+1|k} C' (C P_{k+1|k} C' + R)^{-1}, && (6.43)
\end{aligned}
$$

where we simply applied the standard derivation for the time varying Kalman filter using the following time varying system matrices: $A_k = A$, $C_k = \gamma_k C$, and $\mathrm{Cov}(v_k) = R$.

### Controller design

Derivation of the optimal feedback control law and the corresponding value for the objective function will follow the dynamic programming approach based on the cost-to-go iterative procedure.

Define the optimal value function $V_k(x_k)$ as follows:

$$
\begin{aligned}
V_N(x_N) &\overset{\Delta}{=} \mathbb{E}[x_N' W_N x_N \mid \mathcal{F}_N] \\
V_k(x_k) &\overset{\Delta}{=} \min_{u_k} \mathbb{E}[x_k' W_k x_k + \nu_k u_k' U_k u_k + V_{k+1}(x_{k+1}) | \mathcal{F}_k].
\end{aligned} \qquad (6.44)
$$

where $k = N-1, \dots, 1$. Using dynamic programming theory [**?**], one can show that $J_N^* = V_0(x_0)$. Under TCP-like protocols the following lemma holds true:

**Lemma 6.6.** *The value function $V_k(x_k)$ defined in Equations (6.44) for the system dynamics of Equations (6.29)-(6.29) under TCP-like protocols can be written as:*

$$
V_k(x_k) = \mathbb{E}[\ x_k' S_k x_k \mid \mathcal{F}_k] + c_k, \quad k = N, \dots, 0 \qquad (6.45)
$$

*where the matrix $S_k$ and the scalar $c_k$ can be computed recursively as follows:*

$$
\begin{aligned}
S_k &= A' S_{k+1} A + W_k - \bar{\nu} A' S_{k+1} B (B' S_{k+1} B + U_k)^{-1} B' S_{k+1} A && (6.46)\\
c_k &= \mathrm{trace}\big((A' S_{k+1} A + W_k - S_k) P_{k|k}\big) + \mathrm{trace}(S_{k+1} Q) + \mathbb{E}[c_{k+1} \mid \mathcal{F}_k] && (6.47)
\end{aligned}
$$

*with initial values $S_N = W_N$ and $c_N = 0$. Moreover the optimal control input is given by:*

$$
u_k = -(B' S_{k+1} B + U_k)^{-1} B' S_{k+1} A\, \hat{x}_{k|k} = L_k\, \hat{x}_{k|k}. \qquad (6.48)
$$

*Proof.* The proof follows an induction argument. The claim is certainly true for $k = N$ with the choice of parameters $S_N = W_N$ and $c_N = 0$. Suppose now that the claim is true for $k+1$, i.e. $V_{k+1}(x_{k+1}) = \mathbb{E}[\, x'_{k+1}S_{k+1}x_{k+1} \mid \mathcal{F}_{k+1}] + c_{k+1}$. The value function at time step $k$ is the following:

$$\begin{aligned}
V_k(x_k) &= \min_{u_k} \mathbb{E}[x'_kW_kx_k + \nu_k u'_kU_ku_k + V_{k+1}(x_{k+1}) \mid \mathcal{F}_k] \\
&= \min_{u_k} \mathbb{E}[x'_kW_kx_k + \nu_k u'_kU_ku_k + \mathbb{E}[x'_{k+1}S_{k+1}x_{k+1} + c_{k+1} \,|\, \mathcal{F}_{k+1}] \;|\mathcal{F}_k] \\
&= \min_{u_k} \mathbb{E}[x'_kW_kx_k + \nu_k u'_kU_ku_k + x'_{k+1}S_{k+1}x_{k+1} + c_{k+1}|\mathcal{F}_k] \qquad (6.49) \\
&= \mathbb{E}[x'_kW_kx_k + x'_kA'S_{k+1}Ax_k \mid \mathcal{F}_k] + \mathrm{trace}(S_{k+1}Q) + \mathbb{E}[c_{k+1} \mid \mathcal{F}_k] + \\
&\quad + \bar{\nu} \min_{u_k} \Big( u'_k(U_k + B'S_{k+1}B)u_k + 2u'_kB'S_{k+1}A\,\hat{x}_{k|k} \Big)
\end{aligned}$$

where we used Lemma 1(c) to get the third equality, and Equation (6.36) to obtain the last equality. The value function is a quadratic function of the input, therefore the minimizer can be simply obtained by solving $\frac{\partial V_k}{\partial u_k} = 0$, which gives Equation (6.48). The optimal feedback is thus a simple linear function of the estimated state. If we substitute the minimizer back into Equation (6.49) we get:

$$\begin{aligned}
V_k(x_k) &= \mathbb{E}[x'_kW_kx_k + x'_kA'S_{k+1}Ax_k \mid \mathcal{I}_k] + \mathrm{trace}(S_{k+1}Q) + \mathbb{E}[c_{k+1} \mid \mathcal{I}_k] - \\
&\quad - \bar{\nu}\hat{x}'_{k|k}A'S_{k+1}B(U_k + B'S_{k+1}B)^{-1}B'S_{k+1}A\hat{x}_{k|k} \qquad (6.50) \\
&= \mathbb{E}[x'_kW_kx_k + x'_kA'S_{k+1}Ax_k - \bar{\nu}x'_kA'S_{k+1}B(U_k + B'S_{k+1}B)^{-1}B'S_{k+1}Ax_k \mid \mathcal{I}_k] + \\
&\quad + \mathrm{trace}(S_{k+1}Q) + \mathbb{E}[c_{k+1} \mid \mathcal{I}_k] + \bar{\nu}\,\mathrm{trace}(A'S_{k+1}B(U_k + B'S_{k+1}B)^{-1}B'S_{k+1}\,P_{k|k})
\end{aligned}$$

where we used Lemma 1(b). Therefore, the claim given by Equation (6.45) is satisfied also for time step $k$ for all $x_k$ if and only if the Equations (6.46) and (6.47) are satisfied. □

Since $J^*_N(\bar{x}_0, P_0) = V_0(x_0)$, from the lemma it follows that the cost function for the optimal LQG using TCP-like protocols is given by:

$$J^*_N = \bar{x}'_0S_0\bar{x}_0 + \mathrm{trace}(S_0P_0) + \sum_{k=0}^{N-1} \mathrm{trace}((A'S_{k+1}A + W_k - S_k)\mathbb{E}_\gamma[P_{k|k}] + S_{k+1}Q), \qquad (6.51)$$

where we used the fact $\mathbb{E}[x'_0S_0x_0] = \bar{x}'_0S_0\bar{x}_0 + \mathrm{trace}(S_0P_0)$, and $\mathbb{E}_\gamma[\cdot]$ explicitly indicates that the expectation is calculated with respect to the arrival sequence $\{\gamma_k\}$.

It is important to remark that the error covariance matrices $\{P_{k|k}\}_{k=0}^N$ are stochastic since they depend on the sequence $\{\gamma_k\}$. Moreover, since the matrix $P_{k+1|k+1}$ is a nonlinear function of the previous time step matrix covariance $P_{k|k}$, as can be observed from Equations (6.39) and (6.43), the exact expected value of these matrices, $\mathbb{E}_\gamma[P_{k|k}]$, cannot be computed analytically, as shown in [**?**]. However, they can be bounded by computable deterministic quantities, as shown in [**?**] from which we can derive the following lemma:

**Lemma 6.7** ([**?**]). *The expected error covariance matrix $\mathbb{E}_\gamma[P_{k|k}]$ satisfies the following bounds:*

$$\widetilde{P}_{k|k} \leq \mathbb{E}_\gamma[P_{k|k}] \leq \widehat{P}_{k|k} \qquad \forall k \geq 0, \tag{6.52}$$

*where the matrices $\widehat{P}_{k|k}$ and $\widetilde{P}_{k|k}$ can be computed as follows:*

$$\widehat{P}_{k+1|k} = A\widehat{P}_{k|k-1}A' + Q - \bar{\gamma}A\widehat{P}_{k|k-1}C'(C\widehat{P}_{k|k-1}C' + R)^{-1}C\widehat{P}_{k|k-1}A' \tag{6.53}$$

$$\widehat{P}_{k|k} = \widehat{P}_{k|k-1} - \bar{\gamma}\widehat{P}_{k|k-1}C'(C\widehat{P}_{k|k-1}C' + R)^{-1}C\widehat{P}_{k|k-1} \tag{6.54}$$

$$\widetilde{P}_{k+1|k} = (1-\bar{\gamma})A\widetilde{P}_{k|k-1}A' + Q \tag{6.55}$$

$$\widetilde{P}_{k|k} = (1-\bar{\gamma})\widetilde{P}_{k|k-1} \tag{6.56}$$

*where the initial conditions are $\widehat{P}_{0|0} = \widetilde{P}_{0|0} = P_0$.*

*Proof.* The proof is based on the observation that the matrices $P_{k+1|k}$ and $P_{k|k}$ are concave and monotonic functions of $P_{k|k-1}$. The proof is given in [**?**] and is thus omitted. □

From this lemma it follows that also the minimum achievable cost $J_N^*$, given by Equation (6.51), cannot be computed analytically, but can bounded as follows:

$$J_N^{min} \leq J_N^* \leq J_N^{max} \tag{6.57}$$

$$J_N^{max} = \bar{x}_0' S_0 \bar{x}_0 + \text{trace}(S_0 P_0) + \sum_{k=0}^{N-1} \text{trace}(S_{k+1}Q)) + \sum_{k=0}^{N-1} \text{trace}\Big((A'S_{k+1}A + W_k - S_k)\widehat{P}_{k|k}\Big) \tag{6.58}$$

$$J_N^{min} = \bar{x}_0' S_0 \bar{x}_0 + \text{trace}(S_0 P_0) + \sum_{k=0}^{N-1} \text{trace}(S_{k+1}Q) + \sum_{k=0}^{N-1} \text{trace}\Big((A'S_{k+1}A + W_k - S_k)\widetilde{P}_{k|k}\Big) \tag{6.59}$$

### Finite and Infinite Horizon LQG control

The results derived in the previous sections can be summarized in the following theorem:

**Theorem 6.7.** *Consider the system (6.29)-(6.31) and consider the problem of minimizing the cost function (6.33) within the class of admissible policies $u_k = f(\mathcal{F}_k)$, where $\mathcal{F}_k$ is the information available under TCP-like schemes, given in Equation (6.32). Then:*

1. *The separation principle still holds for TCP-like communication, since the optimal estimator, given by Equations (6.37),(6.39),(6.40),(6.42) and (6.43), is independent of the control input $u_k$.*

2. *The optimal estimator gain $K_k$ is time-varying and stochastic since it depends on the past observation arrival sequence $\{\gamma_j\}_{j=1}^k$.*

3. *The optimal control input, given by Equations (6.48) and (6.46) with initial condition $S_N = W_N$, is a linear function of the estimated state $\hat{x}_{k|k}$, i.e. $u_k = L_k \hat{x}_{k|k}$, and is independent of the process sequences $\{\nu_k, \gamma_k\}$.*

*Proof.* The proof follows from the results given in the previous sections. □

The infinite horizon LQG can be obtained by taking the limit for $N \rightarrow +\infty$ of the previous equations. However, as explained above, the matrices $\{P_{k|k}\}$ depend nonlinearly on the specific realization of the observation sequence $\{\gamma_k\}$, therefore the expected error covariance matrices $\mathbb{E}_\gamma[P_{k|k}]$ and the minimal cost $J_N^*$ cannot be computed analytically and do not seem to have limit [**?**]. Differently from standard LQG optimal regulator [**?**], the estimator gain does not converge to a steady state value, but is strongly time-varying due to its dependence on the arrival process $\{\gamma_k\}$. Moreover, while the standard LQG optimal regulator always stabilizes the original system, in the case of observation and control packet losses, the stability can be lost if the arrival probabilities $\bar{\nu}, \bar{\gamma}$ are below a certain threshold. This observation come from the study of existence of solution for a Modified Riccati Algebraic Equation (MARE), $S = \Pi(S, A, B, W, U, \nu)$, which was introduced by [**?**] and studied in [**?**], [**?**] and [**?**], where the nonlinear operator $\Pi(\cdot)$ is defined as follows:

$$\Pi(S, A, B, Q, R, \nu) \stackrel{\Delta}{=} A'SA + W - \nu\, A'SB(B'SB + U)^{-1}B'SA \tag{6.60}$$

In particular, Equation (6.46), i.e. $S_{k+1} = \Pi(S_k, A, B, W, U, \nu)$, is the dual of the estimator equation presented in [**?**], i.e. $P_{k+1} = \Pi(P_k, A', C', Q, R, \gamma)$. The results about the MARE are summarized in the following lemma

**Lemma 6.8.** *Consider the modified Riccati equation defined in Equation (6.60). Let $A$ be unstable, $(A, B)$ be controllable, and $(A, W^{\frac{1}{2}})$ be observable. Then:*

1. *The MARE has a unique strictly positive definite solution $S_\infty$ if and only if $\nu > \nu_c$, where $\nu_c$ is the critical arrival probability defined as:*

$$\nu_c \stackrel{\Delta}{=} \inf_{\nu} \{0 \leq \nu \leq 1, S \geq 0) \,|\, S = \Pi(S, A, B, W, U, \nu)\}.$$

2. *The critical probability $\nu_c$ satisfy the following analytical bounds:*

$$\begin{aligned} p_{min} &\leq \nu_c \leq p_{max} \\ p_{min} &\stackrel{\Delta}{=} 1 - \frac{1}{\max_i |\lambda_i^u(A)|^2} \\ p_{max} &\stackrel{\Delta}{=} 1 - \frac{1}{\prod_i |\lambda_i^u(A)|^2} \end{aligned}$$

*where $\lambda_i^u(A)$ are the unstable eigenvalues of $A$. Moreover, $\nu_c = p_{min}$ when $B$ is square and invertible, and $\nu_c = p_{max}$ when $B$ is rank one.*

3. *The critical probability can be numerically computed via the solution of the following quasi-convex LMIs optimization problem:*

$$\begin{aligned}
\nu_c &= \mathrm{argmin}_{\bar{\nu}} \Psi_\nu(Y,Z) > 0, \quad 0 \leq Y \leq I. \\
\Psi_\nu(Y,Z) &= \begin{bmatrix} Y & \sqrt{\nu}(YA' + ZB') & \sqrt{1-\nu}YA' \\ \sqrt{\nu}(AY + BZ') & Y & 0 \\ \sqrt{1-\nu}AY & 0 & Y \end{bmatrix}
\end{aligned}$$

4. *If $\nu > \nu_c$, then $\lim_{k \to +\infty} S_k = S_\infty$ for all initial conditions $S_0 \geq 0$, where*

$$S_{k+1} = \Pi(S_k, A, B, W, U, \nu)$$

.

*Proof.* The proof of facts (a),(c), and (d) can be found in [**?**]. The proof $\nu_c = p_{min}$ when $B$ is square and invertible can be found in [**?**], and the proof $\nu_c = p_{max}$ when $B$ is rank one in [**?**]. □

In [**?**] statistical analysis of the optimal estimator was given, which we report here for convenience:

**Theorem 6.8** ([**?**])**.** *Consider the system (6.29)-(6.31) and the optimal estimator under TCP-like protocols, given by Equations (6.37),(6.39),(6.40),(6.42) and (6.43). Assume that $(A, Q^{\frac{1}{2}})$ is controllable, $(A, C)$ is observable, and $A$ is unstable. Then there exists a critical observation arrival probability $\gamma_c$, such that the expectation of estimator error covariance is bounded if and only if the observation arrival probability is greater than the critical arrival probability, i.e.*

$$\mathbb{E}_\gamma[P_{k|k}] \leq M \ \forall k \text{ iff } \bar{\gamma} > \gamma_c.$$

*where $M$ is a positive definite matrix possibly dependent on $P_0$. Moreover, it is possible to compute a lower and an upper bound for the critical observation arrival probability $\gamma_c$, i.e.:*

$$p_{min} \leq \gamma_c \leq \gamma_{max} \leq p_{max}$$

*, where:*

$$\gamma_{max} \stackrel{\Delta}{=} \inf_{\gamma}\{0 \leq \gamma \leq 1, P \geq 0) \,|\, P = \Pi(P, A', C', Q, R, \gamma)\},$$

*where $p_{min}$ and $p_{max}$ are defined in Lemma 6.8.*

*Proof.* The proof can be found in [**?**] and is therefore omitted. □

Using the previous theorem and the results from the previous section, we can prove the following theorem for the infinite horizon optimal LQG under TCP-like protocols:

**Theorem 6.9.** *Consider the same system as defined in the previous theorem with the following additional hypothesis: $W_N = W_k = W$ and $U_k = U$. Moreover, let $(A, B)$ and $(A, Q^{\frac{1}{2}})$ be controllable, and let $(A, C)$ and $(A, W^{\frac{1}{2}})$ be observable. Moreover, suppose that $\bar{\nu} > \nu_c$ and $\bar{\gamma} > \gamma_{max}$, where $\nu_c$ and $\gamma_{max}$ are defined in Lemma 6.8 and in Theorem 6.8, respectively. Then we have:*

1. *The infinite horizon optimal controller gain is constant:*
$$\lim_{k\to\infty} L_k = L_\infty = -(B'S_\infty B + U)^{-1}B'S_\infty A \tag{6.61}$$
2. *The infinite horizon optimal estimator gain $K_k$, given by Equation (6.43), is stochastic and time-varying since it depends on the past observation arrival sequence $\{\gamma_j\}_{j=1}^k$.*
3. *The expected minimum cost can be bounded by two deterministic sequences:*
$$\frac{1}{N}J_N^{min} \leq \frac{1}{N}J_N^* \leq \frac{1}{N}J_N^{max} \tag{6.62}$$
*where $J_N^{min}, J_N^{max}$ converge to the following values:*
$$\begin{aligned} J_\infty^{max} &\overset{\Delta}{=} \lim_{N\to+\infty}\frac{1}{N}J_N^{max} \\ &= \operatorname{trace}((A'S_\infty A + W - S_\infty)(\widehat{P}_\infty - \bar{\gamma}\widehat{P}_\infty C'(C\widehat{P}_\infty C' + R)^{-1}C\widehat{P}_\infty)) + \operatorname{trace}(S_\infty Q) \\ J_\infty^{min} &\overset{\Delta}{=} \lim_{N\to+\infty}\frac{1}{N}J_N^{min} \\ &= (1-\bar{\gamma})\operatorname{trace}\Big((A'S_\infty A + W - S_\infty)\widetilde{P}_\infty\Big) + \operatorname{trace}(S_\infty Q), \end{aligned}$$
*and the matrices $S_\infty, \overline{P}_\infty, \underline{P}_\infty$ are the positive definite solutions of the following equations:*
$$\begin{aligned} S_\infty &= A'S_\infty A + W - \bar{\nu}\, A'S_\infty B(B'S_\infty B + U)^{-1}B'S_\infty A \\ \overline{P}_\infty &= A\overline{P}_\infty A' + Q - \bar{\gamma}\, A\overline{P}_\infty C'(C\overline{P}_\infty C' + R)^{-1}C\overline{P}_\infty A' \\ \underline{P}_\infty &= (1-\bar{\gamma})A\underline{P}_\infty A' + Q \end{aligned}$$

*Proof.* (a) Since by hypothesis $\bar{\nu} > \nu_c$, from Lemma 6.8(d) follows that $\lim_{k\to+\infty} S_k = S_\infty$. Therefore Equation (6.61) follows from Equation (6.48).

(b) This follows from the dependence on the arrival sequence $\{\gamma_k\}$ of the optimal state estimator given by Equations (6.37),(6.39),(6.40),(6.42) and (6.43). Since $\bar{\nu} > \nu_c$

(c) Equation (6.53) can be written in terms of the MARE as:
$$\widehat{P}_{k+1|k} = \Pi(\widehat{P}_{k|k-1}, A', C', Q, R, \gamma)$$
, therefore since $\bar{\gamma} > \gamma_{max}$ from Lemma 6.8(d) it follows that $\lim_{k\to+\infty}\widehat{P}_{k|k-1} = \overline{P}_\infty$, where $\overline{P}_\infty$ is the solution of the MARE $\overline{P}_\infty = \Pi(\overline{P}_\infty, A', C', Q, R, \gamma)$.

Also $\lim_{k\to+\infty} \widetilde{P}_{k|k-1} = \underline{P}_\infty$, where $\widetilde{P}_{k|k-1}$ is defined in Equation (6.55) and $\underline{P}_\infty$ is the solution of the Lyapunov equation $\widehat{P}_\infty = \tilde{A}\widehat{P}_\infty\tilde{A}' + Q$, where $\tilde{A} = \sqrt{1-\bar{\gamma}}A$. Such solution clearly exists since $\sqrt{1-\bar{\gamma}} < \frac{1}{p_{min}} = \frac{1}{\max_i |\lambda_i^u(A)|}$ and thus the matrix $\tilde{A}$ is strictly stable. From Equations (6.54) and (6.56) it follows that $\lim_{k\to+\infty} \widehat{P}_{k|k} = \overline{P}_\infty - \bar{\gamma}\overline{P}_\infty C'(C\overline{P}_\infty C' + R)^{-1}C\overline{P}_\infty$ and $\lim_{k\to+\infty} \widetilde{P}_{k|k} = (1-\bar{\gamma})\underline{P}_\infty$. Also $\lim_{k\to+\infty} S_{k+1} = \lim_{k\to+\infty} S_k = S_\infty$. Finally from Equations (6.57) - (6.59) and the previous observations follow the claim.

□

## 6.13 LQG control for UDP-like protocols

In this section equations for the optimal estimator and controller design for the case of communication protocols that do not provide any kind of acknowledgment of successful packet delivery (UDP-like). This case corresponds to the information set $G_k$, as defined in Equation (6.32). Some of the derivations are analogous to the previous section and are therefore skipped.

### Estimator Design

We derive the equations for the optimal estimator using similar arguments to the standard Kalman filtering equations. The innovation step is given by:

$$\begin{aligned}
\hat{x}_{k+1|k} &\overset{\Delta}{=} \mathbb{E}[x_{k+1}|\mathcal{G}_k] = \mathbb{E}[Ax_k + \nu_k Bu_k + w_k|\mathcal{G}_k] \\
&= A\mathbb{E}[x_k|\mathcal{G}_k] + \mathbb{E}[\nu_k]Bu_k \\
&= A\hat{x}_{k|k} + \bar{\nu}Bu_k \qquad (6.63)
\end{aligned}$$

$$\begin{aligned}
e_{k+1|k} &\overset{\Delta}{=} x_{k+1} - \hat{x}_{k+1|k} \\
&= Ax_k + \nu_k Bu_k + w_k - (A\hat{x}_{k|k} + \bar{\nu}Bu_k) \\
&= Ae_{k|k} + (\nu_k - \nu)Bu_k + w_k \qquad (6.64)
\end{aligned}$$

$$\begin{aligned}
P_{k+1|k} &\overset{\Delta}{=} \mathbb{E}[e_{k+1|k}e'_{k+1|k}\ |\mathcal{G}_k] \\
&= A\mathbb{E}[e_{k|k}e'_{k|k}|\mathcal{G}_k]A' + \mathbb{E}[(\nu_k - \nu)^2]Bu_k u'_k B' + \mathbb{E}[w_k w'_k] \\
&= AP_{k|k}A' + \bar{\nu}(1-\bar{\nu})Bu_k u'_k B' + Q, \qquad (6.65)
\end{aligned}$$

where we used the independence and zero-mean of $w_k$, $(\nu_k - \bar{\nu})$, and $\mathcal{G}_k$, and the fact that $u_k$ is a deterministic function of the information set $\mathcal{G}_k$. Note how under UDP-like communication, differently from TCP-like, the error covariance $P_{k+1|k}$ depends explicitly on the control input $u_k$. This is the main difference with control feedback systems under TCP-like protocols.

The correction step is the same as for the TCP case:

$$\begin{aligned}\hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + \gamma_{k+1}K_{k+1}(y_{k+1} - C\hat{x}_{k+1|k})\\ P_{k+1|k+1} &= P_{k+1|k} - \gamma_{k+1}K_{k+1}CP_{k+1|k}, &(6.66)\\ K_{k+1} &\overset{\Delta}{=} P_{k+1|k}C'(CP_{k+1|k}C' + R)^{-1}, &(6.67)\end{aligned}$$

where again we considered a time varying system with $A_k = A$ and $C_k = \gamma_k C$ as we did for the optimal estimator under TCP-like protocols.

### Controller design: General case

In this section, we show that the optimal LQG controller, under UDP-like communication protocols, is in general not a linear function of the state estimate, and that the estimator and controller design cannot be separated anymore. To show this, we construct a counter-example considering a simple scalar system and we proceed using the dynamic programming approach. Let us consider the scalar system where $A = 1, B = 1, C = 1, W_N = W_k = 1, U_k = 0, R = 1, Q = 0$. Similarly to the TCP case, we define the value function, $V_k(x_k)$, as in Equations (6.44) where we just need to substitute the information set $\mathcal{F}_k$ with $\mathcal{G}_k$. For $k = N$, the value function is given by $V_N(x_N) = \mathbb{E}[x'_N W_N x_N \mid \mathcal{G}_N] = \mathbb{E}[x_N^2 \mid \mathcal{G}_N]$. For $k = N-1$ we have:

$$\begin{aligned}V_{N-1}(x_{N-1}) &= \min_{u_{N-1}} \mathbb{E}[x_{N-1}^2 + V_N(x_N) \mid \mathcal{G}_{N-1}]\\ &= \min_{u_{N-1}} \mathbb{E}[x_{N-1}^2 + x_N^2 \mid \mathcal{G}_{N-1}]\\ &= \min_{u_{N-1}} \mathbb{E}[x_{N-1}^2 + (x_{N-1} + \nu_{N-1}u_{N-1})^2 \mid \mathcal{G}_{N-1}]\\ &= \min_{u_{N-1}} (\mathbb{E}[2x_{N-1}^2|\mathcal{G}_{N-1}] + \mathbb{E}[\nu_{N-1}^2]u_{N-1}^2 + 2u_{N-1}\mathbb{E}[\nu_{N-1}]\mathbb{E}[x_{N-1}|\mathcal{G}_{N-1}])\\ &= \min_{u_{N-1}} (\mathbb{E}[2x_{N-1}^2|\mathcal{G}_{N-1}] + \bar{\nu}u_{N-1}^2 + 2\bar{\nu}u_{N-1}\hat{x}_{N-1|N-1}),\end{aligned}$$

where we used the independence of $\nu_{N-1}$ and $\mathcal{G}_{N-1}$, and the fact that $u_{N-1}$ is a deterministic function of the information set $\mathcal{G}_{N-1}$. The cost is a quadratic function of the input $u_{N-1}$, therefore the minimizer can be simply obtained by finding $\frac{\partial V_{N-1}}{\partial u_{N-1}} = 0$, which is given by $u_{N-1}^* = -\hat{x}_{N-1|N-1}$. If we substitute back $u_{N-1}^*$ into the value function we have:

$$\begin{aligned}V_{N-1}(x_{N-1}) &= \mathbb{E}[2x_{N-1}^2|\mathcal{G}_{N-1}] - \bar{\nu}\hat{x}_{N-1|N-1}^2\\ &= \mathbb{E}[(2-\bar{\nu})x_{N-1}^2|\mathcal{G}_{N-1}] + \bar{\nu}P_{N-1|N-1}\end{aligned}$$

where we used Lemma 6.5(b). Before proceeding note that:

$$
\begin{aligned}
P_{N-1|N-1} &= P_{N-1|N-2} - \gamma_{N-1}\frac{P^2_{N-1|N-2}}{P_{N-1|N-2}+1}\\
&= P_{N-1|N-2} - \gamma_{N-1}\left(P_{N-1|N-2} - 1 + \frac{1}{P_{N-1|N-2}+1}\right)\\
&= (1-\gamma_{N-1})\left(P_{N-2|N-2} + \bar{\nu}(1-\bar{\nu})u^2_{N-2}\right) + \gamma_{N-1} + \\
&\quad +\gamma_{N-1}\frac{1}{P_{N-2|N-2} + \bar{\nu}(1-\bar{\nu})u^2_{N-2} + 1}\\
\mathbb{E}[P_{N-1|N-1}|\mathcal{G}_{N-2}] &= (1-\bar{\gamma})\left(P_{N-2|N-2} + \bar{\nu}(1-\bar{\nu})u^2_{N-2}\right) + \bar{\gamma} + \bar{\gamma}\frac{1}{P_{N-2|N-2} + \bar{\nu}(1-\bar{\nu})u^2_{N-2} + 1}\\
\mathbb{E}[x^2_{N-1}|\mathcal{G}_{N-2}] &= \mathbb{E}[(x_{N-2} + \nu_{N-2}u_{N-2})^2|\mathcal{G}_{N-2}]\\
&= \mathbb{E}[x^2_{N-2}|\mathcal{G}_{N-2}] + 2\mathbb{E}[\nu_{N-2}]\mathbb{E}[x_{N-2}|\mathcal{G}_{N-2}]u_{N-2} + \mathbb{E}[\nu_{N-2}]u^2_{N-2}\\
&= \mathbb{E}[x^2_{N-2}|\mathcal{G}_{N-2}] + 2\bar{\nu}\hat{x}_{N-2|N-2}u_{N-2} + \bar{\nu}u^2_{N-2},
\end{aligned}
$$

where we used Equations (6.65)-(6.67), and the fact that $u_{N-2}$ and $P_{N-2|N-2}$ are a deterministic function of the information set $\mathcal{G}_{N-2}$. Using the previous equations we proceed to compute the value function for $k = N-2$:

$$
\begin{aligned}
V_{N-2}(x_{N-2}) &= \min_{u_{N-2}} \mathbb{E}[x^2_{N-2} + V_{N-1}(x_{N-1}) \mid \mathcal{G}_{N-2}]\\
&= \min_{u_{N-2}} \mathbb{E}[x^2_{N-2} + (2-\bar{\nu})x^2_{N-1} + \bar{\nu}P_{N-1|N-1} \mid \mathcal{G}_{N-2}]\\
&= (3-\bar{\nu})\mathbb{E}[x^2_{N-2}|\mathcal{G}_{N-2}] + \bar{\nu}(1-\bar{\gamma})P_{N-2|N-2} + \bar{\nu}\bar{\gamma} + \\
&\quad + \min_{u_{N-1}}\Bigg(2\bar{\nu}(2-\bar{\nu})\hat{x}_{N-2|N-2}u_{N-2} + \bar{\nu}(2-\bar{\nu})u^2_{N-2} + \\
&\quad + \bar{\nu}^2(1-\bar{\nu})(1-\bar{\gamma})u^2_{N-2} + \bar{\nu}\bar{\gamma}\frac{1}{P_{N-2|N-2} + \bar{\nu}(1-\bar{\nu})u^2_{N-2} + 1}\Bigg)
\end{aligned}
$$

The first three terms inside the round parenthesis are convex quadratic functions of the control input $u_{N-2}$, however the last term is not. Therefore, the minimizer $u^*_{N-2}$ is, in general, a non-linear function of the information set $\mathcal{G}_k$. The nonlinearity of the optimal controller arises from the fact that the correction error covariance matrix $P_{k+1|k+1}$ is a non-linear function of the innovation error covariance $P_{k+1|k}$, as it can be seen in Equations (6.66) and (6.67). The only case when $P_{k+1|k+1}$ is linear in $P_{k+1|k}$ is when measurement noise covariance $R = 0$ and the observation matrix $C$ is square and invertible, from which follows that the optimal control is linear in the estimated states. However it is important to remark that the separation principle still does not hold even for this special case, since the control input affects the estimator error covariance.

We can summarize these results in the following theorem:

**Theorem 6.10.** *Let us consider the stochastic system defined in Equa-*

*tions (6.29) with horizon $N \geq 2$. Then:*

1. *The separation principle does not hold since the estimator error covariance depends on the control input, as shown in Equation (6.65).*

2. *The optimal control feedback $u_k = g_k^*(\mathcal{G}_k)$ that minimizes the cost functional defined in Equation (6.33) under UDP-like protocols is, in general, a nonlinear function of information set $\mathcal{G}_k$.*

3. *The optimal control feedback $u_k = g_k^*(\mathcal{G}_k)$ is a linear function of the estimated state $\hat{x}_{k|k}$ if and only if the matrix $C$ is invertible and there is no measurement noise.*

The next section will compute explicitly the optimal control for the special case and will give necessary and sufficient conditions for stability and performance of the infinite horizon scenario.

### Special Case: R=0 and C invertible

Without loss of generality we can assume $C = I$, since the linear transformation $z = Cx$ would give an equivalent system where the matrix $C$ is the identity. Let us now consider the case when there is no measurement noise, i.e. $R = 0$. These assumption mean that it is possible to measure the state $x_k$ when a packet is delivered. In this case the estimator Equations (6.65)-(6.67) simplify as follows:

$$\begin{aligned} K_{k+1} &= I && (6.68)\\ P_{k+1|k+1} &= (1-\gamma_{k+1})P_{k+1|k} \\ &= (1-\gamma_{k+1})(A'P_{k|k}A + Q + \bar{\nu}(1-\bar{\nu})Bu_k u_k' B') && (6.69)\\ \mathbb{E}[P_{k+1|k+1}|\mathcal{G}_k] &= (1-\bar{\gamma})(A'P_{k|k}A + Q + \bar{\nu}(1-\bar{\nu})Bu_k u_k' B') && (6.70)\end{aligned}$$

where in the last equation we used independence of $\gamma_{k+1}$ and $\mathcal{G}_k$, and we used the fact that $P_{k|k}$ is a deterministic function of $\mathcal{G}_k$.

Similarly to what done in the analysis of TCP-like optimal control, we claim that the value function $V_k^*(x_k)$ can be written as follows:

$$V_k(x_k) = \hat{x}_{k|k}' S_k \hat{x}_{k|k} + \operatorname{trace}(T_k P_{k|k}) + \operatorname{trace}(D_k Q) \qquad (6.71)$$

for $k = N, \ldots, 0$. This is clearly true for $k = N$, in fact we have:

$$V_N(x_N) = \mathbb{E}[x_N' W_N x_N | \mathcal{G}_N] = \hat{x}_{N|N}' W_N \hat{x}_{N|N} + \operatorname{trace}(W_N P_{N|N})$$

where we used Lemma 6.5(b), therefore the statement is satisfied by $S_N = W_N, T_N = W_N, D_N = 0$. Note that Equation (6.71) can be rewritten as follows:

$$V_k(x_k) = \mathbb{E}[x_k' S_k x_k | \mathcal{G}_k] + \operatorname{trace}\big((T_k - S_k)P_{k|k}\big) + \operatorname{trace}(D_k Q)$$

where we used once again Lemma 6.5(b). Moreover, to simplify notation we define $H_k \overset{\Delta}{=} (T_k - S_k)$. Let us suppose that Equation (6.71) is true for $k+1$ and let us show by induction it holds true for $k$:

$$\begin{aligned}
V_k(x_k) &= \min_{u_k} \mathbb{E}[x_k' W_k x_k + \nu_k u_k' U_k u_k + V_{k+1}(x_{k+1}) \mid \mathcal{G}_k] \\
&= \min_{u_k} \big(\mathbb{E}[x_k' W_k x_k + \nu_k u_k' U_k u_k + x_{k+1}' S_{k+1} x_{k+1} + \mathrm{trace}(H_{k+1} P_{k+1|k+1}) + \mathrm{trace}(D_{k+1} Q) \mid \mathcal{G}_k]\big) \\
&= \mathbb{E}[x_k'(W_k + A' S_{k+1} A) x_k | \mathcal{G}_k] + \mathrm{trace}(S_{k+1} Q) + (1-\bar{\gamma})\mathrm{trace}(H_{k+1}(A' P_{k|k} A + Q)) + \mathrm{trace}(D_{k+1} Q) + \\
&\quad + \min_{u_k} \Big(\bar{\nu} u_k' U_k u_k + \bar{\nu} u_k' B' S_{k+1} B u_k + 2\bar{\nu} u_k' B' S_{k+1} A \hat{x}_{k|k} + \bar{\nu}(1-\bar{\nu})(1-\bar{\gamma})\mathrm{trace}(H_{k+1} B u_k u_k' B')\Big) \\
&= \mathbb{E}[x_k'(W_k + A' S_{k+1} A) x_k | \mathcal{G}_k] + \mathrm{trace}\big((D_{k+1} + (1-\bar{\gamma}) H_{k+1}) Q\big) + (1-\bar{\gamma})\mathrm{trace}(A H_{k+1} A' P_{k|k}) + \\
&\quad + \mathrm{trace}(S_{k+1} Q) + \bar{\nu} \min_{u_k} \Big(u_k'\big(U_k + B'(S_{k+1} + (1-\bar{\nu})(1-\bar{\gamma}) H_{k+1}) B\big) u_k + 2 u_k' B' S_{k+1} A \hat{x}_{k|k}\Big) \\
&= \hat{x}_{k|k}' (W_k + A' S_{k+1} A) \hat{x}_{k|k} + \mathrm{trace}\big((D_{k+1} + (1-\bar{\gamma}) T_{k+1} + \bar{\gamma} S_{k+1}) Q\big) + \\
&\quad + \mathrm{trace}\Big((W_k + \bar{\gamma} A' S_{k+1} A + (1-\bar{\gamma}) A T_{k+1} A') P_{k|k}\Big) + \\
&\quad + \bar{\nu} \min_{u_k} \Big(u_k'\big(U_k + B'((1-\bar{\alpha}) S_{k+1} + \bar{\alpha} T_{k+1}) B\big) u_k + 2 u_k' B' S_{k+1} A \hat{x}_{k|k}\Big),
\end{aligned}$$

where we defined $\bar{\alpha} = (1-\bar{\nu})(1-\bar{\gamma})$, we used Lemma 6.5(c) to get the second equality, and Equations (6.36) and (6.70) to get the last equality. Since the quantity inside the big round parenthesis a convex quadratic function, the minimizer is the solution of $\frac{\partial V_k}{\partial u_k} = 0$ which is given by:

$$\begin{aligned}
u_k^* &= -\Big(U_k + B'\big((1-\bar{\alpha}) S_{k+1} + \bar{\alpha} T_{k+1}\big) B\Big)^{-1} B' S_{k+1} A \, \hat{x}_{k|k} && (6.72) \\
&= L_k \, \hat{x}_{k|k} && (6.73)
\end{aligned}$$

which is linear function of the estimated state $\hat{x}_{k|k}$. Substituting back into the value function we get:

$$\begin{aligned}
V_k(x_k) &= \hat{x}_{k|k}' (W_k + A' S_{k+1} A) \hat{x}_{k|k} + \mathrm{trace}\big((D_{k+1} + (1-\bar{\gamma}) T_{k+1} + \bar{\gamma} S_{k+1}) Q\big) + \\
&\quad + \mathrm{trace}\Big((W_k + A' S_{k+1} A + (1-\bar{\gamma}) A T_{k+1} A') P_{k|k}\Big) - \bar{\nu} \hat{x}_{k|k}' A' S_{k+1} B L_k \hat{x}_{k|k} \\
&= \hat{x}_{k|k}' (W_k + \bar{\gamma} A' S_{k+1} A - \bar{\nu} \hat{x}_{k|k}' A' S_{k+1} B L_k) \hat{x}_{k|k} + \mathrm{trace}\big((D_{k+1} + (1-\bar{\gamma}) T_{k+1} + \bar{\gamma} S_{k+1}) Q\big) + \\
&\quad + \mathrm{trace}\Big((W_k + A' S_{k+1} A + (1-\bar{\gamma}) A T_{k+1} A') P_{k|k}\Big),
\end{aligned}$$

where we used Lemma 6.5(b) in the last equality. From the last equation we see that the value function can be written as in Equation (6.71) if and only if the following equations are satisfied:

$$\begin{aligned}
S_k &= A' S_{k+1} A + W_k - \bar{\nu} A' S_{k+1} B \left(U_k + B' \left((1-\bar{\alpha}) S_{k+1} + \bar{\alpha} T_{k+1}\right) B\right)^{-1} B' S_{k+1} A \\
&= \Phi^S_{\gamma,\nu}(S_{k+1}, T_{k+1}) && (6.74) \\
T_k &= (1-\bar{\gamma}) A' T_{k+1} A + \bar{\gamma} A' S_{k+1} A + W_k \\
&= \Phi^T_{\gamma,\nu}(S_{k+1}, T_{k+1}) && (6.75) \\
D_k &= (1-\bar{\gamma}) T_{k+1} + \bar{\gamma} S_{k+1} + D_{k+1} && (6.76)
\end{aligned}$$

The optimal minimal cost for the finite horizon, $J_N^* = V_0(x_0)$ is then given by:

$$J_N^* = \overline{x}_0' S_0 \overline{x}_0 + \text{trace}(S_0 P_0) + \sum_{k=1}^{N} \text{trace}\Big(\big((1-\bar{\gamma})T_k + \bar{\gamma}S_k\big)Q\Big) \tag{6.77}$$

For the infinite horizon optimal controller, necessary and sufficient condition for the average minimal cost $J_\infty \overset{\Delta}{=} \lim_{N\to+\infty} \frac{1}{N} J_N^*$ to be finite is that the coupled iterative Equations (6.74) and (6.75) should converge to a finite value $S_\infty$ and $T_\infty$ as $N \to +\infty$. In the work of Imer *et al.* [**?**] similar equations were derived for the optimal LQG control under UDP for the same framework with the additional conditions $Q = 0$ and $B$ square and invertible. They find necessary and sufficient conditions for those equations to converge. Unfortunately, these conditions do not hold for the general case when $B$ in not square. This is a very frequent situation in control systems, where in general we simply have $(A, B)$ controllable.

**Theorem 6.11.** *Also, assume that the pair $(A, W^{1/2})$ is observable. Consider the following operator:*

$$\Upsilon(S,T,L) = A'SA + W + 2\bar{\nu}A'SBL + \bar{\nu}L'\Big(U + B'\big((1-\bar{\alpha})S + \bar{\alpha}T\big)B\Big)L \tag{6.78}$$

*Then the following claims are equivalent:*

1. *There exist a matrix $\tilde{L}$ and positive definite matrices $\tilde{S}$ and $\tilde{T}$ such that:*
$$\tilde{S} > 0,\ \tilde{T} > 0,\ \tilde{S} = \Upsilon(\tilde{S},\tilde{T},\tilde{L}),\ \tilde{T} = \Phi^T(\tilde{S},\tilde{T})$$

2. *Consider the sequences:*
$$S_{k+1} = \Phi^S(S_k, T_k), \quad T_{k+1} = \Phi^T(S_k, T_k)$$
*where the operators $\Phi^S(\cdot), \Phi^T(\cdot)$ are defined in Equations (6.74) and (6.75). For any initial condition $S_0, T_0 \geq 0$ we have*
$$\lim_{k\to\infty} S_k = S_\infty, \quad \lim_{k\to\infty} T_k = T_\infty$$
*and $S_\infty, T_\infty$ are the unique positive definite solution of the following equations*
$$S_\infty > 0,\ T_\infty > 0,\ S_\infty = \Phi^S(S_\infty, T_\infty), \quad T_\infty = \Phi^T(S_\infty, T_\infty)$$

The convergence of Equations (6.74) and (6.75) depend on the control and observation arrival probabilities $\bar{\gamma}, \bar{\nu}$. General analytical conditions for convergence are not available, but some necessary and sufficient conditions can be found.

**Lemma 6.9.** *Let us consider the fixed points of Equations (6.74) and (6.75), i.e. $S = \Phi^S(S,T), T = \Phi^T(S,T)$ where $S,T \geq 0$. Let $A$ be unstable. A necessary condition for existence of solution is*

$$|A|^2(\bar{\gamma} + \bar{\nu} - 2\bar{\gamma}\bar{\nu}) < \bar{\gamma} + \bar{\nu} - \bar{\gamma}\bar{\nu} \tag{6.79}$$

*where $|A| \stackrel{\Delta}{=} \max_i |\lambda_i(A)|$ is the largest eigenvalue of the matrix $A$.*

**Lemma 6.10.** *Let us consider the fixed points of Equations (6.74) and (6.75), i.e. $S = \Phi^S(S,T), T = \Phi^T(S,T)$ where $S,T \geq 0$. Let $A$ be unstable, $(A, W^{1/2})$ observable and $B$ square and invertible. Then a sufficient condition for existence of solution is*

$$|A|^2(\bar{\gamma} + \bar{\nu} - 2\bar{\gamma}\bar{\nu}) < \bar{\gamma} + \bar{\nu} - \bar{\gamma}\bar{\nu} \tag{6.80}$$

*where $|A| \stackrel{\Delta}{=} \max_i |\lambda_i(A)|$ is the largest eigenvalue of the matrix $A$.*

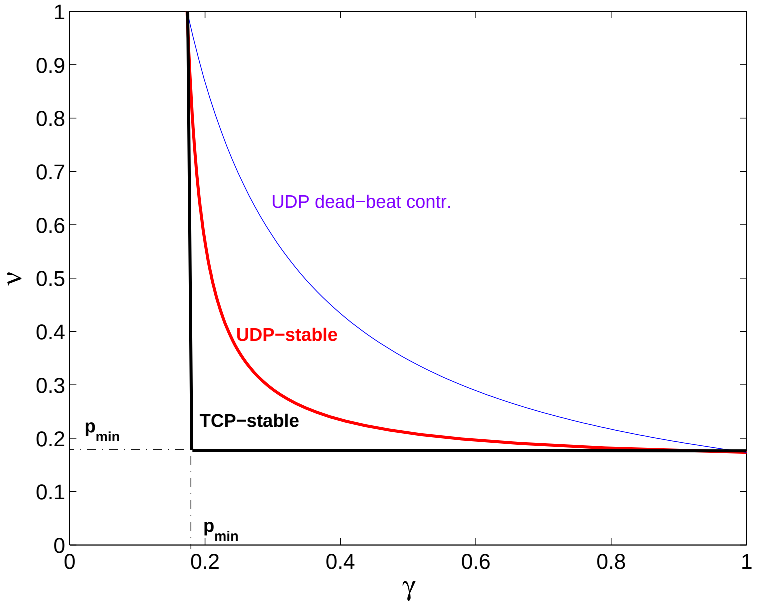


**Figure 6.10:** Region of convergence for UDP-like and TCP-like optimal control in the scalar case. The bounds are tight in the scalar case. The thin solid line corresponds to the boundary of the stability region for a dead-beat controller under UDP-like protocols as given by [**?**], which is much more restrictive than what can be achieved with optimal UDP controllers.

A graphical representation of the stability bounds are shown in Figure 6.10, where we considered a scalar system with parameters $|A| = 1.1$, which gives the critical probability $p_{min} = 1 - 1/|A|^2 = 1.173$ as defined in Theorem 6.8. The critical arrival probabilities for TCP-like optimal control are

$\gamma_c = \nu_c = p_{min}$. The boundary for the stability region of optimal control under UDP-like protocols given in Lemma 6.10 can be written also as $\bar{\nu} > \frac{\bar{\gamma}(A^2-1)}{\bar{\gamma}(2A^2-1)+1-A^2}$ for $\bar{\gamma} > p_{min}$. It is important to remark that the stability region of optimal control under UDP-like protocols is larger than the stability region obtained using a dead-beat controller proposed in [**?**], i.e. $u_k = -\gamma_k B^{-1} A y_k = -\gamma_k B^{-1} A x_k$, which is given by $\bar{\gamma}\bar{\nu} > 1 - 1/|A|^2$ and graphically shown in Figure 6.10 . This is not surprising since the dead-beat controller is rather aggressive and requires a large gain $L$, which increases the estimator error covariance in Equation (6.70). Indeed, as shown in the constructive proof of Lemma 6.10, controllers with similar structure but smaller gains, i.e. $u_k = -\eta\gamma_k B^{-1} A y_k = -\eta\gamma_k B^{-1} A x_k$ where $\eta < 1$, have a larger region of stability.

We can summarize the results of this section in the following theorem

**Theorem 6.12.** *Consider the system (6.29)-(6.31) and consider the problem of minimizing the cost function (6.33) within the class of admissible policies $u_k = f(\mathcal{G}_k)$, where $\mathcal{G}_k$ is the information available under TCP-like schemes, given in Equation (6.32). Assume also that $R = 0$ and $C$ is square and invertible. Then:*

1. *The optimal estimator gain is constant and in particular $K_k = I$ if $C = I$.*

2. *The infinite horizon optimal control exists if and only if there exists positive definite matrices $S_\infty, T_\infty > 0$ such that $S_\infty = \Phi^S(S_\infty, T_\infty)$ and $T_\infty = \Phi^T(S_\infty, T_\infty)$, where $\Phi^S$ and $\Phi^S$ are defined in Equations (6.74) and (6.75).*

3. *The infinite horizon optimal controller gain is constant:*
$$\lim_{k\to\infty} L_k = L_\infty = -(B'(\bar{\alpha}T_\infty + (1-\bar{\alpha})S_\infty)B + U)^{-1} B' S_\infty A \quad (6.81)$$

4. *A necessary condition for existence of $S_\infty, T_\infty > 0$ is*
$$|A|^2(\bar{\gamma} + \bar{\nu} - 2\bar{\gamma}\bar{\nu}) < \bar{\gamma} + \bar{\nu} - \bar{\gamma}\bar{\nu} \quad (6.82)$$
*where $|A| \overset{\Delta}{=} \max_i |\lambda_i(A)|$ is the largest eigenvalue of the matrix $A$. This condition is also sufficient if $B$ is square and invertible.*

5. *The expected minimum cost converges:*
$$J^*_\infty = \lim_{k\to\infty} \frac{1}{N} J^*_N = \mathrm{trace}\big((1-\bar{\gamma})T_\infty + \bar{\gamma}S_\infty)Q\big) \quad (6.83)$$

In the scenario considered in this section when $R = 0$ and $C$ is invertible, it is possible to directly compare the performance of optimal control under TCP-like and UDP-like protocols in terms of the infinite horizon cost $J^*_\infty$. Let us consider for example the scalar system with the following parameters $A = 1.1, B = C = Q = W = U = 1, R = 0$. For simplicity also consider

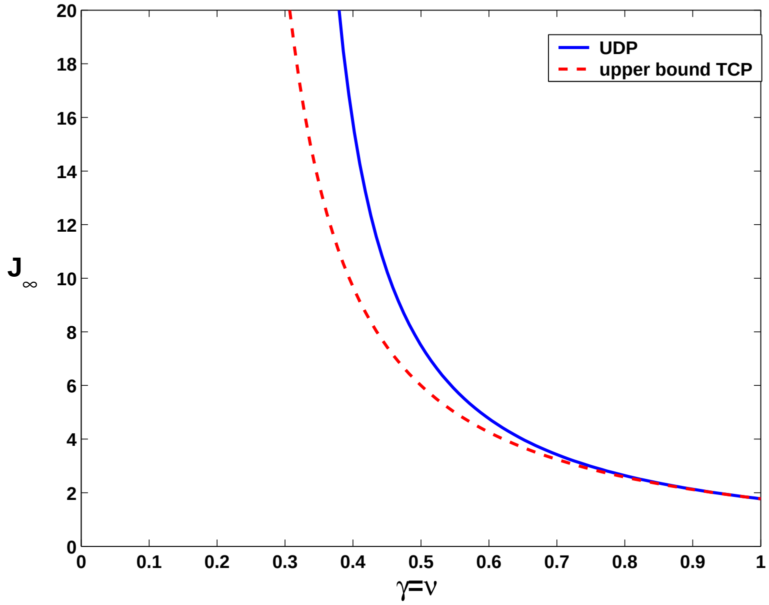


**Figure 6.11:** Exact infinite horizon cost using optimal LQG control under UDP-like and upper bound under TCP-like communication protocols in the scalar case.

symmetric communication channels for sensor reading and control inputs, i.e. $\bar{\nu} = \bar{\gamma}$. Using results from Theorem 6.9 and Theorem 6.12 we can compute the infinite horizon cost using optimal controllers under UDP-like and an upper bound on the cost under TCP-like communication protocols, which are shown in Fig. 6.11. As expected optimal control performance under TCP-like is better than UDP-like, however the two curves are comparable for moderate packet loss. Although the TCP-like curve is only an upper bound of the true expected cost, it has been observed to be rather close to the empirical cost [**?**]. The observation that TCP-like and UDP-like optimal control performances seem remarkably close is extremely valuable since UDP-like protocols are much simpler to implement than TCP-like.

## 6.14 Appendix

**Lemma 6.11.** *Let $S, T \in \mathbb{M} = \{M \in \mathbb{R}^{n\times n} | M \geq 0\}$. Consider the operators $\Phi^S(S,T)$, and $\Phi^T(S,T)$ as defined in Equations (6.74) and (6.75), and consider the sequences $S_{k+1} = \Phi^S(S_k, T_k)$ and $T_{k+1} = \Phi^T(S_k, T_k)$. Consider $L^*_{S,T} = -\big(U + B'\big((1-\bar{\alpha})S + \bar{\alpha}T\big)B\big)^{-1} B'SA$. operators*

*Then the following facts are true:*

*1.*

$$\Upsilon(S,T,L) = (1-\frac{\bar\nu}{1-\bar\alpha})A'SA+W+\frac{\bar\nu}{1-\bar\alpha}\big(A+(1-\bar\alpha)BL\big)'S\big(A+(1-\bar\alpha)BL\big)+\bar\nu L'UL+\bar\nu\bar\alpha L'B'TBL$$

*2.* $\Phi^S(S,T) = \min_L \Upsilon(S,T,L)$

*3.* $0 \le \Upsilon(S,T,L^*_{S,T}) = \Phi^S(S,T) \le \Upsilon(S,T,L)\ \forall L$

*4. If* $S_{k+1} > S_k$ *and* $T_{k+1} > T_k$*, then* $S_{k+2} > S_{k+1}$ *and* $T_{k+2} > T_{k+1}$*.*

*5. If the pair* $(A, W^{1/2})$ *is observable and* $S = \Phi^S(S,T)$ *and* $T = \Phi^T(S,T)$*, then* $S > 0$ *and* $T > 0$*.*

*Proof.* Fact (a) can be easily checked by direct substitution.

(b) If $U$ is invertible then it is easy to verify by substitution that

$$\begin{aligned}\Upsilon(S,T,L) &= \Phi^S(S,T) + \bar\nu(L-L^*_{S,T})'\Big(U + B'\big((1-\bar\alpha)S+\bar\alpha T\big)B\Big)(L-L^*_{S,T})\\ &\ge \Phi^S(S,T)\end{aligned}$$

(c) The non-negativeness follows from the observation that $\Upsilon(S,T,L)$ is a sum of positive semi-definite matrices. In fact $(1-\frac{\bar\nu}{1-\bar\alpha}) = \frac{\bar\gamma(1-\bar\nu)}{\bar\nu+\bar\gamma(1-\bar\nu)} \ge 0$ and $0 \le \bar\alpha \le 1$.
The equality $\Upsilon(S,T,L^*_{S,T}) = \Phi^S(S,T)$ can be verified by direct substitution. The last inequality follows directly from Fact (b).

(d)

$$\begin{aligned}S_{k+2} = {}& \Phi^S(S_{k+1},T_{k+1}) = \Upsilon(S_{k+1},T_{k+1},L^*_{S_{k+1},T_{k+1}})\\ \ge {}& \Upsilon(S_k,T_k,L^*_{S_{k+1},T_{k+1}}) \ge \Upsilon(S_k,T_k,L^*_{S_k,T_k})\\ = {}& \Phi^S(S_k,T_k) = S_{k+1}\\ T_{k+2} = {}& \Phi^T(S_{k+1},T_{k+1}) \ge \Phi^T(S_k,T_k) = T_{k+1}\end{aligned}$$

(e) First observe that $S = \Phi^S(S,T) \ge 0$ and $T = \Phi^T(S,T) \ge 0$. Thus, to prove that $S,T > 0$, we only need to establish that $S,T$ are nonsingular. Suppose they are singular, the there exist vectors $0 \ne v_s \in \mathcal{N}(S)$ and $0 \ne v_t \in \mathcal{N}(T)$, i.e. $Sv_s = 0$ and $Tv_t = 0$, where $\mathcal{N}(\cdot)$ indicates the null space. Then

$$\begin{aligned}0 = {}& v_s'Sv_s = v_s'\Phi^S(S,T)v_s = v_s'\Upsilon(S,T,L^*_{S,T})v_s\\ = {}& (1-\tfrac{\bar\nu}{1-\bar\alpha})v_s'A'SAv_s + v_s'Wv_s + \star\end{aligned}$$

where $\star$ indicates other terms. Since all the terms are positive semi-definite matrices, this implies that all the term must be zero:

$$\begin{aligned}&v_s'A'SAv_s = 0 \Longrightarrow SAv_s = 0 \Longrightarrow Av_s \in \mathcal{N}(S)\\ &v_s'Wv_s = 0 \Longrightarrow W^{1/2}v_s = 0\end{aligned}$$

As a result, the null space $\mathcal{N}(S)$ is $A$-invariant. Therefore, $\mathcal{N}(S)$ contains an eigenvector of $A$, i.e. there exists $u \ne 0$ such that $Su = 0$ and $Au = \sigma u$. As before, we conclude that $Wu$=0. This implies (using the PBH test) that the pair $(A, W^{1/2})$ is not observable, contradicting the hypothesis. Thus, $\mathcal{N}(S)$ is empty, proving that $S > 0$. The same argument can be used to prove that also $T > 0$. □

### Proof of Theorem 6.11

(a)⇒(b) The main idea of the proof consists in the proving of the convergence of several monotonic sequences. Consider the sequences $V_{k+1} = \Upsilon(V_k, Z_k, \tilde{L})$ and $Z_{k+1} = \Phi^T(V_k, Z_k)$ with initial conditions $V_0 = Z_0 = 0$. It is easy to verify by substitution that $V_1 = W + \bar{\nu}\tilde{L}'U\tilde{L} \geq 0 = V_0$ and $Z_1 = W \geq 0 = Z_0$. Lemma 6.11(a) shows that the operator $\Upsilon(V, Z, \tilde{L})$ is linear and monotonically increasing in $V$ and $Z$, i.e.
$(V_{k+1} \geq V_k, Z_{k+1} \geq Z_k) \Rightarrow (V_{k+2} \geq V_{k+1}, Z_{k+2} \geq Z_{k+1})$. Also the operator $\Phi^T(V, Z)$ is linear and monotonically increasing in $V$ and $Z$. Since $V_1 \geq V_0$ and $Z_1 \geq Z_0$, using an induction argument we have that $V_{k+1} \geq V_k, Z_{k+1} \geq Z_k$ for all time $k$, i.e. the sequences are monotonically increasing. These sequences are also bounded, in fact $(V_0 \leq \tilde{S}), (Z_0 \leq \tilde{T}) \Rightarrow (V_1 = \Upsilon(0, 0, \tilde{L}) \leq \Upsilon(\tilde{S}, \tilde{T}, \tilde{L}) = \tilde{S}), (Z_1 = \Phi^T(0, 0) \leq \Phi^T(\tilde{S}, \tilde{T}) = \tilde{T})$ and the same argument can be inductively used to show that $V_k \leq \tilde{S}$ and $Z_k \leq \tilde{T}$ for all $K$. Consider now the sequences $S_k, T_k$ as defined in the theorem initialized with $S_0 = T_0 = 0$. By direct substitution we find that $S_1 = W \geq 0 = S_0$ and $T_1 = W \geq 0 = T_0$. By Lemma 6.11(d) follows that the sequences $S_k, T_k$ are monotonically increasing. Moreover, by Lemma 6.11(c) it follows that $(S_k \leq V_k, T_k \leq Z_k) \Rightarrow (S_{k+1} = \Phi^S(S_k, T_k) \leq \Upsilon(S_k, T_k, \tilde{L}) \leq \Upsilon(V_k, Z_k, \tilde{L}) = V_{k+1}), T_{k+1} = \Phi^T(S_k, T_k) \leq \Phi^T(V_k, Z_k) = Z_{k+1})$. Since this is verified for $k = 0$, it inductively follows that $(S_k \leq V_k, T_k \leq Z_k)$ for all $k$. Finally since $V_k, Z_k$ are bounded, we have that $(S_k \leq \tilde{S}, T_k \leq \tilde{T}$. Since $S_k, T_k)$ are monotonically increasing and bounded, it follows that $\lim_{k\to\infty} S_k = S_\infty$ and $\lim_{k\to\infty} T_k = T_\infty$, where $S_\infty, T_\infty$ are semi-definite matrices. From this it easily follows that these matrices have the property $S_\infty = \Phi^S(S_\infty, T_\infty), T_\infty = \Phi^T(S_\infty, T_\infty)$. Definite positiveness of $S_\infty$ follows from Lemma 6.11(e) using the hypothesis that $(A, W^{1/2})$ is observable. The same argument can be used to prove that $T_\infty > 0$. Finally proof of uniqueness of solution and convergence for all initial conditions $S_0, T_0$ can be obtained similarly to Theorem 1 in [**?**] and it is therefore omitted.

(b)⇒(a)
This part follows easily by choosing $\tilde{L} = L^*_{S_\infty, T_\infty}$, where $L^*$ is defined in Lemma 6.11. Using Lemma 6.11(c) we have $S_\infty = \Phi^S(S_\infty, T_\infty) = \Upsilon(S_\infty, T_\infty, \tilde{L})$, therefore the statement is verified using $\tilde{S} = S_\infty$ and $\tilde{T} = T_\infty$.

### Proof of Lemma 6.9

To prove the necessity condition, it is sufficient to show that there exist some initial conditions $S_0, T_0 \geq 0$ for which the sequences $S_{k+1} =$

$\Phi^S(S_k, T_k), T_{k+1} = \Phi^T(S_k, T_k)$ are unbounded, i.e. $\lim_{k\to\infty} S_k = \lim_{k\to\infty} T_k = \infty$. To do so, suppose that at some time-step $k$ we have $S_k \geq s_k vv'$ and $T_k \geq t_k vv'$, where $s_k, t_k > 0$, and $v$ is the eigenvector corresponding to the largest eigenvalue of $A'$, i.e. $A'v = \lambda_{max} v$ and $|\lambda_{max}| = |A'| = |A|$. Then we have:

$$\begin{aligned}
S_{k+1} &= \Phi^S(S_k, T_k) \geq \Phi^S(s_k vv', t_k vv') \\
&= \min_L \Upsilon(s_k vv', t_k vv', L) \\
&= \min_L \Big( s_k A'vv'A + W + 2 s_k \bar{\nu} A'vv'BL + \\
&\qquad + \bar{\nu} L' \big(U + B'\big((1-\bar{\alpha})s_k vv' + \bar{\alpha} t_k vv'\big)B\big)L\Big) \\
&\geq \min_L \Big( s_k |A|^2 vv' + 2 s_k \bar{\nu} \lambda_{max} vv'BL + \\
&\qquad + \bar{\nu} L'B'\big((1-\bar{\alpha})s_k vv' + \bar{\alpha} t_k vv'\big)BL\Big) \\
&= \min_L \Big( s_k |A|^2 vv' - \frac{|A|^2 \bar{\nu} s_k^2}{\xi_k} vv' + \\
&\quad + \bar{\nu}\xi_k (\lambda_{max} s_k^2 I + \frac{1}{\xi_k} BL)' vv' (\lambda_{max} s_k^2 I + \frac{1}{\xi_k} BL)\Big) \\
&\geq s_k |A|^2 vv' - \frac{|A|^2 \bar{\nu} s_k^2}{(1-\bar{\alpha})s_k + \bar{\alpha} t_k} vv' \\
&= |A|^2 s_k \Big(1 - \frac{\bar{\nu} s_k}{(1-\bar{\alpha})s_k + \bar{\alpha} t_k}\Big) vv' \\
&= s_{k+1} vv'
\end{aligned}$$

where $I$ is the identity matrix and $\xi_k = (1-\bar{\alpha})s_k + \bar{\alpha} t_k$. Similarly we have:

$$\begin{aligned}
T_{k+1} &= \Phi^T(S_k, T_k) \geq \Phi^T(s_k vv', t_k vv') \\
&= (1-\bar{\gamma}) t_k A'vv'A + \bar{\gamma} s_k A'vv'A + W \\
&\geq (1-\bar{\gamma}) t_k |A^2| vv' + \bar{\gamma} s_k |A|^2 vv' \\
&= |A|^2 \big((1-\bar{\gamma}) t_k + \bar{\gamma} s_k\big) vv' \\
&= t_{k+1} vv'
\end{aligned}$$

We can summarize the previous results as follows:

$$\begin{aligned}
&(S_k \geq s_k vv', T_k \geq t_k vv') \Rightarrow (S_{k+1} \geq s_{k+1} vv', T_{k+1} \geq t_{k+1} vv') \\
&s_{k+1} = \phi^s(s_k, t_k) = |A|^2 s_k \Big(1 - \frac{\bar{\nu} s_k}{(1-\bar{\alpha})s_k + \bar{\alpha} t_k}\Big), \\
&t_{k+1} = \phi^t(s_k, t_k) = |A|^2 \big((1-\bar{\gamma}) t_k + \bar{\gamma} s_k\big)
\end{aligned}$$

Let us define the following sequences:

$$\begin{aligned} S_{k+1} &= \Phi^S(S_k, T_k), \quad T_{k+1} = \Phi^T(S_k, T_k), \quad S_0 = T_0 = vv' \\ s_{k+1} &= \phi^s(s_k, t_k), \quad t_{k+1} = \phi^t(s_k, t_k), \quad s_0 = t_0 = 1 \\ \tilde{S}_k &= s_k vv', \qquad \tilde{T}_k = t_k vv' \end{aligned}$$

From the previous derivations, we have that $S_k \geq \tilde{S}_k, T_k \geq \tilde{T}_k$ for all time $k$. Therefore, it is sufficient to find when the scalar sequences $s_k, t_k$ diverges to find the necessary conditions. It should be evident also that the operators $\phi^s(s,t), \phi^t(s,t)$ are monotonic in their arguments. Also, it should be clear that the only fixed points of $s = \phi^s(s,t), t = \phi^t(s,t)$ are $s = t = 0$. Therefore we should find when the origin is an unstable equilibrium point, since in this case $\lim_{k\to\infty} s_k, t_k = \infty$. Note that $t = \phi^t(s,t)$ can be written as:

$$\begin{aligned} t &= \Phi^T(s,t) = (1-\bar{\gamma})|A|^2 t + \bar{\gamma}|A|^2 s \\ &= \psi(s) = \frac{\bar{\gamma}|A|^2 s}{1-(1-\bar{\gamma})|A|^2} \end{aligned}$$

with the additional assumption $1-(1-\bar{\gamma})A^2 > 0$. A necessary condition for the stability of the origin is that the origin of restricted map $z_{k+1} = \phi(z_k, \psi(z_k))$ is stable. The restricted map is given by:

$$\begin{aligned} z_{k+1} &= |A|^2 z_k \left(1 - \bar{\nu}\frac{z_k}{(1-\bar{\alpha})z_k + \bar{\alpha}\frac{\bar{\gamma}|A|^2}{1-(1-\bar{\gamma})A^2} z_k}\right) \\ &= |A|^2 \left(1 - \frac{\bar{\nu}}{(1-\bar{\alpha}) + \bar{\alpha}\frac{\bar{\gamma}|A|^2}{1-(1-\bar{\gamma})A^2}}\right) z_k \\ &= |A|^2 \left(1 - \frac{\bar{\nu}(1-(1-\bar{\gamma})|A|^2)}{\bar{\gamma}+\bar{\nu}-\bar{\gamma}\bar{\nu}-\bar{\nu}(1-\bar{\gamma})|A|^2}\right) z_k \\ &= \left(\frac{\bar{\gamma}(1-\bar{\nu})|A|^2}{\bar{\gamma}+\bar{\nu}-\bar{\gamma}\bar{\nu}-\bar{\nu}(1-\bar{\gamma})|A|^2}\right) z_k \end{aligned}$$

This is a linear map and it is stable only if the term inside the parenthesis is smaller than unity, i.e.

$$\begin{aligned} \left(\frac{\bar{\gamma}(1-\bar{\nu})|A|^2}{\bar{\gamma}+\bar{\nu}-\bar{\gamma}\bar{\nu}-\bar{\nu}(1-\bar{\gamma})|A|^2}\right) &< 1 \\ \bar{\gamma}(1-\bar{\nu})|A|^2 &< \bar{\gamma}+\bar{\nu}-\bar{\gamma}\bar{\nu}-\bar{\nu}(1-\bar{\gamma})|A|^2 \\ |A|^2(\bar{\gamma}+\bar{\nu}-2\bar{\gamma}\bar{\nu}) &< \bar{\gamma}+\bar{\nu}-\bar{\gamma}\bar{\nu} \end{aligned}$$

which concludes the lemma.

### Proof of Lemma 6.10

The proof is constructive. In fact, we find a control feedback gain $\tilde{L}$ that satisfies the conditions stated in Theorem 6.11(a). Let $\tilde{L} = -\eta B^{-1}A$ where $\eta > 0$ is a positive scalar that is to be determined. Also consider $S = sI, T = tI$, where $I$ is the identity matrix and $s, t > 0$ are positive scalars. Then we have

$$\begin{aligned}
\Upsilon(sI, tI, \tilde{L}) &= A'sA + W - 2\bar{\nu}\eta A'sA + \bar{\nu}A'B^{-'}UB^{-1}A + \\
&\quad +\bar{\nu}\eta^2 A'\big((1-\bar{\alpha})s + \bar{\alpha}t\big)A \\
&\leq |A|^2\Big(s - 2\bar{\nu}s\eta + \bar{\nu}\big((1-\bar{\alpha})s + \bar{\alpha}t\big)\eta^2\Big)I + wI \\
&= \varphi^s(s, t, \eta)I && (6.84)\\
\Phi^T(sI, tI) &= \bar{\gamma}A'sA + (1-\bar{\gamma})A'tA + W \\
&\leq \big(\bar{\gamma}|A|^2 s + (1-\bar{\gamma})|A|^2 t\big)I + wI \\
&\leq \varphi^t(s, t)I && (6.85)
\end{aligned}$$

where $w = |W + \bar{\nu}A'B^{-'}UB^{-1}A| > 0$ and $I$ is the identity matrix. Let us consider the following scalar operators and sequences:

$$\begin{aligned}
\varphi^s(s, t, \eta) &= |A|^2(1 - 2\bar{\nu}\eta + \bar{\nu}(1-\bar{\alpha})\eta^2)s + \bar{\nu}\bar{\alpha}\eta^2 t + w \\
\varphi^t(s, t) &= \bar{\gamma}|A|^2 s + (1-\bar{\gamma})|A|^2 t + w \\
s_{k+1} &= \varphi^s(s_k, t_k, \eta), \quad t_{k+1} = \varphi^t(s_k, t_k), \quad s_0 = t_0 = 0
\end{aligned}$$

The operators are clearly monotonically increasing in $s, t$, and since $s_1 = \varphi^s(s_0, t_0, \eta) = w \geq s_0$ and $t_1 = \varphi^t(s_0, t_0) = w \geq t_0$, it follows that the sequences $s_k, t_k$ are monotonically increasing. If these sequences are bounded, then they must converge to $\tilde{s}, \tilde{t}$. Therefore $s_k, t_k$ are bounded if and only if there exist $\tilde{s}, \tilde{t} > 0$ such that $\tilde{s} = \varphi^s(\tilde{s}, \tilde{t}, \eta)$ and $\tilde{t} = \varphi^t(\tilde{s}, \tilde{t})$. Let us find the fixed points:

$$\begin{aligned}
\tilde{t} &= \varphi^t(\tilde{s}, \tilde{t}) \Rightarrow \\
\tilde{t} &= \frac{\bar{\gamma}|A|^2}{1 - (1-\bar{\gamma})|A|^2}\tilde{s} + w_t
\end{aligned}$$

where $w_t \overset{\Delta}{=} \frac{w}{1-(1-\bar{\gamma})|A|^2} > 0$, and we must have $1-(1-\bar{\gamma})|A|^2 > 0$ to guarantee that $\tilde{t} > 0$. Substituting back into the operator $\varphi^s$ we have:

$$\begin{aligned}
\tilde{s} &= |A|^2(1-2\bar{\nu}\eta+\bar{\nu}(1-\bar{\alpha})\eta^2)\tilde{s}+\bar{\nu}\bar{\alpha}\eta^2\frac{\bar{\gamma}|A|^2}{1-(1-\bar{\gamma})|A|^2}\tilde{s}+ \\
&\quad +\bar{\nu}\bar{\alpha}\eta^2 w_t + w \\
&= |A|^2\left(1-2\bar{\nu}\eta+\bar{\nu}\Big((1-\bar{\alpha})+\frac{\bar{\gamma}\bar{\alpha}|A|^2}{1-(1-\bar{\gamma})|A|^2}\Big)\eta^2\right)\tilde{s}+w(\eta) \\
&= |A|^2\left(1-2\bar{\nu}\eta+\bar{\nu}\frac{\bar{\gamma}+\bar{\nu}-\bar{\gamma}\bar{\nu}-\bar{\nu}(1-\bar{\gamma})|A|^2}{1-(1-\bar{\gamma})|A|^2}\eta^2\right)\tilde{s}+w(\eta) \\
&= a(\eta)\tilde{s}+w(\eta)
\end{aligned}$$

where $w(\eta) \overset{\Delta}{=} \bar{\nu}\bar{\alpha}\eta^2 w_t + w > 0$. For a positive solution $\tilde{s}$ to exist, we must have $a(\eta) < 1$. Since $a(\eta)$ is a convex function of the free parameter $\eta$, we can try to increase the basin of existence of solutions by choosing $\eta^* = \operatorname{argmin}_\eta a(\eta)$, which can be found by solving $\frac{da}{d\eta}(\eta^*) = 0$ and is given by:

$$\eta^* = \frac{1-(1-\bar{\gamma})|A|^2}{\bar{\gamma}+\bar{\nu}-\bar{\gamma}\bar{\nu}-\bar{\nu}(1-\bar{\gamma})|A|^2}$$

Therefore a sufficient condition for existence of solutions are given by:

$$\begin{aligned}
a(\eta^*) &< 1 \\
|A|^2\left(1-\bar{\nu}\frac{1-(1-\bar{\gamma})|A|^2}{\bar{\gamma}+\bar{\nu}-\bar{\gamma}\bar{\nu}-\bar{\nu}(1-\bar{\gamma})|A|^2}\right) &< 1 \\
\left(\frac{\bar{\gamma}(1-\bar{\nu})|A|^2}{\bar{\gamma}+\bar{\nu}-\bar{\gamma}\bar{\nu}-\bar{\nu}(1-\bar{\gamma})|A|^2}\right) &< 1
\end{aligned}$$

which is the same bound for the necessary condition of convergence in Lemma 6.9.

If this condition is satisfied then $\lim_{k\to\infty} s_k = \tilde{s}$ and $\lim_{k\to\infty} t_k = \tilde{t}$. Let us consider now the sequences $\bar{S}_k = s_k I$, $\bar{T}_k = t_k I$, $S_{k+1} = \Upsilon(S_k, T_k, \tilde{L})$ and $T_{k+1} = \Phi^T(S_k, T_k)$, where $\tilde{L} = -\eta^* B^{-1}A$, $S_0 = T_0 = 0$, and $s_k, t_k$ where defined above. These sequences are all monotonically increasing. From Equations (6.84) and (6.85) it follows that $(S_k \le s_k I, T_k \le t_k I) \Rightarrow (S_{k+1} = \le s_{k+1}I, T_{k+1} \le t_k I)$. Since this is verified for $k = 0$ we can claim that $S_k < \tilde{s}I$ and $T_k < \tilde{t}I$ for all $k$. Since $S_k, T_k$ are monotonically increasing and bounded, then they must converge to positive semi-definite matrices $\tilde{S}, \tilde{T} \ge 0$ which solve the equations $\tilde{S} = \Upsilon(\tilde{S}, \tilde{T}, \tilde{L})$ and $\tilde{T} = \Phi^T(\tilde{S}, \tilde{T})$. Since, by hypothesis, the pair $(A, W^{1/2})$ is observable, using similar arguments of Lemma 6.11(e), it is possible to show that $\tilde{S}, \tilde{T} > 0$. Therefore $\tilde{S}, \tilde{T}, \tilde{L}$ satisfy the conditions of statement (a) Theorem 6.11, from which if follows statement (b) of the same theorem. This implies that the sufficient conditions derived here guarantee the claim of the lemma.

**Bruno** The text below is from twoblock.tex, which was written by Vijay. Integrate as appropriate. You might want to leave this in its own file, to make revisions between authors easier for subversion to handle.

## 6.15 Introduction

In the second part of the chapter, we will consider the two block design with an analog erasure channel inside the control loop. As discussed earlier, the analog erasure model (also referred to as the packet erasure or packet loss model) can be described as follows. The channel operates in discrete time steps. At every time step, the channel accepts as input a finite dimensional real vector $r(k)$. The value of the output of the channel $y(k)$ is chosen according to an *erasure process*. At every time step, the erasure process assumes either the value $T$ or the value $R$. If the value at time $k$ is $T$, $y(k+1) = r(k)$ and a successful transmission is said to have occurred. Otherwise, $y(k+1) = \phi$ and an erasure event, or a packet loss, is said to have occurred at time $k$. The symbol $\phi$ denotes that the receiver does not receive any data; however, the receiver is aware that an erasure event has occurred at that time. Note that we have assumed that the channel introduces a constant delay of one time step.

While an analog erasure model has an infinite capacity in an information theoretic sense, it is often a useful representation for the cases when the communication protocols allow for large data packets to be transmitted at every time step. For instance, the minimum size of an ethernet data packet is 72 bytes. This is much more space for carrying information than usually required inside a control loop. If the data packets allow for transmission of control and sensing data to a high fidelity, the quantization effects are often ignored and an analog erasure model adopted.

To begin with, consider a set-up in which the linear time invariant process evolves as

$$x(k+1) = Ax(k) + Bu(k) + w(k),$$

where the state $x(k) \in \mathbf{R}^n$, the control variable $u(k) \in \mathbf{R}^m$ and $w(k)$ is process noise considered to be white and Gaussian with zero mean and covariance $R_w > 0$. The initial condition $x(0)$ is assumed to be Gaussian with zero mean and covariance $P(0)$. The process is observed using a sensor of the form

$$y(k) = Cx(k) + v(k),$$

where $v(k)$ is measurement noise that is again white Gaussian with zero mean and covariance $R_v > 0$. We suppose all the sources of randomness in the system (initial condition, process and measurement noise, and the erasure process for the channels) to be independent. The inputs are chosen to minimize the cost

$$J_{LQG} = E\left[\sum_{k=1}^{K}\big(x^T(k)Qx(k) + u^T(k)Ru(k)\big) + x^T(K+1)P(K+1)x(K+1)\right],$$

where the expectation at time $k$ is taken with respect to the future values of the packet erasure events, the initial condition, and the measurement and process noises. Further, the matrices $P(K+1)$, $Q$ and $R$ are all assumed to be positive definite. The pair $(A, B)$ is assumed to be controllable, and the pair $(A, C)$ is assumed to be observable.

As with the one-block design, we can consider two cases:

1. When there is only one channel in the control loop, present between the sensor and the controller. Such a situation can arise, e.g., when the controller is co-located with the process and the sensor is remote, or the controller has access to large transmission power.

2. When there are two channels present. In addition to the sensor-controller channel, there is an additional channel between the controller and the actuator. In this case, it is also important to specify the action that the actuator takes when it does not receive a packet. The action depends on the amount of processing, memory and information about the process that is assumed to be available at the actuator. We consider the simplest choice, which is to apply zero control input if no packet was received. Other actions by the actuator can be treated in a similar fashion.

For most of the following discussion we assume that the erasures on the two channels occur in an i.i.d. fashion, independently of each other. However, erasures according to a Markov model, or correlated across the channels can be easily considered.

### Two block design

As discussed earlier, two block design involves designing both an encoder at the input of a channel, and a decoder at the output to minimize the estimation / control cost. Note that the decoder for the sensor-controller channel and the encoder for the controller-actuator channel are merely the controller.

To begin with, we will consider only the sensor-controller channel to be present. To proceed, we must define the class of encoders that we will consider. The information theoretic capacity of an analog erasure channel is infinite. Thus, the only constraints we impose on the encoder are that the transmitted vector is some causal (possibly time-varying) function of the measurements available to the encoder until time $k$ and that the dimension of the vector is finite. We will sometimes refer to the encoder as an encoding algorithm. For the controller-actuator channel, the choice of decoder will depend on the knowledge and processing available at the actuator. We will consider the case of this channel being present briefly towards the end.

At every time step $k$, the encoder at the sensor calculates a vector $s(k) = f\left(k, \{y(j)\}_{j=0}^{k}\right)$ and transmits it. Note that we have not assumed that the encoder has access to any acknowledgements from the decoder about which transmissions have been successful. However, we will show that the presence of such acknowledgements does not improve the optimal performance achievable by a suitable encoder.

Denote by $\mathcal{I}(k)$ the information set that the decoder can utilize to calculate the control at time $k$. As an example, if no erasures were happening, $\mathcal{I}(k) = \{y(0), y(1), \cdots, y(k-1)\}$. More generally, given any packet erasure pattern, we can define a time stamp $t_s(k)$ at every time step $k$ such that the erasures did not allow any information transmitted by the encoder after time $t_s(k)$ to reach the decoder. Without loss of generality, we can restrict attention to information-set feedback controllers. For a given information set $\mathcal{I}(.)$, denote the minimal value of the cost $J_{LQG}$ that can be achieved with the optimal controller design by $J^{\star}_{LQG}(\mathcal{I})$, and the smallest sigma algebra generated by the information set as $\mathbf{I}(.)$. If two information sets $I^1(.)$ and $I^2(.)$ are such that $\mathbf{I}^1(k) \subseteq \mathbf{I}^2(k)$, we have $J^{\star}_{LQG}(\mathcal{I}^2) \leq J^{\star}_{LQG}(\mathcal{I}^1)$.

Consider an algorithm $\mathcal{A}_1$ in which at every time step $k$, the sensor transmits all measurements $y(0)$, $y(1)$, $\cdots$, $y(k)$ to the decoder. Note that this algorithm is not a valid encoding algorithm since the dimension of the transmitted vector is not bounded, as $k$ increases. However, with this algorithm, for any drop sequence, the decoder has access to an information set of the form $\mathcal{I}^{\max}(k) = \{y(0), y(1), \cdots, y(t_s(k))\}$, where $t_s(k) \leq k-1$ is the time stamp defined above. This is the maximal information set that the decoder can have access to with any algorithm, in the sense that $\mathbf{I}(k) \subseteq \mathbf{I}^{\max}(k)$, for any other algorithm that yields the information set $\mathcal{I}(k)$. Thus, one way to achieve the optimal value of the cost function is to utilize an algorithm that makes $\mathcal{I}^{\max}(k)$ available to the sensor at every time $k$ along with a controller that optimally utilizes this set. Further, one such encoder algorithm is $\mathcal{A}_1$. However, as discussed above, $\mathcal{A}_1$ is not a valid encoding algorithm. Surprisingly, as shown below, we can achieve the same performance with an algorithm that transmits a vector with finite dimension.

We begin with the following separation principle when the decoder has access to the maximal information set. Denote by $\hat{\alpha}(k|\beta(k))$ the minimum mean squared error (MMSE) estimate of the random variable $\alpha(k)$ based on the information $\beta(k)$.

**Theorem 6.13** (Separation Principle with Maximal Information Set)**.** *Consider the control problem as defined above, when the decoder has access to the maximal information set $\mathcal{I}^{\max}(k)$ at every time step. Then, the optimal* *control input is given by*

$$u(k) = \hat{u}_{LQ}\left(k|\mathcal{I}^{\max}(k), \{u(j)\}_{j=0}^{k-1}\right),$$

*where $u_{LQ}(k)$ is the optimal LQ control law.*

The proof of this result is similar to the standard separation principle (see, e.g., [**?**, Chapter 9]) and is omitted here. For our setting, the importance of this result lies in the fact that it recognizes that $\hat{u}_{LQ}\left(k|\mathcal{I}^{\max}(k), \{u(j)\}_{j=0}^{k-1}\right)$ (or, in turn, $\hat{x}_{LQ}\left(k|\mathcal{I}^{\max}(k), \{u(j)\}_{j=0}^{k-1}\right)$) is a sufficient statistic to calculate the control input that achieves the minimum possible cost for *any* encoding algorithm. Utilizing the fact that the optimal MMSE estimate of $x(k)$ is linear in the effects of the maximal information set and the previous control inputs, we can identify the quantity that the encoder should transmit that depends only on the measurements. We have the following result.

**Theorem 6.14** (Separation of the Effect of the Control Inputs)**.** *The quantity $\hat{x}_{LQ}\left(k|\mathcal{I}^{\max}(k), \{u(j)\}_{j=0}^{k-1}\right)$ can be calculated as*

$$\hat{x}_{LQ}\left(k|\mathcal{I}^{\max}(k), \{u(j)\}_{j=0}^{k-1}\right) = \bar{x}_{LQ}\left(k|\mathcal{I}^{\max}(k)\right) + \psi(k),$$

*where $\bar{x}_{LQ}\left(k|\mathcal{I}^{\max}(k)\right)$ depends only on $\mathcal{I}^{\max}(k)$ but not on the control inputs and $\psi(k)$ depends only on the control inputs $\{u(j)\}_{j=0}^{k-1}$. Further both $\bar{x}_{LQ}\left(k|\mathcal{I}^{\max}(k)\right)$ and $\psi(k)$ can be calculated recursively.*

*Proof.* The proof follows readily from noting that $\hat{x}_{LQ}\left(k|\mathcal{I}^{\max}(k), \{u(j)\}_{j=0}^{k-1}\right)$ can be obtained from the Kalman filter which is affine in both measurements and control inputs. We can identify

$$\begin{aligned}
\bar{x}_{LQ}\left(k|\mathcal{I}^{\max}(k)\right) &= A^{k-t_s(k)-1}\breve{x}(t_s(k)+1|t_s(k)) \\
\psi(k) &= A^{k-t_s(k)-1}\breve{\psi}(t_s(k)+1) + \sum_{i=0}^{k-t_s(k)-2} A^i B u(k-i-1),
\end{aligned}$$

where $\breve{x}(j+1|j)$ evolves as

$$\begin{aligned}
M^{-1}(j|j) &= M^{-1}(j|j-1) + C^T R_v^{-1} C \\
M^{-1}(j|j)\breve{x}(j|j) &= M^{-1}(j|j-1)\breve{x}(j|j-1) + C^T R_v^{-1} y(j) \\
M(j|j-1) &= AM(j-1|j-1)A^T + R_w \\
\breve{x}(j|j-1) &= A\breve{x}(j-1|j-1),
\end{aligned}$$

with the initial conditions $\breve{x}(0|-1) = 0$ and $M(0|-1) = \Pi(0)$, and $\breve{\psi}(j)$ evolves as

$$\begin{aligned}
\breve{\psi}(j) &= Bu(j-1) + \Gamma(j-1)\breve{\psi}(j-1) \\
\Gamma(j) &= AM^{-1}(j-1|j-1)M(j-1|j-2),
\end{aligned}$$

with the initial condition $\breve{\psi}(0) = 0$. □

Now consider the following algorithm $\mathcal{A}_2$. At every time step $k$, the encoder calculates and transmits the quantity $\breve{x}(k|k)$ using the algorithm in the above proof. The decoder calculates the quantity $\psi(k)$. If the transmission is successful, the decoder calculates

$$\begin{aligned}\hat{x}_{LQ}\left(k+1|\mathcal{I}^{\max}(k+1), \{u(j)\}_{j=0}^{k}\right) &= \bar{x}_{LQ}\left(k+1|\mathcal{I}^{\max}(k+1)\right) + \psi(k) \\ &= A\breve{x}(k|k) + \psi(k).\end{aligned}$$

If the transmission is unsuccessful, the decoder calculates

$$\hat{x}_{LQ}\left(k+1|\mathcal{I}^{\max}(k+1), \{u(j)\}_{j=0}^{k}\right) = A^{k-t_s(k)}\bar{x}_{LQ}\left(k+1|\mathcal{I}^{\max}(t_s(k)+1)\right)+\psi(k),$$

where the quantity $\bar{x}_{LQ}\left(k+1|\mathcal{I}^{\max}(t_s(k)+1)\right)$ is stored in the memory from the last successful transmission (note that only the last successful transmission needs to be stored). Using the Theorems 6.13 and 6.14 clearly allows us to state the following result.

**Theorem 6.15** (Optimality of the Algorithm $\mathcal{A}_2$)**.** *Algorithm $\mathcal{A}_2$ is optimal in the sense that it allows the controller to calculate the control input $u(k)$ that minimizes $J_{LQG}$.*

*Proof.* At every time step, the algorithm $\mathcal{A}_2$ makes $\hat{x}_{LQ}\left(k+1|\mathcal{I}^{\max}(k+1), \{u(j)\}_{j=0}^{k}\right)$ available to the controller. Thus, the controller can calculate the same control input as with the algorithm $\mathcal{A}_1$ which together with an LQ controller yields the minimum value of $J_{LQG}$. □

Note that the optimal algorithm is non-linear (in particular, it is a switched linear system). This is not unexpected, in view of the non-classical information pattern in the problem.

**Remarks**

• *Boundedness of the Transmitted Quantity:* It should be emphasized that the quantity $\breve{x}(k|k)$ that the encoder transmits is not the estimate of $x(k)$ (or the state of some hypothetical open loop process) based only on the measurements $y(0), \cdots, y(k)$. In particular under the constraint on the erasure probability that we derive later, the state $x(k)$ is stable and hence the measurements $y(k)$ are bounded. *Thus, the quantity $\breve{x}(k|k)$ is bounded.* This can also be seen from the recursive filter used in the proof of Theorem 6.14. If the closed loop system $x(k)$ is unstable due to high erasure probabilities, $\breve{x}(k|k)$ would, of course, not be bounded. However, the optimality result implies that the system cannot be stabilized by transmitting any other bounded quantity (such as measurements).

• *Optimality for any Erasure Pattern and the 'Washing Away' Effect:* The optimality of the algorithm required no assumption about the erasure statistics. The optimality result holds for an arbitrary erasure sequence, and at every time step (not merely in an average sense). Moreover, any successful transmission 'washes away' the effect of the previous erasures in the sense that it ensures that the control input is identical to the case as if all previous transmissions were successful.

• *Presence of Delays:* We assumed that the communication channel introduces a constant delay of one time step. However, the same algorithm continues to remain optimal even if the channel introduces larger (or even time-varying) delays, as long as there is the provision of a time stamp from the encoder regarding the time it transmits any vector. The decoder uses the packet it receives at any time step only if it was transmitted later than the quantity it has stored from the previous time steps. If this is not true due to packet re-ordering, the decoder continues to use the quantity stored from previous time steps. Further, if the delays are finite, the stability conditions derived below remain unchanged. Infinite delays are equivalent to packet erasures, and can be handled by using the same framework.

**Stability and Performance:** Both the stability and performance of the system with this optimal coding algorithm in place can be analyzed by assuming specific models for the erasure process. For pedagogical ease, we adopt the i.i.d. erasure model, with an erasure occurring at any time step with probability $p$. Due to the separation principle, to obtain the stability conditions, we need to consider the conditions under which the LQ control cost for the system, and the covariance of the estimation error between the state of the process $x(k)$ and the estimate at the controller $\hat{x}(k)$ remain bounded, as time $k$ increases. Under the controllability and observability assumptions the LQ cost remains bounded, if the control value does. Define the estimation error and its covariance as

$$\begin{aligned} e(k) &= x(k) - \hat{x}(k) \\ P(k) &= E\left[e(k)e^T(k)\right], \end{aligned}$$

where the expectation is taken with respect to the process and measurement noises, and the initial condition (but not the erasure process). Due to the 'washing away' effect of the algorithm, the error of the estimate at the decoder evolves as

$$e(k+1) = \begin{cases} \bar{e}(k+1) & \text{no erasure} \\ Ae(k) & \text{erasure event,} \end{cases}$$

where $\bar{e}(k)$ is the error between $x(k)$ and the estimate of $x(k)$ given all control inputs $\{u(j)\}_{j=0}^{k-1}$ and measurements $\{y(j)\}_{j=0}^{k-1}$. Thus, the error covariance evolves as

$$P(k+1) = \begin{cases} M(k+1) & \text{with probability } 1-p \\ AP(k)A^T + R_w & \text{with probability } p, \end{cases}$$

where $M(k)$ is the covariance of the error $\bar{e}(k)$. Thus, we obtain

$$E[P(k+1)] = (1-p)M(k+1) + pR_w + pAE[P(k)]A^T,$$

where the extra expectation for the error covariance is taken over the erasure process in the channel. Since the system is observable, $M(k)$ converges exponentially to a steady state value $M^\star$. Thus, the necessary and sufficient condition for the convergence of the above discrete algebraic Lyapunov recursion is

$$p\rho(A)^2 < 1,$$

where $\rho(A)$ is the spectral radius of $A$. Due to the optimality of the algorithm considered above, this condition is necessary for stability of the system with any causal encoding algorithm. In particular, for the strategy of simply transmitting the latest measurement from the sensor as considered in the one block design, this condition turns out to be necessary for stability (though not sufficient for a general process model). For achieving stability with this condition, we require an encoding strategy, such as the recursive algorithm provided above.

This analysis can be generalized to more general erasure models. For example, for a Gilbert-Eliot type channel model, the necessary and sufficient condition for stability is given by

$$q_{00}\rho(A)^2 < 1,$$

where $q_{00}$ is the conditional probability of an erasure event at time $k+1$, provided an erasure occurs at time $k$. In addition, by calculating the terms $E[P(k)]$ and the LQ control cost of the system with full state information, the performance $J_{LQG}$ can also be calculated through the separation principle proved above. The value of the cost function thus achieved provides a lower bound to the value of the cost function achievable using any other encoding or control algorithm, for the same probability of erasure. An alternative viewpoint is to consider the encoding algorithm above as a means for transmitting data with lesser frequency to achieve the same level of performance, than, e.g., transmitting measurements to the controller.

**Higher Order Moments:** It can be seen that the treatment above can be extended to consider the stability of higher order moments of the estimation error, or the state value. In fact, the entire steady state probability distribution function of the estimation error can be calculated.

### Extensions and Open Questions

The above framework was explained for a very simple set-up of an LQG problem. It is natural to consider its generalization to other models by removing various assumptions. We consider some of these assumptions below. We also point out some of the open questions.

- *Channel between Controller and Actuator:* The encoding algorithm presented above continues to remain optimal when a channel is present between the controller and the actuator (as considered in Figure **??**), as long as there is a provision for acknowledgement from actuator to controller for any successful transmission, and the protocol that the actuator follows in case of an erasure is known at the controller. This is because these two assumptions are enough for the separation principle to hold. If no such acknowledgement is available, the control input begins to have a dual effect and the optimal algorithm is still unknown. Moreover, the problem of designing the optimal encoder for the controller-actuator channel can also be considered. This design will intimately depend on the information that is assumed to be known at the actuator (e.g., the cost function, the system matrices and so on). Algorithms that optimize the cost function for such information sets are largely unknown. A simpler version of the problem would involve either

- analyzing the stability and performance gains for given encoding and decoding algorithms employed by the controller and the actuator respectively, or,
- considering algorithms that are stability optimal, in the sense of designing recursive algorithms that achieve the largest stability region for any possible causal encoding algorithm.

Both these directions have seen research activity. For the first direction, algorithms typically involve transmitting some future control inputs at every time step, or the actuator using some linear combination of past control inputs if an erasure occurs. The second direction has identified the stability conditions that are necessary for any causal algorithm. Moreover, recursive designs that can achieve stability when these conditions are satisfied have also been identified. Surprisingly, the design is in the form of a *universal actuator* that does not require access to the model of the plant. Even if such knowledge were available, the stability conditions do not change. Thus, the design is stability optimal.

• *Presence of a Communication Network:* So far we have concentrated on the case when the sensor and the controller are connected using a single communication channel. A typical scenario, particularly in a wireless context, would instead involve a communication network with multiple such channels. If no encoding algorithm is implemented, and every node in the network (including the sensor) transmits simply the measurements, the network can be replaced by a giant erasure channel with the equivalent erasure probability being some measure of the reliability of the network. However, the performance degrades rapidly as the network size increases. If encoding is permitted, such an equivalence breaks down. The optimal algorithm is an extension of the single channel case, and is provided in [**?**]. The stability and performance calculations are considerably more involved. However, the stability condition has an interesting interpretation in terms of the capacity for fluid networks. The necessary and sufficient condition for stability can be expressed as the inequality

$$p_{\text{max-cut}}\rho(A)^2 < 1,$$

where $p_{\text{max-cut}}$ is the max-cut probability calculated in a manner similar to the min-cut capacity of fluid networks. We construct cut-sets by dividing the nodes in the network into two sets with one set containing the sensor, and the other the controller. For each cut-set, we calculate the cut-set erasure probability by multiplying the erasure probabilities of all the channels from the set containing the sensor to the set containing the controller. The maximum such cut-set erasure probability (over all possible cut-sets) denotes the max-cut probability of the network. The improvement in the performance and stability region of the system by using the encoding algorithm increases drastically with the size and the complexity of the network.

• *Multiple Sensors:* Another direction in which the above framework can be extended is to consider multiple sensors observing the same process. As with the case with one sensor, one can identify the necessary stability conditions and a lower bound for the achievable cost function with any causal coding algorithm. These stability conditions are also sufficient and recursive algorithms for achieving stability when these conditions are satisfied have been identified. These conditions are a natural extension of the stability conditions for the single sensor case. As an example, for the case of two sensors described by sensing matrices $C_1$ and $C_2$ that transmit data to the controller across erasure channels for which erasure events are i.i.d. with probabilities $p_1$ and $p_2$ respectively, the stability conditions are given by the set

$$\begin{aligned} p_2\rho(A_1)^2 &< 1 \\ p_1\rho(A_2)^2 &< 1 \\ p_1p_2\rho(A)^2 &< 1, \end{aligned}$$

where $\rho(A_i)$ denotes the spectral radius of the unobservable part of the system matrix $A$, when the pair $(A, C_i)$ is represented in the observability canonical form. However, the problem of identifying distributed encoding algorithms to be followed at each sensor for achieving the lower bounds on the achieved cost function remains largely open. This problem is related to the track fusion problem that considers identifying algorithms for optimal fusion of information from multiple sensors that interact intermittently (e.g., see [**?**]). That transmitting estimates based on local data from each sensor is not optimal is long known. While algorithms that achieve a performance close to the lower bound of the cost function have been identified, a complete solution is not available.

• *Inclusion of More Communication Effects:* Our discussion has focussed on modeling the loss of data transmitted over the channel. In our discussion of the optimal encoding algorithms, we also briefly considered the possibility of data being delayed or received out of order. An important direction for future work is to consider other effects due to communication channels. Both from a theoretical perspective, and for many applications such as underwater systems, an important effect is to impose a limit on the number of bits that can be communicated for every successful transmission. Some recent work [**?**, **?**] has considered the analog digital channel in which the channel supports $n$ bits per time step and transmits them with a certain probability $p$ at every time step. Stability conditions for such a channel have been identified and are a natural combination of the stability conditions for the analog erasure channel above and the ones for a noiseless digital channel, as considered elsewhere in the book. The performance of optimal encoding algorithms and the optimal performance that is achievable remain unknown. Another channel effect that has largely been ignored is the addition of channel noise to the data received successfully.

- *More General Performance Criteria:* Our treatment focussed on a particular performance measure - a quadratic cost, and the stability notions emanating from that measure. Other cost functions may be relevant in applications. Thus the cost function may be related to target tracking, measures such as $H_2$ or $H_\infty$ [**?**], or some combination of communication and control costs. The analysis and optimal encoding algorithms for such measures are expected to differ significantly. An an example, for target tracking, the properties of the reference signal that needs to be tracked can be expected to play a significant role. Similarly, for $H_\infty$ related costs, the sufficient statistic, and hence the encoding algorithms to transmit it, may be vastly different than the LQG case. Finally, a distributed control problem with multiple processes, sensors and actuators is a natural direction to consider.

- *More General Plant Dynamics:* The final direction is to consider plant dynamics that are more general than the linear model that we have considered. Moving to models such as jump linear systems, hybrid systems, and general non-linear systems will provide new challenges and results. As an example, for non-linear plants concepts such as spectral radius no longer hold. Thus, the analysis techniques are likely to be different and measures such as Lyapunov exponents and the Lipschitz constant for the dynamics will likely become important.

Please send references that were cited, preferably in bibtex format. Bibitems are commented out in source file. **Vijay**

# Chapter 7
## Information Flow and Consensus

In this chapter we move from the problem of estimation and control of a single system across a communications channel to the challenge of sensing, estimation and control of a multi-agent system, with the information available to the agents represented by a graph of interconnections. We begin with a review of the relevant concepts in graph theory, focused on the use of algebraic techniques to characterize the properties of the interconnection structure. We then apply these concepts to study the problem of a group of agents reaching consensus on a shared property of the system.

The contents of this chapter are currently based on slides from the EECI course, which were generated using some notes from Reza's course at Caltech. Need to go through and make sure that I am not directly making use of any of his material. **RMM**

## 7.1 Graph Theory

In this section we give a brief overview of the field of graph theory, focused on some of the algebraic methods that characterize the properties of the graph in terms of a set of matrices associated with it. These techniques will be very important for helping understand the interactions between dynamic agents across a graph, including the consensus problem in this chapter and the distributed estimation and control problems in the subsequent chapters. More detailed treatments are available in a number of textbooks, including Diestel [**?**], Godsil and Royle [**?**], and Horn and Johnson [**?**]. This section is based in part on a set of course notes originally developed by Reza Olfati-Saber [**?**].†

**RMM**: Need to run this section by Reza and make sure he is OK with the contents.

### Basic Definitions

We define a *directed graph* as a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a set of *vertices* $\mathcal{V}$ and a set of *edges* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We represent a vertex (or *node*) as an element $v_i \in \mathcal{V}$ and an edge (or *link*) as a connection between two vertices, $e_{ij} = (v_i, v_j) \in \mathcal{E}$. We write $|\mathcal{V}|$ for the number of vertices in the graph, also known as the *order* of the graph. qAn edge has an orientation given by the ordering of the vertices, so the edge $e_{ij}$ is distinct from the edge $e_{ji}$. We call $v_i$ the head of the edge and $v_j$ the tail. A directed graph is also referred to

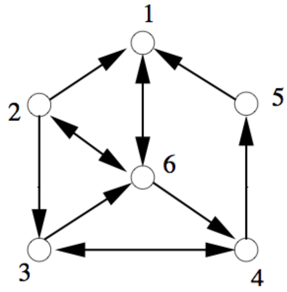


**Figure 7.1:** A graph with 6 vertices.

as a *digraph.*

We say that two vertices $v_i$ and $v_j$ are *adjacent* if there exists and edge $e = (v_i, v_j)$ and vertex $v_j$ is called a *neighbor* of $v_i$. We write $\mathcal{N}_i =$ set of all neighbors of $v_i$ and we say that a graph $\mathcal{G}$ is *complete* if all vertices are adjacent to each other. We define the *out-degree* of a vertex $v_i$, written $\deg_{\text{out}}(V_i)$, as the number of edges whose head is $v_i$. Similarly, the *in-degree* of a vertex $v_i$, $\deg_{\text{in}}(v_i)$ is the number of edges with tail $v_i$.

**Example 7.1 6 node graph**
Consider a graph given by the vertices $\mathcal{V} = \{1, 2, 3, 4, 5, 6\}$ and a set of edges

$$\begin{aligned}\mathcal{E} = \{&(1,6),(2,1),(2,3),(2,6),(6,2),(3,4),\\ &(3,6),(4,3),(4,5),(5,1),(6,1),(6,2),(6,4)\},\end{aligned}$$

as shown in Figure 7.1. Node 1 has an in-degree of 3 and an out-degree of 1. Its neighbor set is given by $\mathcal{N}_1 = \{v_6\}$. Node 2 has an in-degree of 1 and an out-degree of 3. Its neighbor set is given by $\mathcal{N}_2 = \{v_1, v_2, v_6\}$. $\nabla$

In many instances the orientation of the edges in a graph will not matter and we can ignore the ordering of the verticies in an edge $e_{ij} = (v_i, v_j)$. Formally, we will consider a graph to be *undirected* if $e_{ij} \in \mathcal{E}$ implies that $e_{ji} \in \mathcal{E}$. In these cases it will often be easier to simply say that the graph is undirected and consider an edge $e_{ij}$ to be equivalent to an edge $e_{ji}$. For an undirected graph the indegree and outdegree are the same, so we simply refer to the *degree of a vertex.* An undirected graph is *regular* (or *k-regular*) if all vertices of a graph have the same degree $k$. A directed graph is *balanced* if the out-degree is equal to the in-degree at each vertex.

**Example 7.2**
Figure 7.3 shows three examples of graphs. $\nabla$

## Connectedness of Graphs

A key set of properties of a graph have to do with whether there exists paths that connect its nodes. Formally, a *path* is a subgraph $\pi = (\mathcal{V}, \mathcal{E}_\pi) \subset \mathcal{G}$ with

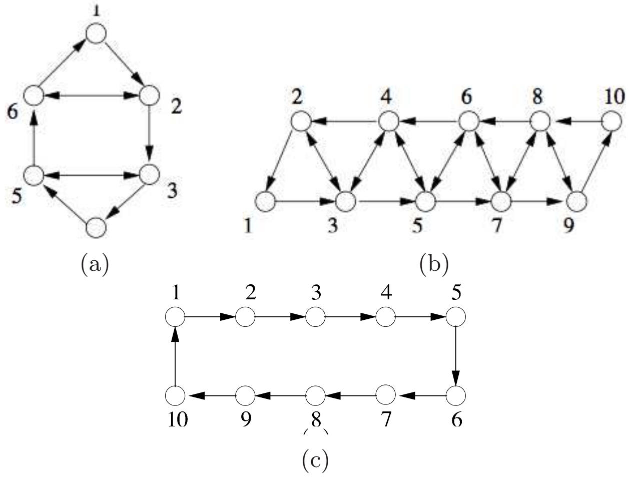


**Figure 7.2:** Examples of graphs with different properties.

distinct vertices $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$ and

$$\mathcal{E}_\pi := \{(v_1, v_2), (v_2, v_3), \ldots, (v_{m-1}, v_m)\}.$$

The *length* of $\pi$ is defined as $|\mathcal{E}_\pi| = m-1$. A *cycle* (or $m$-cycle) $C = (\mathcal{V}, \mathcal{E}_C)$ is a path (of length $m$) with an extra edge $(v_m, v_1) \in \mathcal{E}$. We define the *distance* between two vertices $v$ and $w$ as the length of the shortest path between them.

An undirected graph $\mathcal{G}$ is called *connected* if there exists a path $\pi$ between any two distinct vertices of $\mathcal{G}$. For a connected graph $\mathcal{G}$, the length of the maximum distance between two vertices is called the *diameter* of $\mathcal{G}$. A graph with no cycles is called *acyclic*. A *tree* is a connected acyclic graph.

A digraph is called *strongly connected* if there exists a directed path $\pi$ between any two distinct vertices of $\mathcal{G}$. A digraph is called *weakly connected* if there exists an undirected path between any two distinct vertices of $\mathcal{G}$.

**Example 7.3**
Figure **??** shows examples of graphs and their connectedness properties. $\nabla$

## Matrices Associated with a Graph

In order to characterize the properties of a graph, we will use matrices to represent the structure of the graph. The properties of these matrices can then be related back to the properties of the graph.

The *adjacency matrix* $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ of a graph $\mathcal{G}$ of order $n$ is given by:

$$a_{ij} := \begin{cases} 1 & \text{if } (v_i, v_j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

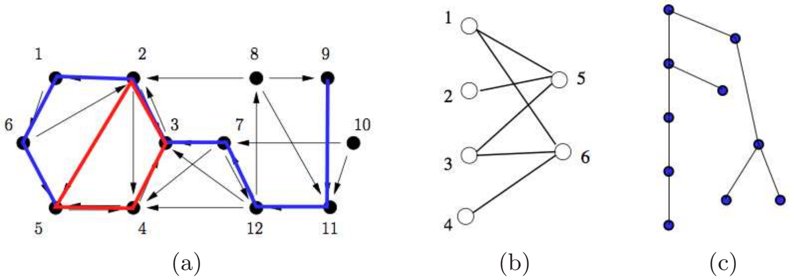


**Figure 7.3:** Examples of graphs with different properties.

The *degree matrix* of a graph as a diagonal $n \times n$ ($n = |\mathcal{V}|$) matrix

$$\Delta = \text{diag}\{\deg_{\text{out}}(v_i)\}$$

with diagonal elements equal to the out-degree of each vertex and zero everywhere else. The *Laplacian matrix* $L$ of a graph is defined as

$$L = \Delta - A$$

. It follows from the definition that the row sums of the Laplacian are all 0.

**Example 7.4 6 node graph**
Consider the graph shown in Example **??**. The adjancy matrix and Laplacian are given by

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}, \qquad L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ -1 & 3 & -1 & 0 & 0 & -1 \\ 0 & 0 & 2 & -1 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & -1 & 0 & -1 & 0 & 3 \end{bmatrix}.$$

$\nabla$

### Periodic Graphics and Weighted Graphs

A graph with the property that the set of all cycle lengths has a common divisor $k > 1$ is called *k-periodic*. A graph without cycles is said to be *acyclic*.

A *weighted graph* is graph $(\mathcal{V}, \mathcal{E})$ together with a map $\varphi : \mathcal{E} \to \mathbb{R}$ that assigns a real number $w_{ij} = \varphi(e_{ij})$ called a *weight* to an edge $e_{ij} = (v_i, v_j) \in \mathcal{E}$. The set of all weights associated with $\mathcal{E}$ is denoted by $\mathcal{W}$. A weighted graph can be represented as a triplet $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$.

In some applications it is natural to "normalize" the Laplacian by the outdegree. We define the *weighted Laplacian* as

$$\tilde{L} := \Delta^{-1} L = I - \tilde{A} = I - \Delta^{-1} A$$

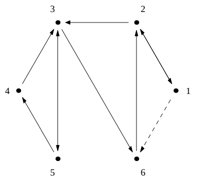


**Figure 7.4:** Formation control graph.

, where $\tilde{A} = \Delta^{-1} A$ (weighted adjacency matrix).

**Example 7.5 Weighted Laplacian for formation graph**
Consider the graph in Figure 7.4. The weighted Laplacian is given by

$$L = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$\nabla$

## Gershgorin Disk Theorem

Add some explanatory text indicating what we are going to use all of this for. **RMM**

**Theorem 7.1** (Gershgorin Disk Theorem)**.** *Let $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ and define the deleted absolute row sums of $A$ as*

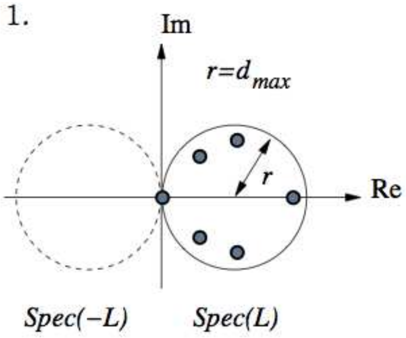
$$r_i := \sum_{j=1, j\neq i}^{n} |a_{ij}| \tag{7.1}$$

*Then all the eigenvalues of $A$ are located in the union of $n$ disks*

$$G(A) := \bigcup_{i=1}^{n} G_i(A), \ \textit{with} \quad G_i(A) := \{z \in \mathbb{C} : |z - a_{ii}| \leq r_i\} \tag{7.2}$$

*Furthermore, if a union of $k$ of these $n$ disks forms a connected region that is disjoint from all the remaining $n-k$ disks, then there are precisely $k$ eigenvalues of $A$ in this region.*

*Sketch of proof.* Let $\lambda$ be an eigenvalue of $A$ and let $v$ be a corresponding eigenvector. Choose $i$ such that $|v_i| = \max_j |v_j > 0$. Since $v$ is an eigenvec-



**RMM**: Figure is currently not referenced in the text

**Figure 7.5:** Graphical description of the Gershgorin disk theorem.

tor,

$$\lambda v_i = \sum_i A_{ij} v_j \quad \Longrightarrow \quad (\lambda - a_{ii}) v_i = \sum_{i \neq j} A_{ij} v_j$$

Now divide by $v_i \neq 0$ and take the absolute value to obtain

$$|\lambda - a_{ii}| = |\sum_{j \neq i} a_{ij} v_j| \leq \sum_{j \neq i} |a_{ij}| = r_i$$

□

We can use the Gershgorin disk theorem to reason about the eigenvalues of the Laplacian and the weighted Laplacian.

**Proposition 7.2.** *Let $L$ be the Laplacian matrix of a digraph $\mathcal{G}$ with maximum vertex out–degree of $d_{max} > 0$. Then all the eigenvalues of $A = -L$ are located in a disk*

$$B(\mathcal{G}) := \{s \in \mathbb{C} : |s + d_{max}| \leq d_{max}\} \tag{7.3}$$

*that is located in the closed LHP of $s$-plane and is tangent to the imaginary axis at $s = 0$.*

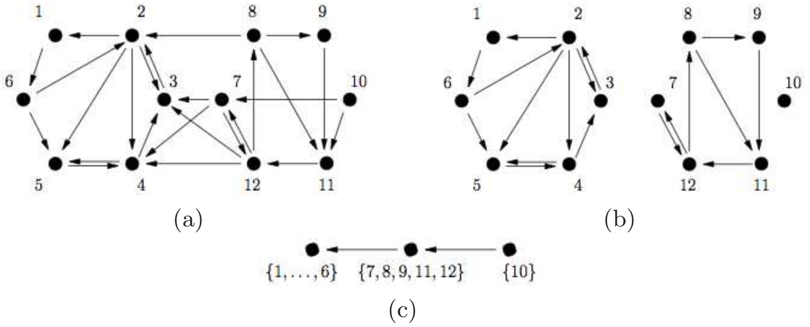
**Proposition 7.3.** *Let $\tilde{L}$ be the weighted Laplacian matrix of a digraph $\mathcal{G}$. Then all the eigenvalues of $A = -L$ are located inside a disk of radius 1 that is located in the closed LHP of $s$-plane and is tangent to the imaginary axis at $s = 0$.*

Another property of the Laplacian is that its rank determines the connectivity of the graph.

**Theorem 7.4** (Olfati-Saber)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ be a weighted digraph of order $n$ with Laplacian $L$. If $\mathcal{G}$ is strongly connected, then rank$(L) = n - 1$.*

**RMM**: check

The proof for the directed case can be found in standard textbooks on graph theory, such as those listed at the beginning of this section.†. The



(a)

(b)

(c)

**Figure 7.6:** Irreducibility of a graph.

proof for the undirected case is available in [OSM04]. Note that for directed graphs, we require that $\mathcal{G}$ be strongly connected; the converse statement is not true.

### Perron-Frobenius Theory

The spectrum of a matrix $A$ is defined as $\text{spec}(A) = \{\lambda_1, \ldots, \lambda_n\}$, where $\lambda_i$, $i = 1, \ldots, n$ are the eigenvalues of $A$. The distance to the largest eigenvalue $\rho(A) = |\lambda_n| = \max_k |\lambda_k|$ is called the *spectral radius* of $A$.

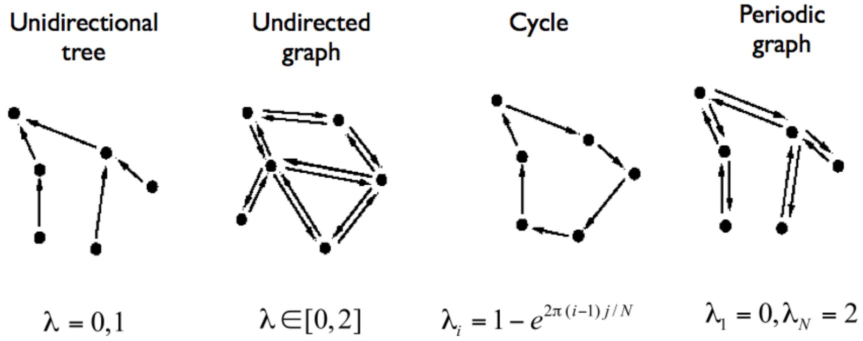
**Theorem 7.5** (Perron's Theorem, 1907)**.** *If $A \in \mathbb{R}^{n\times n}$ is a positive matrix ($A > 0$), then*

1. *$\rho(A) > 0$;*
2. *$r = \rho(A)$ is an eigenvalue of $A$;*
3. *There exists a positive vector $x > 0$ such that $Ax = \rho(A)x$;*
4. *$|\lambda| < \rho(A)$ for every eigenvalue $\lambda \neq \rho(A)$ of $A$, i.e. $\rho(A)$ is the unique eigenvalue of maximum modulus; and*
5. *$[\rho(A)^{-1}A]^m \to R$ as $m \to +\infty$ where $R = xy^T$, $Ax = \rho(A)x$, $A^T y = \rho(A)y$, $x > 0$, $y > 0$, and $x^T y = 1$.*

**Theorem 7.6** (Perron's Theorem for Non–Negative Matrices)**.** *If $A \in \mathbb{R}^{n\times n}$ is a non-negative matrix ($A \geq 0$), then $\rho(A)$ is an eigenvalue of $A$ and there is a non–negative vector $x \geq 0$, $x \neq 0$, such that $Ax = \rho(A)x$.*

A directed graph is irreducible if, given any two vertices, there exists a path from the first vertex to the second. (Irreducible = strongly connected) A matrix is irreducible if it is not similar to a block upper triangular matrix via a permutation. A digraph is irreducible if and only if its adjacency matrix is irreducible.

**Example 7.6**
Consider the graph in Figure 7.6.



**RMM**: Figure is currently not referenced

**Figure 7.7:** Spectra of the Laplacian for classes of graphs.

**RMM** Complete example

$\nabla$

**Theorem 7.7** (Frobenius)**.** *Let $A \in \mathbb{R}^{n\times n}$ and suppose that $A$ is irreducible and non-negative. Then*

1. *$\rho(A) > 0$;*
2. *$r = \rho(A)$ is an eigenvalue of $A$;*
3. *There is a positive vector $x > 0$ such that $Ax = \rho(A)x$;*
4. *$r = \rho(A)$ is an algebraically simple eigenvalue of $A$; and*
5. *If $A$ has $h$ eigenvalues of modulus $r$, then these eigenvalues are all distinct roots of $\lambda^h - r^h = 0$.*

Using the Perron and Frobenius theorems, we can study additional properties of the Laplacian matrix of a graph. In particular, it can be shown that If $\mathcal{G}$ is strongly connected, the zero eigenvalue of $L$ is simple. If $\mathcal{G}$ is aperiodic, all nonzero eigenvalues lie in the interior of the Gershgorin disk. If $\mathcal{G}$ is $k$-periodic, $L$ has $k$ evenly spaced eigenvalues on the boundary of the Gershgorin disk.

**Theorem 7.8** (Variant of Courant-Fischer)**.** *Let $A \in \mathbb{R}^{n\times n}$ be a Hermitian matrix with eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and let $w_1$ be the eigenvector of $A$ associated with the eigenvalue $\lambda_1$. Then*

$$\lambda_2 = \min_{\substack{x \neq 0,\, x \in \mathbb{C}^n, \\ x \perp w_1}} \frac{x^* A x}{x^* x} = \min_{\substack{x^* x = 1, \\ x \perp w_1}} x^* A x \tag{7.4}$$

*Proof.* Since $A$ is Hermitian matrix, it is unitary diagonalizable (see Theorem **??**), i.e. $A = U \Lambda U^*$ where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. Let $U =$

$[w_1|w_2|\cdots|w_n]$ ($w_k$ is the $k$th column of $U$). Then

$$\begin{aligned} x^*Ax &= x^*U\Lambda U^*x = (U^*x)^*\Lambda(U^*x) \\ &= \sum_{i=1}^{n}\lambda_i|(U^*x)_i|^2 = \sum_{i=1}^{n}\lambda_i|w_i^*x|^2 = \sum_{i=2}^{n}\lambda_i|w_i^*x|^2 \quad (x\perp w_1) \\ &\geq \lambda_2\sum_{i=2}^{n}|w_i^*x|^2 = \lambda_2\sum_{i=1}^{n}|w_i^*x|^2 \quad (x\perp w_1) = \lambda_2\sum_{i=1}^{n}|(U^*x)_i|^2 \\ &= \lambda_2(x^*UU^*x) = \lambda_2 x^*x \end{aligned} \tag{7.5}$$

Thus, for $x\perp w_1$ and $x \neq 0$

$$x^*Ax \geq \lambda_2 x^*x$$

where the equality is achieved with $x = w_2$. □

The second eigenvalue of the Laplacian $\lambda_2$ is called the *algebraic connectivity* of $L$.

### Cyclically Separable Graphs

**Definition 7.1** (Cyclic separability)**.** A digraph $\mathcal{G} = (\mathcal{V},\mathcal{E})$ is *cyclically separable* if and only if there exists a partition of the set of edges $\mathcal{E} = \cup_{k=1}^{n_c}\mathcal{E}_k$ such that each partition $\mathcal{E}_k$ corresponds to either the edges of a cycle of the graph, or a pair of directed edges $ij$ and $ji$ that constitute an undirected edge. A graph that is not cyclically separable is called *cyclically inseparable.*

**Lemma 7.1.** *Let $L$ be the Laplacian matrix of a cyclically separable digraph $\mathcal{G}$ and set $u = -Lx, x \in \mathbb{R}^n$. Then $\sum_{i=1}^{n} u_i = 0, \forall x \in \mathbb{R}^n$ and $\mathbf{1} = (1,\ldots,1)^T$ is the left eigenvector of $L$.*

*Proof.* The proof follows from the fact that by definition of cyclic separability. We have

$$-\sum_{i=1}^{n} u_i = \sum_{ij\in\mathcal{E}}(x_j - x_i) = \sum_{k=1}^{n_c}\sum_{ij\in\mathcal{E}_k}(x_j - x_i) = 0$$

because the inner sum is zero over the edges of cycles and undirected edges of the graph. □

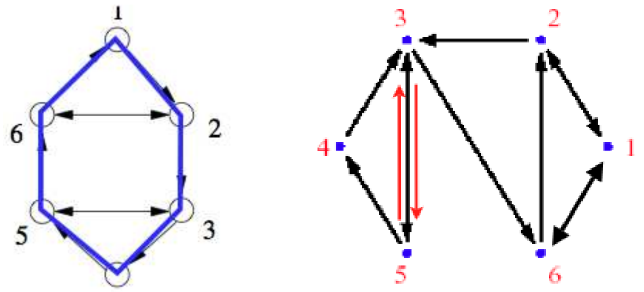
**Example 7.7 Cyclic separability**

∇

Let $\mathcal{G} = (\mathcal{V},\mathcal{E})$ be a digraph. We say $\mathcal{G}$ is *balanced* if and only if the in–degree and out–degree of all vertices of $\mathcal{G}$ are equal, i.e.

$$\deg_{\text{out}}(v_i) = \deg_{\text{in}}(v_i), \quad \forall v_i \in \mathcal{V} \tag{7.6}$$

Let $\mathcal{G}$ be a digraph with a weighted adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ that has the property $a_{ii} = 0$. Then, $\mathcal{G}$ is balanced if and only if $w_l = \mathbf{1}$.



(a) (b)

**Figure 7.8:** Cyclic separability.

**Theorem 7.9.** *A digraph is cyclically separable if and only if it is balanced.*

*Proof.* Assume the graph is cyclically separable. Then any arbitrary vertex $v_i$ of the graph belongs to a finite number of cycles and/or undirected edges. The main property of a cycle is that corresponding to any directed edge arriving at a vertex, there is one edge leaving that vertex and therefore the in–degree and out-degree of any vertex are equal, i.e. the graph is balanced.

Now, let us assume that the graph is balanced, we show that it is cyclically separable. Suppose the opposite holds, meaning that the graph is not cyclically separable. Then there exists a directed edge $(v_k, v_l)$ of the graph which does not belong to any cycles and/or undirected edges. Set $x_i = 0, \forall i \neq l$ and let $x_l = 1$. Define $u = -Lx$, we have $u_i = 0, \forall i \neq k$ and $u_k = x_l - x_k = 1$ (notice that $u_l = 0$ since $k$ is not an out–neighbor of $l$). Thus $\sum_{i=1}^{n} u_i = \mathbf{1}^T u = 1 \neq 0$. But we know that $\mathbf{1}$ eigenvector of $L$ for any balanced graph, thus $\mathbf{1}^T u = -\mathbf{1}^T Lx = 0, \forall x$. This is a contradiction which means every directed edge of a balanced graph belongs to a cycle or an undirected edge, i.e. the graph is cyclically separable. □ □

## 7.2 Consensus algorithms

The *consensus problem* involves a group of agents reaching an agreement on a decision in a decentralized problem. In this sectoin we describe one approach to solving this problem, with the agents communicating on a graph.†

**RMM**: Rewrite

### Average Consensus

Consider a collection of $N$ agents that communicate along a set of undirected links described by a graph $\mathcal{G}$. Each agent has a state $x_i$ with initial value $x_i(0)$ and together they wish to determine the average of the initial states $\text{Ave}(x_0) = 1/N \sum x_0^i$.†

**RMM**: Ave not defined

The agents implement the following consensus algorithm:

$$x^i_{k+1} = \epsilon \sum_{j \in \mathcal{N}_i} (x^j_k - x^i_k) = -\epsilon |\mathcal{N}_i| (x^i_k - \text{Ave}(x^{\mathcal{N}_i}_k))$$

which is equivalent to the dynamical system

$$x_{k+1} == -\epsilon L x_k.$$

**Proposition 7.10.** *If the graph is connected, there exists an $\epsilon$ such that the state of the agents converges to $x^*_i = Ave(x_0)$ exponentially fast.*

- Proposition 1 implies that the spectra of $L$ controls the stability (and convergence) of the consensus protocol.
- To (partially) prove this theorem, we need to show that the eigenvalues of $L$ are all positive.

$$\dot{x} = -Lx \qquad L = \Delta - A$$

Note first that the subspaced spanned by $\mathbf{1} = (1, 1, \ldots, 1)^T$ is an invariant subspace since $L \cdot \mathbf{1} = 0$ Assume that there are no other eigenvectors with eigenvalue 0. Hence it suffices to look at the convergence on the complementary subspace $\mathbf{1}^\perp$.

Let $\delta$ be the disagreement vector

$$\delta = x - \text{Ave}(x(0))\,\mathbf{1}$$

and take the square of the norm of $\delta$ as a Lyapunov function candidate, i.e. define

$$V(\delta) = \|\delta\|^2 = \delta^T \delta \tag{7.7}$$

Differentiating $V(\delta)$ along the solution of $\dot{\delta} = -L\delta$, we obtain

$$\dot{V}(\delta) = -2\delta^T L \delta < 0, \quad \forall \delta \neq 0, \tag{7.8}$$

where we have used the fact that $\mathcal{G}$ is connected and hence has only 1 zero eigenvalue (along $\mathbf{1}$). Thus, $\delta = 0$ is globally asymptotically stable and $\delta \to 0$ as $t \to +\infty$, i.e. $x^* = \lim_{t \to +\infty} x(t) = \alpha_0 \mathbf{1}$ because $\alpha(t) = \alpha_0 = \text{Ave}(x(0)), \forall t > 0$. In other words, the average–consensus is globally asymptotically achieved. □

For an undirected graph with Laplacian $L$, the rate of convergence for the consensus protocol is bounded by the second smallest eigenvalue $\lambda_2$

**Corollary 7.10.1.** *Consider a network of integrators with a directed information flow $\mathcal{G}$ and vertices that apply the consensus protocol. Then, $\alpha = Ave(x)$ is an invariant quantity if and only if $\mathcal{G}$ is balanced.*

*Remarks.*

- Balanced graphs generalized undirected graphs and retain many key properties

### Consensus on Directed Graphs

**RMM** Talk through the case where the graph is directed. This includes balanced graphs, for which we recover the directed case, but also non-balanced graphs, where we reach a consensus but the value is not the average. Can also talk here about using different link weights, though can't talk about the effect on rate since that is not covered until the next section.

### Consensus over Communication Channels

**RMM** The plan for this subsection is to talk about modifications to the basic consensus algorithm that take into account packet losses, rate limits and delays. Need to look through the literature to make sure we get the right basic results here to be useful.

### Consensus for Idempotent Functions

**RMM** Look at extensions of consensus for computing min, max and other idempotent functions. Can also talk about what happens when we get join/rejoin actions, ala Charpentier and Chandy, though this might go better in a later section.

## 7.3 Effects of Information Topology

**RMM** This section will cover some of the effects of the information topology on the consensus problem. Need to think of a better title, though.

Outline:

- Fixed graphs—rates of convergence ($\lambda_2$)
- Nearest neighbor graphs
- Gossip algorithms
- Eigenvalues of the Laplacian (including small word, scale free, etc)

## 7.4 Applications of Consensus

**RMM** This section will cover some of the applications of consensus algorithms.

Outline:

- Distributed computation (Tsitsiklis, PageRank [Ishi and Tempo])

- Flocking
- Load balancing
- Intrusion detection

## 7.5 Further Reading

### Exercises

# Chapter 8
## Distributed Estimation

In this chapter we consider the problem of state estimation in which we have a collection of sensors that are distributed across a network. We begin by exploring the problem of aggregating data from a decentralized network of sensors, either at a centralized node or across a fully connected network, where the goal is to minimize either communication or computation and the graph structure does not play a central role. We then consider more general distributed estimation problem where the graph is not completely connected, so that different agents on the network have different information at different times. We next investigate the case where the information can be lost as it is sent around the network, requiring the use of more advanced methods of design and analysis to accommodate the network dynamics. Finally, we provide some general remarks about when the estimation problem can be separated from the control problem, allow us to separately solve the (optimal) estimation problem. The next chapter looks in more detail at the distributed control problem, where we wish to design a feedback control system across a graph to solve a given task.

### 8.1 Decentralized Sensor Fusion

Note: The goal of this section is going to be to summary the "classical" results in distributed sensor fusion, focused mainly on the information form. [RMM, 19 Jun 09]

Decide how much of the sensor fusion/information filter work currently in Chapter 2 should be put here instead. For now, the material here comes from the OBC book, but we can either toss this material or move the material from Ch 2 up and integrate them.

**RMM**

Consider a single process with multiple sensors connected together across a network, as shown in Figure 8.1. We assume that the system being observed has dynamics

$$x[k+1] = Ax[k] + Bu + w,$$

where $X \in \mathbb{R}^n$† represents the state, $u \in \mathbb{R}^p$ represents the (deterministic) input, $W \in \mathbb{R}^q$ represents process disturbances, $Y \in \mathbb{R}^q$ represents the system output and $W \in \mathbb{R}^q$ represents measurement noise. We would like

**RMM**: Decide whether to be more formally correct here

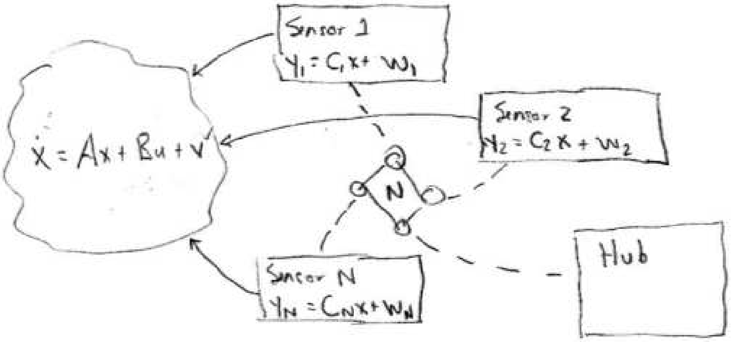


**Figure 8.1:** Schematic diagram of a distributed sensing system. The system on the left represents the system being observed. Multiple sensors take measurements and communication with each other across a communications network. A (optional) centralized hub collections information across the network.

to form an estimation of the state $\hat{x}$, either at each sensor or at the central hub.

**RMM** Introduce some examples here?

- Alice - multiple sensors looking at the environment plus possible need for different information at different points (e.g. urban planning)
- RoboFlag - each robot needs an estimate of the (local) environment plus players need to know the entire centralized hub.

The approach we take to solve this problem depends on the structure of the information pattern. If a centralized hub is available, all sensors can send data to the hub and a centralized Kalman filter can be used to compute the estimate. An alternative, which is more efficient in some settings [**?**], is to have some of the information processing done at the sensor nodes. To see how this can be done, we reformulate the Kalman filter in an alternative form.

**RMM** The text below partially duplicates information contained in Chapter 2. Need to decide what goes where.

Consider the situation described in Figure 8.2, where we have an input/output dynamical system with multiple sensors capable of taking measurements. The problem of sensor fusion involves deciding how to best combine the measurements from the individual sensors in order to accurately estimate the process state $X$. Since different sensors may have different noise characteristics, evidently we should combine the sensors in a way that places more weight on sensors with lower noise. In addition, in some situations we may have different sensors available at different times, so that not all information is available on each measurement update.

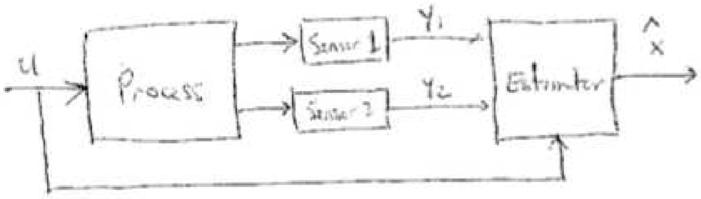


**Figure 8.2:** Sensor fusion

To gain more insight into how the sensor data are combined, we investigate the functional form of $L[k|k]$. Suppose that each sensor takes a measurement of the form

$$Y^i = C^i X + V^i, \qquad i = 1, \dots, q,$$

where the superscript $i$ corresponds to the specific sensor. Let $V^i$ be a zero mean, white noise process with covariance $\sigma_i^2 = R_{V^i}(0)$. It follows from Lemma **??** that

$$L[k|k] = P[k|k] C^T R_W^{-1}.$$

First note that if $P[k|k]$ is small, indicating that our estimate of $X$ is close to the actual value (in the MMSE sense), then $L[k|k]$ will be small due to the leading $P[k|k]$ term. Furthermore, the characteristics of the individual sensors are contained in the different $\sigma_i^2$ terms, which only appears in $R_W$. Expanding the gain matrix, we have

$$L[k|k] = P[k|k] C^T R_W^{-1}, \qquad R_W^{-1} = \begin{bmatrix} 1/\sigma_1^2 & & \\ & \ddots & \\ & & 1/\sigma_q^2 \end{bmatrix}.$$

We see from the form of $R_W^{-1}$ that each sensor is inversely weighted by its covariance. Thus noisy sensors ($\sigma_i^2 \gg 1$) will have a small weight and require averaging over many iterations before their data can affect the state estimate. Conversely, if $\sigma_i^2 \ll 1$, the data is "trusted" and is used with higher weight in each iteration.

An alternative formulation of the Kalman filter is to make use of the inverse of the covariance matrix, called the *information matrix*, to represent the error of the estimate. It turns out that writing the state estimator in this form has several advantages both conceptually and when implementing distributed computations. This form of the Kalman filter is known as the *information filter*.

We begin by defining the information matrix $I$ and the weighted state estimate $\hat{Z}$:

$$I[k|k] = P^{-1}[k|k], \qquad \hat{Z}[k|k] = P^{-1}[k|k] \hat{X}[k|k].$$

**RMM**: Need to sort out subscripts versus superscripts

We also make use of the following quantities, which appear in the Kalman filter equations:

$$\Omega_i[k|k] = (C^i)^T R_{W^i}^{-1}[k|k] C^i, \qquad \Psi_i[k|k] = (C^i)^T R_{W^i}^{-1}[k|k] C^i \hat{X}[k|k].$$

Using these quantities, we can rewrite the Kalman filter equations as

$$\begin{array}{ll} \text{Prediction} & \text{Correction} \\ I[k|k-1] = \left(AI^{-1}[k-1|k-1]A^T + R_W\right)^{-1}, & I[k|k] = I[k|k-1] + \sum_{i=1}^{q} \Omega_i[k|k], \\ \hat{Z}[k|k-1] = I[k|k-1]AI^{-1}[k-1|k-1]\hat{Z}[k-1|k-1] + Bu[k-1], & \hat{Z}[k|k] = \hat{Z}[k|k-1] + \sum_{i=1}^{q} \Psi_i[k|k]. \end{array}$$

**RMM**: These equations need to be checked and also reformatted

Note that these equations are in a particularly simple form, with the information matrix being updated by each sensor's $\Omega_i$ and similarly the state estimate being updated by each sensors $\Psi_i$.

Remarks:

1. Information form allows simple addition for correction step. Intuition: add information through additional data.

2. Sensor fusion: information content = inverse covariance (for each sensor)

3. Variable rate: incorporate new information whenever it arrives. No data $\implies$ no information $\implies$ prediction update only.

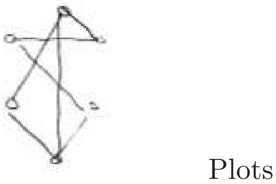
**RMM** Say something here about hierarchical Kalman filtering in the context of the information filter formulation. Basically demonstrate results from Chong using the information filter formulation.

Another classical information pattern is that of a fully connected network. In this case, everyone can send either measurements to each other and we can reconstruct the state using local Kalman (or information) filters.

**RMM** Talk through the computational and communication tradeoffs. This material follows Rao et al and should summarize a number of special cases, including Speyer and Willsky.

## 8.2 Distributed estimation on a graph

A more general case occurs when the information is distributed along a graph, as shown in Figure 8.3. Suppose that we have no central hub and we want each sensor to converge to a single global estimate. For simplicity we



(a) (b)

**Figure 8.3:** Distributed estimation on a graph.

first consider the static system case, in which all nodes should converge to the optimal estimate

$$\hat{x}_\infty = \sum_{j=1}^{N} R_j^{-1} y^j.$$

As our starting point, we make use of the consensus algorithms described in Section **??**. The basic algorithm is as follows:

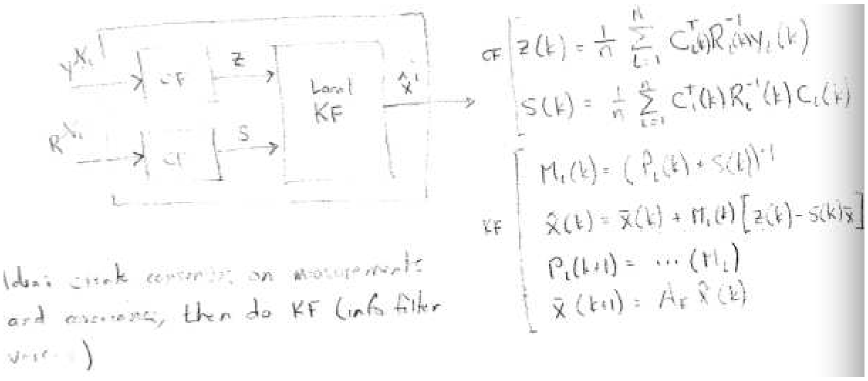
- Each node measures $y^i$
- Each node initializes its state to $x(0) = R_i^{-1} y^i$, where $R_i$ is the covariance associated with sensor $i$
- Run the consensus protocol, which implies that each node converges to the optimal estimate.

From the results on consensus filters, it can be shown that the convergence rate is bounded by $e^{-\lambda_2 t}$, where $\lambda_2$ is the second smallest eigenvalue of the graph Laplacian.†. Many extensions to this basic algorithm are possible, including the case of time-varying communication graphs, delays and intermittent communications.

**RMM**: Think through the directed case

The static algorithm can be extended to the case of a dynamic system in several different ways:

- Fixed graph, ala Durrant-Whyte et al
  - Communication measurements and run full KF
  - Communication local estimate + covariance and account for duplication [**?**]
  - Doesn't handle dropped packets, changing communications graph
- "Microfilter architecture" (Olfati-Saber)



$$z(k) = \frac{1}{n}\sum_{i=1}^{n} C_i^T(k) R_i^{-1}(k) y_i(k)$$

$$S(k) = \frac{1}{n}\sum_{i=1}^{n} C_i^T(k) R_i^{-1}(k) C_i(k)$$

$$M_i(k) = (P_i(k) + S(k))^{-1}$$

$$\hat{x}(k) = \bar{x}(k) + M_i(k)\left[z(k) - S(k)\bar{x}\right]$$

$$P_i(k+1) = \cdots (M_i)$$

$$\bar{x}(k+1) = A_F \hat{x}(k)$$

Idea: track consensus on measurements and covariances, then do KF (info filter version)

- Need CF to convert quickly (compared to KF dynamics) and track measurements
- Resulting filter is *approximate* (may not be optimal prior to convergence), but handles packet delays, etc (inherited from CF properties).
- ??? sending measurements plus covariance matrices (if $R_i(k)$ not constant)

- Consensus on estimates

**Proposition 8.1** (Olfati-Saber)**.** *In the absence of noise, $\hat{x}^i \to x$.*

**RMM** *Proof.* Include sketch

□

Remarks:

1. Can write in discrete time
2. Only approximate KF; lose optimality during transient
3. Can handle varying graph, packet loss, time delay, etc
4. Only requires sensing estimates; $P_i$ is *local* error covariance (doesn't account for neighbor covariance)

Final remarks (on distributed estimation):

1. Distributed KF on a fixed graph (star, completely connected, undirected) is well understood. Basically manipulate the information filter.
2. Alternative approach: use consensus filter on measurements or estimates. Lose optimality, but can handle network effects.

## 8.3 Distributed Estimation with Packet Loss

We now consider what happens if the graph describing the flow of information around the network is not along a fixed graph. We consider a number of cases, starting with the case in which we can only use a subset of the links on a network at a given time, and then moving to more complicated situations in which the network can drop packets in an unknown way.

Outline: **RMM**

- Sensor scheduling with a deterministic pattern
- Multiple sensors, packet erasure (Gupta Phd Sec 3.6?)
- Stochastic sensor scheduling
- Multi-description code? (perhaps as part of broader list of "advanced techniques"?

## 8.4 Combining Estimation and Control

In many applications we wish to make use of our estimate of the state of a system for the purposes of controlling the behavior of the system. In this case, the system state depends on the action of the controller, which itself depends on the estimate of the state. In traditional control systems, it can be shown that a separation principle applies, in which we can design the controller assuming we measure the exact state and design the estimator without taking the specific form of the feedback controller into account. In this section we summarize the situations in which we can similar separation principles in distributed estimation principles. We defer the analysis and design of the distributed controllers to the next chapter.

Write up summary of when we can get a separation principle, following results from Vijay's thesis. **RMM**

## 8.5 Further Reading

The early literature on distributed estimation (and control) focused on extending optimal estimation techniques in which the information distribution network was either completely connected [**?**, **?**] or hierarchical [**?**]. A key question was how to incorporate data taken from a number of sensors into either a centralized node or a (completely connected) set of agents. Much of the early work was focused on the problem of target tracing across a distributed geographical area. A fairly general version of these decentralized estimation results that made use of the information form of the Kalman filter was presented by Rao et al. [**?**].

**RMM** Second paragraph on the more recent literature, including some of our own papers.

# Chapter 9
## Distributed Control

Outline:

- Problem discription: centralized, decentralized, distributed, coupled
- Stability and performance of networked systems (Fax stability criterion + sensitivity analysis by Z. Jin and S. Tonetti)
- (Sub) Optimal distributed control
- Spatially invariant systems (Dulerud, D'Andrea et al + Rotkowitz, Lall)

## 9.1 Introduction

## 9.2 Stability and performance of interconnected systems

## 9.3 Stability of interconnected sytems

Note: Plan to describe the stability conditions from Alex Fax that show how graph topology interacts with dynamics. Other possible things to include:

- Signal flow graphs (useful for computing sensitivity functions, ala Stefania)
- Formula for characteristic equation for the graph Laplacian (from Håkan)

[RMM, 19 Jun 09]

Suppose that each agent's dynamics are governed by

$$\begin{aligned} \dot{x}^i &= Ax^i + Bu^i \\ y^i &= Cx^i \end{aligned} \tag{9.1}$$

Fax [FM04] considers a control law in which each system attempts to stabilize itself relative to its neighbors. This is accomplished by constructing an error for each system that is a weighted combination of the relative outputs of the neighbors:

$$e^i = \sum_{j \in \mathcal{N}^i} \alpha_{ij}(y^j - y^i) \tag{9.2}$$

where $\alpha_{ij}$ is the relative weight. For simplicity, we consider uniform weighting here, so that $\alpha_{ij} = 1/|\mathcal{N}^i|$ where $|\mathcal{N}^i|$ is the number of neighbors of node $i$. The results are easily extended to the more general case.

Given the error (10.3), we apply a compensator that attempts to stabilize the overall system. For simplicity, we assume here that the stabilizer is given by a constant gain

$$u^i = Ke^i, \tag{9.3}$$

with $K \in \mathbb{R}^{m\times m}$ representing the compensation (gain) matrix. In practice, one can use a dynamic compensator to improve performance, but for analysis purposes we can just assume these dynamics are included in the system dynamics (10.2).

The interconnectedness of the system, represented by the neighbor sets $\mathcal{N}_i$ can be studied using tools from graph theory. In particular, for the the case of uniform weighting of the errors, it turns out that the combined error vector $e \in \mathbb{R}^{N\cdot m}$ can be written as

$$e = (\bar{L} \otimes I)x \tag{9.4}$$

where $\otimes$ represents the Kronecker product and $\bar{L}$ is the *weighted Laplacian* associated with the (directed) graph that models the neighbors of each node. The weighted Laplacian is a standard object in graph theory and can be defined as

$$\bar{L} = D^{-1}(D - A)$$

where $D$ is a diagonal matrix whose entries are the out-degree of each node and $A$ is the adjacency matrix for the graph (see [FM04] for more detail). Using this framework, Fax showed the following:

**Theorem 9.1.** *A local controller $K$ stabilizes the formation dynamics in equation (10.2) with error (10.5) and gain $K$ if and only if it stabilizes the set of $N$ systems given by*

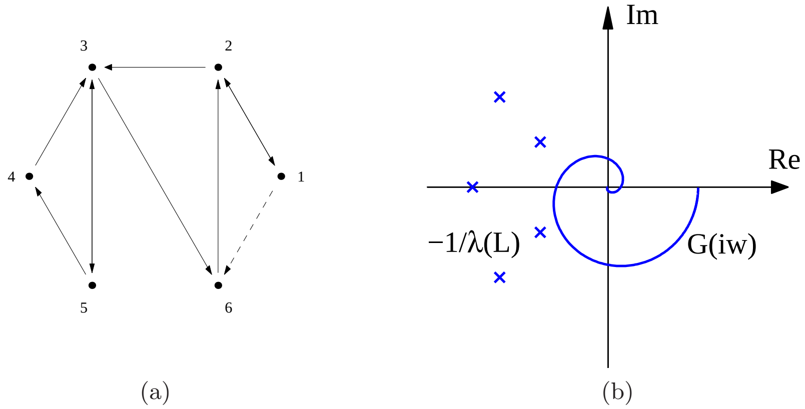
$$\begin{aligned} \dot{x} &= Ax + B\cdot\lambda_i\cdot(Ky) \\ y &= Cx \end{aligned} \tag{9.5}$$

*where $\{\lambda_i\}$ are the eigenvalues of the weighted graph Laplacian $\bar{L}$.*

*Proof.* We make use of the following notational conventions:

- $\widehat{A} = I_N \otimes A$: block diagonal matrix with $A$ as elements
- $A_{(n)} = A \otimes I_n$: replace elemnts of $A$ with $a_{ij}I_n$
- For $X \in \mathbb{R}^{r\times s}$ and $Y \in \mathbb{R}^{N\times N}$, $\widehat{X}Y_{(s)} = \widehat{Y}X_{(r)}$

Let $T$ be a Schur transformation for $L$, so that $U = T^{-1}LT$ is upper triangular. Transform the (stacked) process states as $\tilde{x} = T_{(n)}x$ and the



(a) (b)

**Figure 9.1:** Interpretation of Theorem 1. The left figure shows the graph representation of the interconnected system and the right figure shows the corresponding Nyquist test. The addition of the dashed line to the graph moves the negative, inverse eigenvalues of $\bar{L}$ from the positions marked by circles to those marked by crosses.

(stacked) controller states as $\tilde{\xi} = T_{(n)}\xi$. The resulting dynamics become

$$\frac{d}{dt}\begin{bmatrix}\tilde{x}\\ \tilde{\xi}\end{bmatrix} = \begin{bmatrix}\widehat{A}+\widehat{B}\widehat{K}\widehat{C}U_{(n)} & \widehat{B}\widehat{H}\\ \widehat{G}\widehat{C}U_{(n)} & F\end{bmatrix}\begin{bmatrix}\tilde{x}\\ \tilde{\xi}\end{bmatrix}.$$

This system is upper triangular, and so stability is determined by the elements on the (block) diagonal:

$$\frac{d}{dt}\begin{bmatrix}\tilde{x}_j\\ \tilde{\xi}_j\end{bmatrix} = \begin{bmatrix}A+BKC\lambda_j & BH\\ GC\lambda_j & F\end{bmatrix}\begin{bmatrix}\tilde{x}\\ \tilde{\xi}\end{bmatrix}.$$

This is equivalent to coupling the process and controller with a gain $\lambda_i$. □

This theorem has a very natural interpretation in terms of the Nyquist plot of dynamical system. In the standard Nyquist criterion, one checks for stability of a feedback system by plotting the open loop frequency response of the system in the complex plane and checking for net encirclements of the $-1$ point. The conditions in Theorem 1 correspond to replacing the $-1$ point with $-1/\lambda_i$ for each eigenvalue $\lambda_i$ of $\bar{L}$. This interpretation is illustrated in Figure 10.15. The results can easily be extended to consider weightings that are nonuniform.

Theorem 10.1 illustrates how the *dynamics* of the system, as represented by equation (10.2), interacts with the *information flow* of the system, as represented by the graph Laplacian. In particular, we see that it is the eigenvalues of the Laplacian that are critical for determining stability of the overall system. Additional results in this framework allow tuning of the information flow (considered as both sensed and communicated signals) to improve the transient response of the system [FM04]. Extensions in a stochastic setting [**?**, OSM04] allow analysis of interconnected systems whose dynamics are not identical and where the graph topology changes over time.

## 9.4 (Sub-) Optimal Control

**RMM** Insert Vijay's EECI writeup here

## 9.5 Spatially Invariant Systems

Supplement

# Chapter 10
## Cooperative Control

This chapter is currently a verbatim copy of material from a journal submission. Need to rewrite or get permission to include RMM

This chapter presents a survey of recent research in cooperative control of multi-vehicle systems, using a common mathematical framework to allow different methods to be described in a unified way. The survey has three primary parts: an overview of current applications of cooperative control, an summary of some of the key technical approaches that have been explored, and a description of some possible future directions for research. Specific technical areas that are discussed include formation control, cooperative tasking, spatio-temporal planning and consensus. This chapter is based in large part on a previously published research survey [?].

### 10.1 Introduction

Research on control of multi-vehicle systems performing cooperative tasks dates back to the late 1980s, initially beginning in the field of mobile robotics (see [?] for a more detailed history). Aided by the development of inexpensive and reliable wireless communications systems, research in this area increased substantially in the 1990s. California's Partners for Advanced Transit and Highways (PATH) project [?] demonstrated multiple automobiles driving together in "platoons" and this was quickly followed by other highway automation projects [?, ?]. In the late 1990s and early 2000s, cooperative control of multiple aircraft, especially unmanned aerial vehicles (UAVs), became a highly active research area in the United States [CPR01], spurring further advances. Over the last decade this research area has blossomed, with many new systems being proposed in application areas ranging from military battle systems to mobile sensors networks to commercial highway and air transportation systems. Some of the specific challenges of cooperative control of multi-vehicle systems include uncertainty caused by inter-vehicle communications and distributed operation, system complexity due to the problem size and coupling between tasks, and scaleability to a potentially large collection of vehicles.

The purpose of this article is to provide a survey of some of the recent research in cooperative control of multi-vehicle systems. We focus on research in the last two decades, with some historical notes on work before this period. To help focus the topics that are surveyed, we focus exclusively on control of multi-vehicle systems that are working together to complete a shared task. Several other surveys of the literature in cooperative control are available that complement the current paper (see, e.g., [**?**]).

It will be helpful in the sequel to have a clear notion of some terms that will define the object of the survey, in particular a concise definition of "cooperative", which has been used in many different ways by the broad research communities interested in this topic. For the purposes of this survey, we will consider a *vehicle* to be a dynamical system whose position is given by its location in three dimensional space. We will consider a collection of $N$ vehicles that are performing a shared task, where the task depends on the relationship between the locations of the individual vehicles. The vehicles are able to communicate with each other in carrying out the task, with the individual vehicles able to communicate with some subset of the other vehicles.

We assume that the dynamics of the $i$th vehicle can be written as

$$\begin{aligned} \dot{x}^i &= f^i(x^i, u^i) \quad && x^i \in \mathbb{R}^n, u^i \in \mathbb{R}^m \\ \dot{y}^i &= h^i(x^i) && y^i \in SE(3), \end{aligned}$$

where $x^i$ is the state of the $i$th vehicle, $u^i$ is the input that controls the vehicle's state and $f^i$ is a smooth vector field representing its dynamics. We assume that the location of the vehicle is given by the output $y^i \in SE(3)$, where $SE(3)$ is the set of rigid body configurations (position and orientation). More general formulations allowing position and velocity as part of the location description are possible as well, but will be omitted for simplicity. We let $x = (x^1, \dots, x^N)$ represent the complete state for a collection of $N$ vehicles.

In addition to the location of the vehicle, we will also assume that each vehicle has a discrete state, $\alpha^i$, which we define as the *role* of the vehicle. The role of the vehicle will be represented as an element of a discrete set $\mathcal{A}$ whose definition will depend on the specific cooperative control problem under consideration. As indicated by the terminology, we will generally consider the role variable $\alpha^i$ to represent the portion of the vehicle's overall state that encodes its current actions and its relationship with the overall task being performed. We will assume that the role of a vehicle can change at any time and we will write a change of role as

$$\alpha' = r(x, \alpha),$$

where $\alpha'$ indicates the new value of $\alpha$. We let $\alpha = (\alpha^1, \dots, \alpha^N)$ represent the roles of the collection of $N$ vehicles and write $\alpha^i(t)$ for the role of vehicle $i$ at time $t$.

We assume that the vehicles are able to communicate with some set of other vehicles and we represent the set of possible communication channels by a graph $\mathcal{G}$. The nodes of the graph represent the individual vehicles and a directed edge between two nodes represents the ability of a vehicle to receive information from another vehicle. We write $\mathcal{N}^i(\mathcal{G})$ to represent the neighbors of vehicle $i$, that is, the set of vehicles that vehicle $i$ is able to obtain information from (either by explicit communication or by sensing the position of the other vehicle). In general, $\mathcal{N}^i$ can depend on the locations and roles of the vehicles, in which case we will write $\mathcal{N}^i(x, \alpha)$. The number of neighbors of the $i$th vehicle is given by the number of elements of $\mathcal{N}^i$, written $|\mathcal{N}^i|$.

Given a collection of vehicles with state $x$ and roles $\alpha$, we will define a *task* in terms of a performance function

$$J = \int_0^T L(x, \alpha, u)\, dt + V(x(T), \alpha(T)),$$

where $T$ is the horizon time over which the task should be accomplished, $L$ represents the incremental cost of the task and $V$ represents the terminal cost of the task. As special cases, we can take $T = \infty$ to represent infinite horizon problems or take $L = 0$ to represent tasks in which we are only interested in the final state. We may also have constraints on the states or inputs, although we shall generally consider such constraints to be included in the cost function (eg, via Lagrange multipliers) for ease of presentation.

A *strategy* for a given task is an assignment of the inputs $u^i$ for each vehicle and a selection of the roles of the vehicles. We will assume that the inputs to the vehicles' dynamics are given by control laws of the form

$$u^i = \gamma(x, \alpha)$$

where $\gamma$ is a smooth function. For the choice of roles, we make use of the notion of a *guarded command language* (see [KM04]): a program is a set of commands of the form

$$\{g^i_j(x, \alpha) : r^i_j(x, \alpha)\}$$

where $g^i_j$ is a guard that evaluates to either true or false and $r^i_j$ is a rule that defines how the role $\alpha^i$ should be updated if the rule evaluates to true. Thus, the role evolves according to the update law

$$\alpha^{i\,\prime} = \begin{cases} r^i_j(x, \alpha) & g(x, \alpha) = \text{true} \\ \text{unchanged} & \text{otherwise.} \end{cases}$$

This update is allowed to happen asynchronously, although in practice it may be assigned by a central agent in the system, in which case it may evolve in a more regular fashion. We write $\Sigma^i$ to represent the overall strategy (control law and guarded commands) for the $i$th vehicle. $\Sigma = (\Sigma^1, \dots, \Sigma^N)$ is used to represent the complete strategy for the system.

Using these definitions, we can now provide a more formal description of a cooperative control problem. We say that a task can be *additively* *decoupled* (or just decoupled) if the cost function $J$ can be written as

$$J = \sum_{i=0}^{N} \left( \int_0^T L^i(x^i, \alpha^i, u^i)\, dt + V^i(x^i(T), \alpha^i(T)) \right).$$

If a task is not decoupled, we say that the task is *cooperative*, by which we mean that the task performance depends on the joint locations, roles and inputs of the vehicles. (Note that we are assuming here that all vehicles are trying to solve a common objective and hence not considering adversarial tasks, for which a more careful notation would be required.)

We say that a strategy is *centralized* if $\Sigma^i$ depends on the location or role of any vehicle that is not a neighbor of $i$. A strategy is *decentralized* if

$$\begin{aligned} u^i(x, \alpha) &= u^i(x^i, \alpha^i, x^{-i}, \alpha^{-i}) \\ \{g^i_j(x, \alpha) : r^i_j(x, \alpha)\} &= \{g^i_j(x^i, \alpha^i, x^{-i}, \alpha^{-i}) : r^i_j(x^i, \alpha^i, x^{-i}, \alpha^{-i})\}, \end{aligned}$$

where we use the shorthand $x^{-i}$ and $\alpha^{-i}$ to represent the location and roles of vehicle $i$'s neighbors (hence $x^{-i} = \{x^{j_1}, \dots, x^{j_{m_i}}$ where $j_k \in \mathcal{N}^i$ and $m_i = |\mathcal{N}^i|.\}$. We will mainly be interested in cooperative tasks that can be solved using a decentralized strategy.

We note that the definitions used here are not the most general possible and we have ignored some subtleties regarding the formal definition of the "solution" of a task (i.e., we assume existence and uniqueness of solutions for a given strategy). These details are important and can be found in the various papers referenced in this survey. One alternative set of definitions for cooperative agents can be found in the work of Parker [**?**], which makes use of the notions of local/global goals and control.

With these definitions in hand, we now proceed to consider some of the primary applications of cooperative control of multi-vehicle systems, followed by some of the key technical results that have been proposed in the last decade. We end the paper with a partial listing of some of the open research directions that are currently under exploration.

## 10.2 Applications Overview

In this section we summarize some of the main applications for cooperative control of multi-vehicle systems. This summary is based on those applications of which the author is most aware (including the results of a recent survey of future directions in control, dynamics and systems [Mur03]), as well as a survey of the literature (with emphasis on papers that are frequently referenced by others). Although not comprehensive, the applications cited here demonstrate some of the key features that must be addressed in solving cooperative control problems.

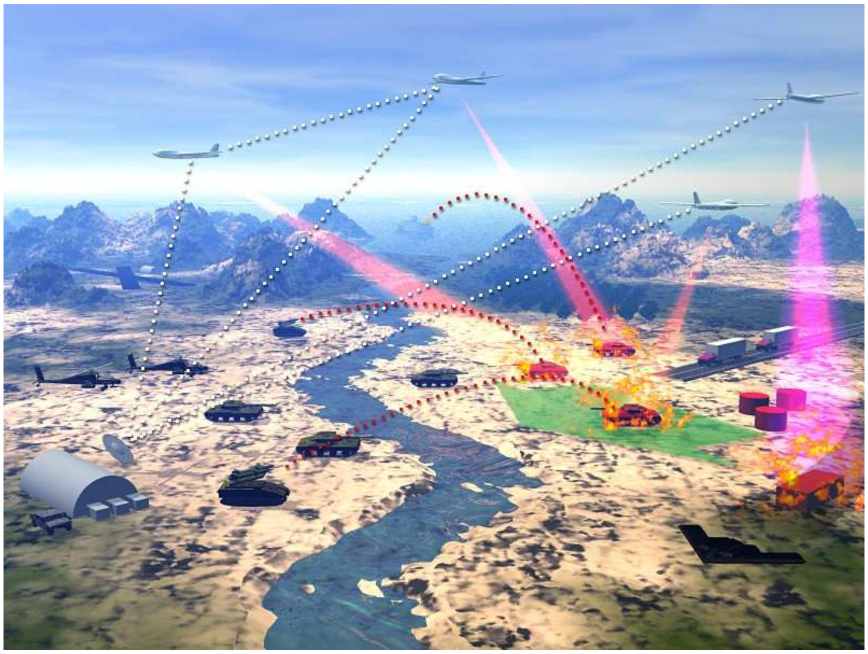


**Figure 10.1:** Battle space management scenario illustrating distributed command and control between heterogeneous air and ground assets. Figure courtesy of DARPA.

### Military Systems

Modern military systems are becoming increasingly sophisticated, with a mixture of manned and unmanned vehicles being used in complex battlefield environments, such as the one depicted in Figure 10.1. Traditional solutions involve a centralized resource allocation (assignment of planes to targets), followed by decentralized execution (each attack vehicle is responsible for a set of targets). More modern battlespace management systems are considering the use of cooperative operation of large collections of distributed vehicles, with location computation, global communication connections and decentralized control actions [Mur03, **?**].

*Formation flight.* One of the simplest cooperative control problems is that of formation flight: a set of aircraft fly in a formation, specified by the relative locations of nearby aircraft. This area has received considerable attention in the literature. Some of the earliest work in this area is that of Parker [**?**], who consider the design of control laws that use a combination of local and global knowledge to maintain a formation.

NASA has experimented with formation flight as a method for reducing drag on a collection of aircraft [Lav02]. The key idea is to locate the aircraft such that the tip vortices of one aircraft help reduce the induced drag of the tailing aircraft. This task requires precise alignment of an aircraft with the aircraft in front of it. To date, demonstrations of this concept in engineering systems have been restricted to small numbers of aircraft. Similar formations in nature can involve many more individuals [**?**].

*Cooperative classification and surveillance.* Chandler et al. [CPR01] define the cooperative classification problem as “the task of optimally and jointly using multiple vehicles’ sightings to maximize the probability of correct target classification”. More generally, we can define the cooperative surveillance problem as that of using a collection of vehicles to maintain a centralized or decentralized description of the state of a geographic area. This description might include the current state of features that are spatially fixed (such as the number of people in a given location) or information about entities that are moving in the region of interest (eg, locations of cars and planes in a given region).

The cooperative classification problem is one in which the performance function involves the collection of maximal amounts of relevant information. One typically assumes that the vehicles can communicate over some range (possibly limited by line of site, especially for ground-based vehicles) and information shared between the vehicles can be used by the vehicles in determining their motion.

*Cooperative attack and rendezvous.* The rendezvous problem involves bringing a collection of vehicles to a common location at a common time. Depending on the application, the rendezvous time may either be fixed ahead of time or determined dynamically, based on when all vehicles reach the same area. Military applications of rendezvous include minimizing exposure to radar by allowing aircraft to fly individual paths that are locally optimized [CPR01].

*Mixed initiative systems.* A variant of the cooperative control problem is the *mixed initiative* cooperative control problem, in which collections of autonomous vehicles and human operators (on the ground or in vehicles) must collectively perform a task or a mission. This class of problems adds the complexity of providing situational awareness to the operators and allow varying levels of control of the autonomous system.

### Mobile Sensor Networks

A second area of application in cooperative control is networks of sensors that can be positioned so as to maximize the amount of information they are able to gather. In this section we provide some examples of the types of cooperative control applications that are being pursued in this area.

*Environmental sampling.* The Autonomous Ocean Sampling Network (AOSN) [**?**], pictured in Figure 10.2 is an example of an environmental sampling network. † The network consists of a collection of robotic vehicles that are used for “adaptive sampling”, in which the motion of the vehicles is based on the

**RMM**: Get original pictures from Naomi [later]

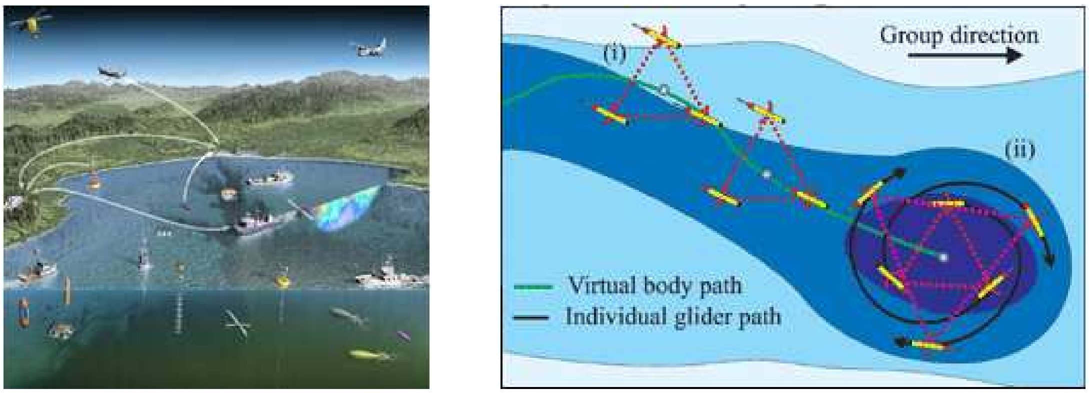


**Figure 10.2:** Autonomous ocean sampling network: (a) a depiction of the collection of vehicles that were part of the summer 2003 experiment; (b) an example of using a collection of gliders for sampling a region of interest.

observations taken by the vehicles. This approach allows the sensors to be positioned in the areas in which they can do the most good, as a function of the data already collected. Because of the distributed nature of the measurements being taken, a cooperative control strategy is used to control the motion of the vehicles. In tests done in the summer of 2003, over 20 vehicles were controlled over 4 weeks to collect data [**?**].

*Distributed aperture observing.* A related application for cooperative control of multi-vehicle systems is distributed aperture (or phased array) imaging, illustrated in Figure 10.3. The proposed TechSat 21 project was sponsored by the US Air Force Research Laboratory (AFRL) and was to have launched a collection of “microsatellites” that would be used to form a “virtual” satellite with a single, large aperture antenna (the project was canceled in 2003). Another example of a distributed aperture observing system is the terrestrial planet finder (TPF), being proposed by NASA. TPF uses optical interferometry to image distance stars and to detect slight shifts in the stars positions that indicated the presence of planets orbiting the stars [**?**].

### Transportation Systems

Finally, the use of cooperative control in transportation systems has received considerable attention over the last few decades.

*Intelligent highways.* Several groups around the world have begun to explore the use of distributed control for problems related to intelligent highway and transportation systems. These problems include increased interaction between individual vehicles to provide safer operations (e.g., collision warning and avoidance), as well as interaction between vehicles and the roadway infrastructure. These latter applications are particularly challenging since they begin to link heterogeneous vehicles through communications systems

**Figure 10.3:** Distributed aperture observing systems: (a) the proposed TechSat 21 concept would use a collection of microsatellites to form the equivalent of a larger aperture imaging system; (b) the terrestrial planet finder uses formation flying to enable optimal interferometry for detecting planets.

that will experience varying bandwidths and latency (time delays) depending on the local environment. Providing safe, reliable, and comfortable operation for such systems is a major challenge that will have application in a variety of consumer, industrial, and military products and systems.

A representative example of this class of applications is the California Partners for Advanced Transit and Highways (PATH) project [**?**]. In 1997 the PATH project developed and demonstrated a system for allowing cars to be driven automatically down a freeway at close spacing, as shown in Figure 10.4a. By decreasing the spacing of cars, the density of traffic on a highway can be increased without requiring additional lanes. Additional work within the PATH project has looked at a variety of other systems for better managing traffic flow [**?**].

*Air traffic control.* Air traffic control systems are another area where methods for cooperative control are being explored (see, e.g., [TPS98]). As the density of air traffic continues to increase, congestion at major airports and automated collision warning systems are becoming increasingly common. Figure 10.4b illustrates some of the complexity of the current air traffic control networks. Next generation air traffic control systems will likely move from a human-controlled, centralized structure within a given region to a more distributed system with “free flight” technologies allowing aircraft to travel in direct paths rather than staying in pre-defined air traffic control corridors. Efforts are now being made to improve the current system by developing cockpit “sensors” such as augmented GPS navigation systems and data links for aircraft to aircraft communication citeatc.

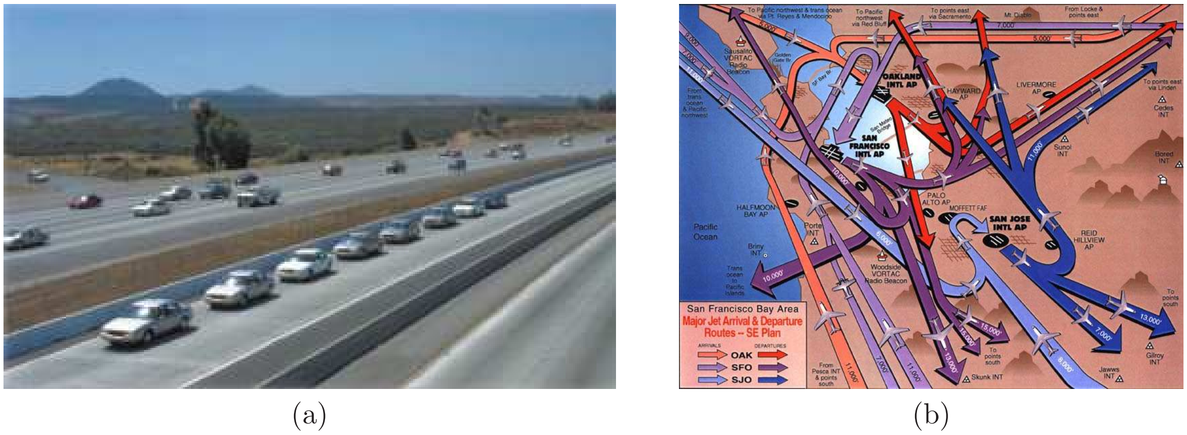


**Figure 10.4:** Transportation systems. (a) A platoon of cars driving down the San Diego freeway as part of the PATH project [**?**]. (b) The San Francisco Bay area aircraft arrival and departure routes (courtesy of Federal Aviation Authority).

## Testbeds

A variety of testbeds have been developed to explore cooperative control problems in laboratory settings. Perhaps the most well known is *RoboCup*, a multi-vehicle game of robot soccer. RoboCup was initially conceived as an attempt to foster research in artificial intelligence, specifically that of multiple vehicles in a highly dynamic environment [**?**]. The RoboCup competition is now held annually and has competitions involving a variety of different physical and simulation platforms. Most of the RoboCup competitions allow the use of centralized computation, although some teams have made use of decentralized strategies [**?**].

A related game, dubbed *RoboFlag* has been developed at Cornell [**?**] and is loosely based on “Capture the Flag” and “Paintball”. Two teams play the game, the red team and the blue team, as depicted in Figure 10.5. The red team’s objective is to infiltrate blue’s territory, grab the blue flag, and bring it back to the red home zone. At the same time, the blue team’s objective is to infiltrate red’s territory, grab the red flag, and bring it back to the blue home zone. The game is thus a mix of offense and defense: secure the opponent’s flag, while at the same time prevent the opponent from securing your flag. Sensing and communications are both limited to provide a more realistic distributed computing environment. The game is meant to provide an example of multi-vehicle, semi-autonomous systems operating in dynamic, uncertain, and adversarial environments. Human operators can also be present in the system and can be used either as high level controllers or as low level (remote) “pilots”. A centralized control unit may be used to coordinate the vehicles, but it must respect the communication constraints (bandwidth and latency) of the system.

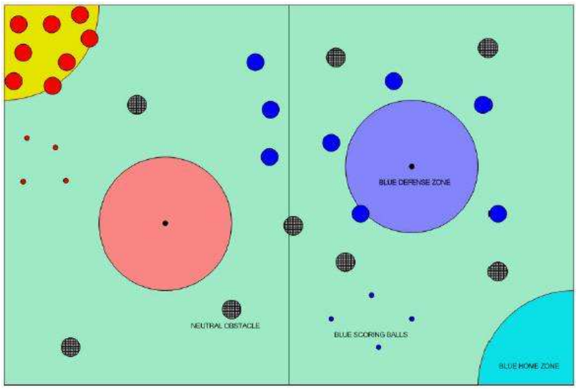


**Figure 10.5:** The RoboFlag playing field [**?**].

Several physical testbeds have also been developed, ranging from wheeled robots such as those used in RoboCup, to hovercraft that provide some of the dynamics more typical of aircraft [**?**, VSR$^+$04], to small-scale aircraft [HRW$^+$04, KKAH04] and helicopters [SKS03, SV99]. These citations are far from complete, but give an example of the range of physical testbeds that have been developed.

## 10.3 Technology Overview

In this section we provide a brief survey of some of the techniques that have been developed for designing strategies for cooperative control tasks. We make use of the mathematical notation defined in the introduction wherever possible. We focus primarily on the problem formulation and the approach used in its solution, leaving the details of the proofs of stability, convergence and optimality to the original papers.

### Formation Control

Many of the applications above have as part of their solution the ability to maintain the position of a set of vehicles relative to each other or relative to a reference. This problem is known as *formation control* and has received considerable attention, both as a centralized and as a decentralized problem.

*Optimization-based approaches.* One way to approach the formation control problem is to formulate it as an optimization problem. If we let $L^i(x^i, x^{-i})$ represent the individual formation error between the $i$th vehicle and its neighbors, then we can establish a cost function

$$L(x, \alpha, u) = \sum L^i(x^i, x^{-1}) + \|u^i\|^2_R,$$

where the summation over the individual formation errors gives the *cumula-*

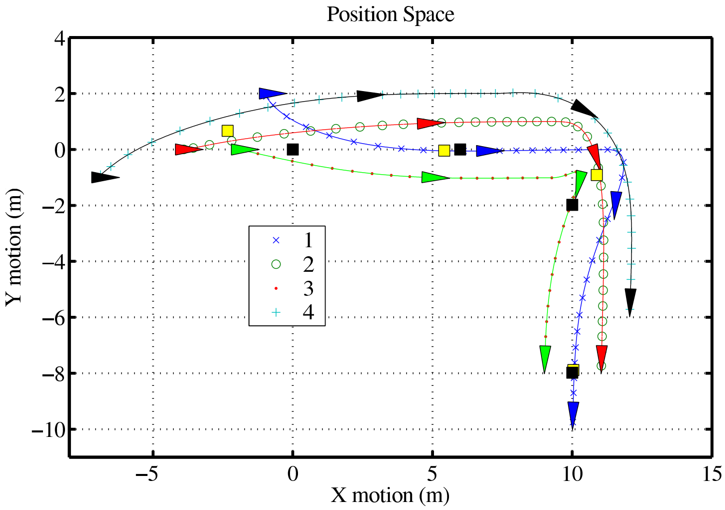


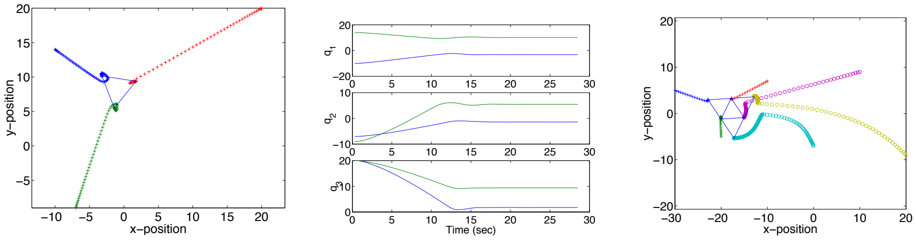
**Figure 10.6:** Four vehicle formation using distributed receding horizon control [DM06].

*tive formation error* [**?**] and the final term is a penalty on the inputs (other forms could be used).

This problem can be solved in either a centralized manner or a distributed manner. One distributed approach is the work of Dunbar *et al.* [DM06], who considers cooperative control problems using receding horizon optimal control. For a cost function whose coupling reflects the communication constraints of the vehicles, he generates distributed optimal control problems for each subsystem and establishes that the distributed receding horizon implementation is asymptotically stabilizing. The communication requirements between subsystems with coupling in the cost function are that each subsystem obtain the previous optimal control trajectory of those subsystems at each receding horizon update. The key requirements for stability are that each distributed optimal control not deviate too far from the previous optimal control, and that the receding horizon updates happen sufficiently fast.

Figure 10.6 shows a simulation of Dunbar's results. The vehicles are flying in "fingertip formation", with vehicles 2 and 3 maintaining position relative to vehicle 1 and vehicle 4 maintaining position relative to vehicle 2. The control goal is to maintain formation around the black square, which is flying along a trajectory that is not known to the individual aircraft. The localized optimization for each vehicle uses a previous optimal path for its neighbors while constraining its own path to stay near the previous path that it communicated to others.

*Potential field solutions.* Another approach to solving the formation control problem is to consider the mechanical nature of the systems and to shape



(a) (b) (c)

**Figure 10.7:** Formation stabilization using potential functions [OSM02]. (a) Stabilization of three vehicles in the plane. (b) Time traces for individual positions of the vehicles. (c) Stabilization of a six vehicle formation.

the dynamics of the formation using potential fields. In this case, the control law for the individual vehicles has the form

$$u^i = \nabla V(\alpha^i, x^i, x^{-1})$$

where $V$ is a potential function that depends on the mode of the vehicle, $\alpha^i$ (typically whether it is a leader or a follower).

A representative body of research in this area is the work of Fiorelli and Leonard, who use the concept of "virtual leaders" that guide the motion of the other vehicles [LF01, OFL04]. They consider two types of potential functions: an interaction function $V_I$ and a potential generated by "leaders", $V_h$. Each function generates a repulsive force if vehicles are very close to each other, an attractive force if the vehicles are within some interaction range but not too close or too far and no force for vehicles beyond a certain radius. Their resulting control law is of the form

$$u^i = -\sum_{j \neq i}^{N} \nabla V_I(\|y^i - y^j\|) - \sum_{k \in \mathcal{L}} \nabla V_h(\|y^i - y^k\|) + f_{v^i},$$

where $\mathcal{L}$ is the set of leaders, $f_{v^i}$ is a dissipative force based on the velocity of the $i$ vehicle, and local coordinates are used for $y^i \in SE(3)$. By appropriate choice of $f_{v^i}$ they are able to show asymptotic stability of various schooling and flocking behaviors.

Other work on the use of potential fields includes that of Olfati-Saber [OSM02], who uses potential functions obtained from structural constraints of a desired formation in a way that leads to a collision-free, distributed, and bounded state feedback law for each vehicle. Figure 10.7 demonstrates some of the results of his algorithm for formation control.

*String stability.* One issue that arises in formation control is that of "string stability," in which disturbances grow as they propagate through a system of vehicles [SH96]. One of the early sources of research on this problem was in the control of vehicle platoons, in which one wanted to ensure that small disturbances at the beginning of a chain of vehicles did not get amplified as one progressed down the chain.

For simplicity, we assume that the disturbances enter through the initial states of the vehicles. String stability is defined in terms of an infinite collection of vehicles and our goal is to find a control law for each of the vehicles so that given $\epsilon > 0$ there exists a $\delta > 0$ such that

$$\sup_i \|x^i(0)\| < \delta \qquad \Longleftrightarrow \qquad \sup_i \|x^i(\cdot)\|_\infty < \epsilon,$$

where the $\infty$ norm is taken with respect to time. In particular, this implies that the motion of each vehicle is bounded for all time. More general norms can also be used, as described in [SH96].

Using this definition, one can show that a system is string stable if the $H_\infty$ gain between any two neighbors is less than one. If this is the case, then disturbances are attenuated as they pass down the chain of vehicles. Conversely, if the dynamics and control laws for each vehicle are identical and if the gain of the transfer function is greater than 1 at some frequency, then disturbances at that frequency can be amplified as they propagate down the chain. These definitions can be generalized to different topologies in which the neighbor sets are more complicated than a single chain.

To help compensate for string instabilities, one can make use of globally transmitted information that allows the vehicles to pre-compensate for disturbances. In essence, one changes the topology of the information flow from one in which each vehicle only sees the vehicle in front of it, to one in which vehicles also have global information about the position of the lead vehicle. Figure 10.8 shows the responses of a set of vehicles with different topologies and different levels of global information. In this simulation, the lead vehicle responds to a step input at time $t = 15$. The variable $\alpha$ controls the amount of mixing between the purely local strategy ($\alpha = 0$) and a purely centralized strategy ($\alpha = 1$).

It is also possible to define the performance in ways that are more structured than string stability, for example asking whether the distances between specified sets of vehicles have certain levels of disturbance attenuation [JM03, Jin06].

*Swarms.* Finally, although not strictly a formation control problem, there has been a great deal of interest in so-called “swarms” of vehicles. Roughly speaking, a swarm is a large collection of vehicles that perform in a collective fashion, such as flying together in a given direction. One early work in swarm-like behavior was that of Reynolds, who developed a set of rules that he used to generate realistic motion of vehicles for animation purposes [Rey87].

An innovative approach to understanding swarm behavior was taken by Jadbabaie, Lin and Morse [JLM03], who described how to achieve coordination of groups of mobile autonomous agents using nearest neighbor rules.

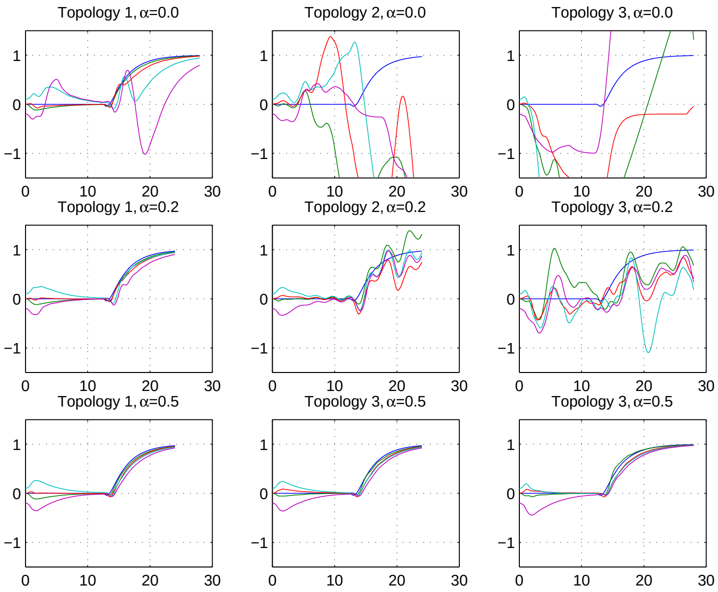


**Figure 10.8:** String stability results for a five vehicle formation [JM04]. Each column represents a different information topology, as shown in the diagram at the top of the column. The first row of plots corresponds to the use of purely local information, while the second two rows allow increasing amounts of global information.

The control law was quite simple, making use of a simple heading model in which each agent updated its heading according to the rule

$$u^i = \frac{1}{1 + |\mathcal{N}^i(t)|} \left( \theta^i(t) + \sum_{j \in \mathcal{N}^i(t)} \theta^j(t) - \theta^i(t) \right)$$

where $\mathcal{N}^i(t)$ is the set of vehicles that are within a radius $r$ of vehicle $i$ at time $t$. The first term is the average heading of the neighbors of vehicle $i$ and hence this control essentially tells each vehicle to steer in the same direction as its neighbors.

Jadbabaie et al. are able to demonstrate that with this control law, all vehicles will converge to a common heading. They make use of an "eventual connectivity" assumption in which the vehicles are connected together across intervals. In other words, while it may never be the case that at a given instant of time the graph describing the interconnectivity is complete, as long as over a suitable interval all vehicles are able to share information, the solution will converge to a common value.

Control laws for swarms often involve using attractive and repulsive functions between nearby vehicles. In addition to the work of Leonard et al. already described above, another representative work in this regard is that of

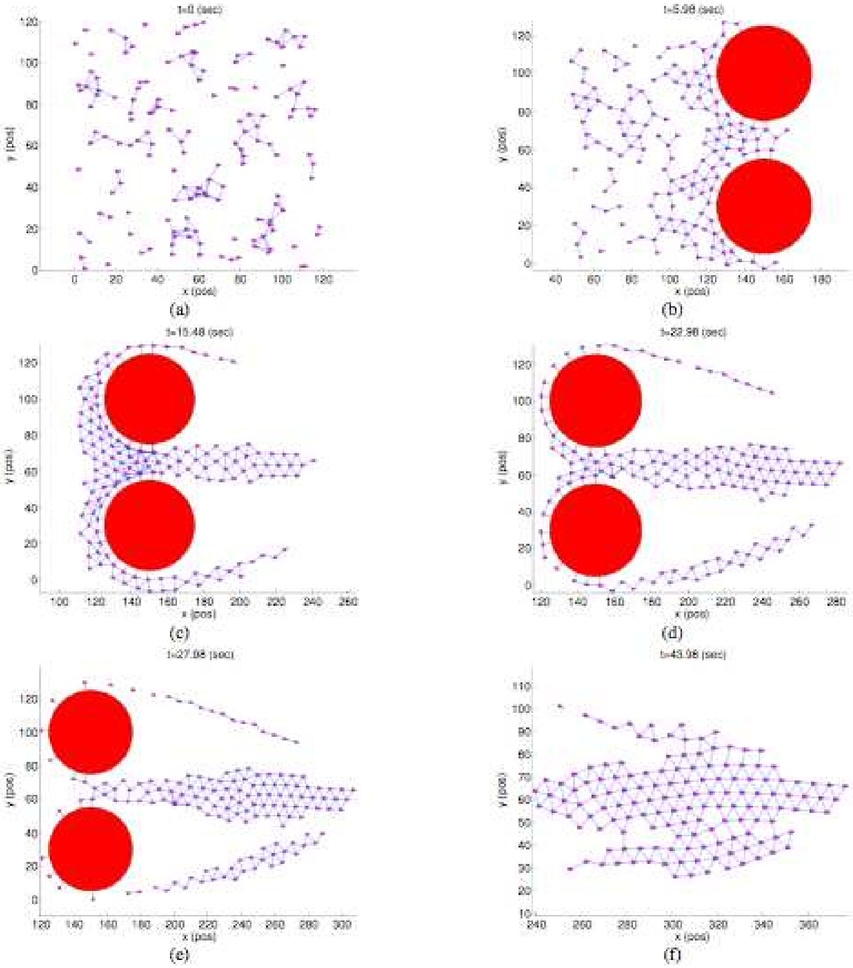


**Figure 10.9:** A squeezing maneuver using flocking algorithms of Olfati-Saber [OS06].

Olfati-Saber [OS06], who makes of a control input consisting of three terms

$$u^i = f_g^i + f_d^i + f_\gamma^i.$$

The first term $f_g^i = -\nabla V(y^i, y^{-1})$ is a gradient-based term where $V$ is a potential function. The second term $f_d^i$ is a damping term based on the relative velocities of neighboring vehicles and has the form $\alpha(q)(v^i - v^j)$. The final term $f_\gamma^i$ is a navigational feedback term that takes into account a group objective, such as moving to a given rendezvous point. Figure 10.9 shows a sample maneuver in which 150 agents squeeze through an opening without collision.

Substantial additional literature on stability analysis and motion control of swarms exists in the literature; see [OS06] for a recent survey.

### Cooperative Tasking

A major element of cooperative control is deciding on the tasks that different vehicles will perform to satisfy the team objective. This essentially amounts to choosing the role of the vehicles, $\alpha^i$.

*MILP formulations.* Several groups have formulated this problem as a mixed integer linear program (MILP) [ED05, RBTH02, SCPP03], in which the integer variables correspond to the role $\alpha^i$.

The work of Richards et al. [RBTH02] considers the problem of designing trajectories for a group of vehicles that collectively visit a set of waypoints within a given set of time constraints. They minimize a cost function of the form

$$J = \bar{t} + \rho_1 \sum_{p=1}^{N} \left( t^p + \rho_2 \sum_{t=0}^{T} \bigl( |u_1(t)| + |u_2(t)| \bigr) \right)$$

where $t^p$ is the time at which the $p$th vehicle completes its task and $\bar{t}$ is the time at which the last vehicle completes its task. This cost function thus trades off the input forces on the vehicles with the time that the overall task is completed as well as the tasks of the individual vehicles.

In the MILP formulation used by Richards et al. [RBTH02], the individual assignments of waypoints to vehicles is handled by using decision variables to constrain the problem such that each waypoint is visited exactly once by a vehicle. This constraint can be written in the form

$$\sum_{t=0}^{T} \sum_{p=1}^{N} K_{pi} b_{ipt} = 1 \qquad \text{for all waypoints } i$$

where $K_{pi}$ is the suitability of vehicle $p$ to visit waypoint $i$ and $b_{ipt}$ is 1 if vehicle $p$ visits waypoint $i$ and time $t$ and zero otherwise.

Figure 10.10 shows an example of the allocation problem applied to set set of 6 vehicles. The scenario includes 12 waypoints that must all be visited, along with a region of no fly zones (obstacles). An approximate method described in [RBTH02] is used to solve the problem in 27 seconds on a standard PC.

A similar approach has been developed independently by Earl and D'Andrea [ED05], in which the MILP formulation is used to solve a subproblem of the RoboFlag example in Section 10.2. Specifically, they solve the problem of guarding a defense zone from attackers that are trying to enter it. They formulate the problem in discrete time to be consistent with the MILP framework; for simplicity we will use a single time discretization here and re-use $t$ as the discrete time.

The objective function is given by

$$J = \sum_{t=0}^{T} \gamma(t) + \rho \sum_{t=0}^{T} |u(t)|$$

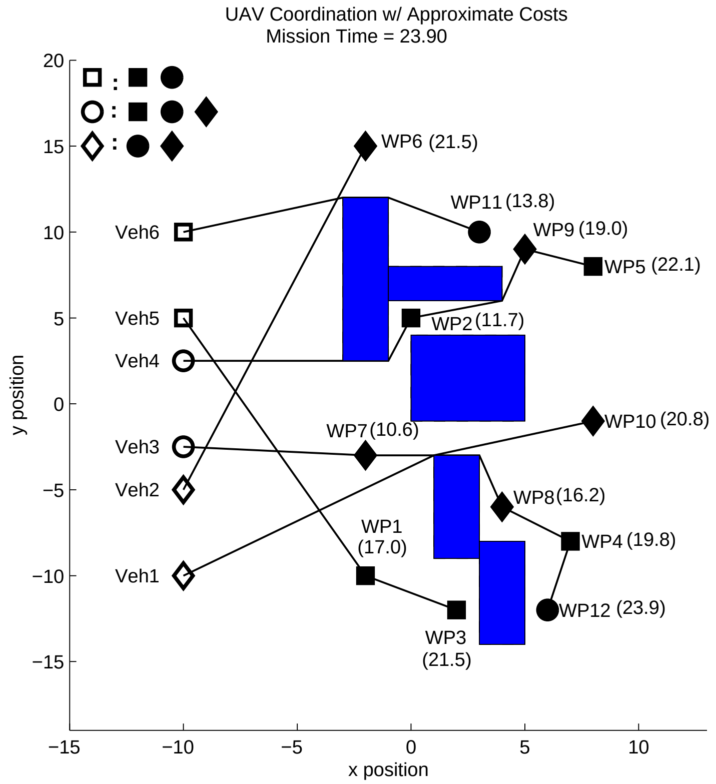


**Figure 10.10:** Resource allocation using mixed integer linear programming (MILP) [RBTH02].

where $\gamma(t)$ is a binary variable that takes on the value 1 if and only if one of the attackers is in the defense zone at time $t$. This function must be minimized while also constraining the position of the defending robots so that they avoid collisions with each other and stay outside of the defense zone.

In addition to the dynamics of the vehicles, a complete description of the problem also requires that we define the dynamics of the attacking robots. We do this using a discrete variable $\beta^i$ for each *attacker* that describes whether an attacker is active or inactive. An attacker is active initially and becomes inactive if it either enters the defense region or is "intercepted" by a defending robot (modeled by a defending robot getting within a certain distance of the attacking robot). We assume that if an attacking robot is active, it moves toward the defense zone in a straight line.

Note that in both of these formulations, the assignment is handled implicitly: the problem does not explicitly assign a given defender to specific attacker, but rather relies on the optimization to choose motions of the group of defenders such that no attackers enter the defense region.

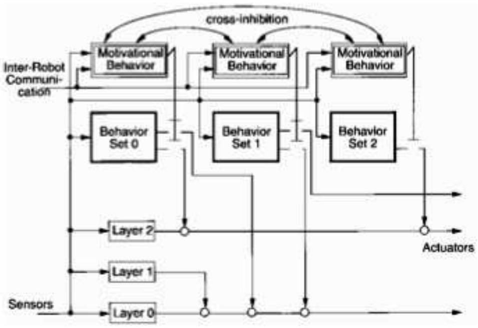


**Figure 10.11:** The ALLIANCE architecture [?].

*Assignment protocols.* Another approach to the cooperative tasking problem has been to develop protocols that are used to decide on who is assigned to what task. By "protocol" we mean a set of rules that are used to determine the individual roles (assignments) of each vehicle. One seeks to prove that this protocol results in all tasks being assigned to a vehicle, even in the presence of changing environmental conditions or failures.

One of the early approaches to distributed task allocation was the ALLIANCE software architecture developed by Parker [?]. Their approach made use of behavior sets that were activated under certain conditions. Each behavior could itself inhibit other behaviors, so that it was possible for a single behavior set to control the motion of the robot. Figure 10.11 illustrates this architecture.

The activation of a behavior set is controlled through "motivational behaviors". Each motivation behavior responds to some set of inputs, including external sensors, inter-robot communications, inhibitory feedback from other behaviors, and internal motivations. The two internal motivations, robot impatience and robot acquiescence, allow the robot to progress when other robots fail to complete a task or when the robot itself fails to accomplish a task. These motivational behaviors can be viewed in the context of the guarded command framework discussed in Section 10.1.

A related approach has been taken by Klavins [?], who constructed a language for describing and verifying protocols for cooperative control. The computation and control language (CCL) uses the guarded command formalism to specify sets of actions for a collection of robots. Figure 10.12 gives an example of how a distributed area denial task can be solved in CCL. In this example, drawn from the RoboFlag game, 6 defensive robots are trying to protect a defense zone for an incoming set of robots, which descend vertically at a fixed speed. The defending robots must move underneath the incoming robots, but are not allowed to run into each other. The defenders are randomly assigned incoming robots and are allowed to talk to their neighbors and switch assignments under a given protocol. A protocol

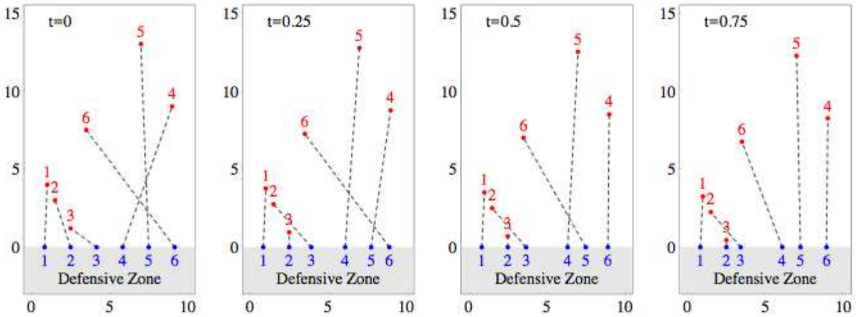


**Figure 10.12:** The RoboFlag Drill.

was developed in [?] that is able to provably solve this problem, including ensuring that no two robots collide and that all defensive robots eventually end up assigned to an incoming robot with no crossing of assignments. Extensions to this approach for observability and controllability have also been developed [DMK06, Del06].

*Other approaches.* Other approaches to the multi-vehicle task assignment problem include the use of genetic algorithms [SRSP06] and tree search [?].

## Spatio-Temporal Planning

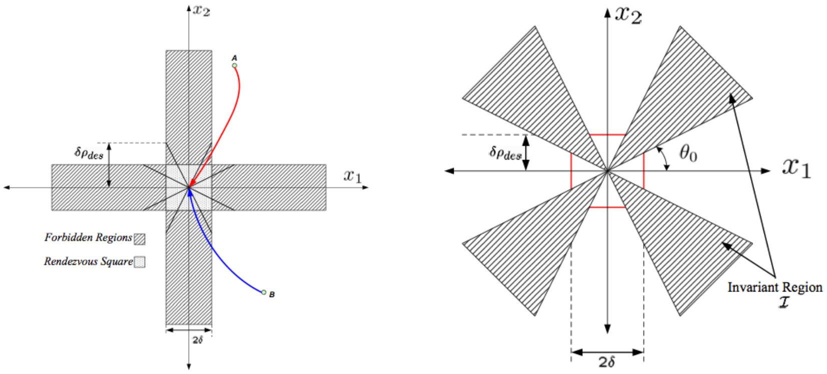
A broad collection of technological developments can be described under the heading of "spatio-temporal planning", in which the paths of the robots and their locations with respect to time are to be specified and controlled. In this section we consider two typical spatio-temporal planning problems: rendezvous and coverage.

*Rendezvous.* The rendezvous problem is a specific cooperative task in which one wants to have a number of individual vehicles meet at a common point at a common time. The key element in the rendezvous problem is that all agents should arrive at the *same* time, so that if one vehicle is delayed the other vehicles should adjust their trajectories so that they do not arrive early.

Bhattacharya et al. [?, TFI$^+$04] formulated the rendezvous problem by defining a rendezvous region $\mathcal{R}$ around the rendezvous point (taken as the origin) and letting $\rho$ be the ratio of the maximum and minimum distances of the vehicles at the time that one of them enters the rendezvous point. Letting $\delta$ be the radius of the rendezvous region and $t_a$ the time at which the first vehicle enters the region, they define $\rho$ as

$$\rho = \frac{\max(\|x_i(t_a)\|)}{\delta}.$$

The goal can then be defined as finding control laws such that from all initial



(a) (b)

**Figure 10.13:** (a) Definition of the rendezvous problem for two scalar agents. (b) Solution via construction of invariant cones.

conditions,

$$\rho \leq \rho_{\text{des}} \leq 1.$$

The case of "perfect" rendezvous corresponds to $\rho = 1$, in which case all vehicles must reach the rendezvous region at precisely the same time.

This problem can be solved using a Lyapunov-based approach that uses feedback to create an invariant cone in the phase space [TFI+04, **?**], as illustrated in Figure 10.13. The problem definition is shown in Figure 10.13a, which shows the phase space for two scalar vehicles. To achieve rendezvous, these vehicles must reach $x = 0$ at approximately the same time, without either of the individual vehicles coming near $x = 0$ before that time. This creates a set of forbidden regions in the phase space. By proper choice of control law, it is possible to render certain cones as invariant, as shown in Figure 10.13b. The resulting trajectories satisfy the rendezvous problem. The feedback in this case is centralized, requiring each vehicle to communicate its position to nearby vehicles.

*Coverage.* The coverage control problem refers to the use of a collection of vehicles to provide sensor coverage for a given geographic area. It is thus one approach to the cooperative surveillance problem. Given a set of $N$ vehicles, we wish to allocate each vehicle to a region in which it is responsible for providing sensor information. The centralized version of this problem is referred to as the locational optimization problem and there is a large literature describing different approaches (see [**?**] for a survey). We focus here on the decentralized solution proposed by Cortes et al. [**?**].

The approach taken by Cortes et al. is to partition a region $Q$ into a set of polytopes $\mathcal{W} = \{W^1, \dots, W^N\}$ that cover $Q$. Each polytope is assigned to a specific vehicle to each region and we let $f^i : \mathbb{R}_+ \to \mathbb{R}_+$ represent the sensing performance of a vehicle based on its distance from a given point, with $f$ small representing good performance. We then form the coverage control

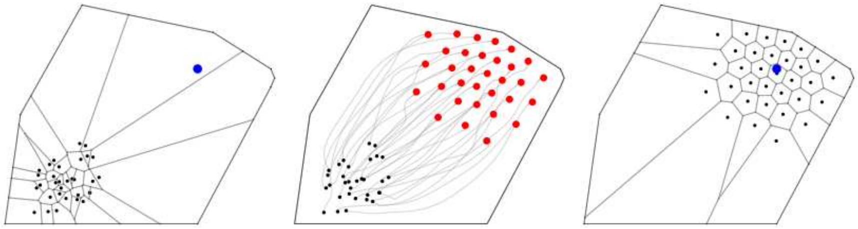


**Figure 10.14:** Coverage control applied to a polygonal region with Gaussian density function around the point in the upper right [**?**].

problem as choosing the locations of each vehicle such that we minimize

$$L = \sum_{i=1}^{n} \int_{W^i} f(\|q - y^i\|)\phi(q)dq, \tag{10.1}$$

where $\phi(q)$ is a distribution density function that represents the importance of a given area.

It can be shown that if the location of the vehicles are fixed, the optimal decomposition of the space $Q$ is a Voronoi decomposition where

$$W^i = \{q \in Q | \|q - y^i\| \le \|q - y^j\|, \forall j \neq i\}.$$

This decomposition corresponds to each vehicle being responsible for the points that are closest to it. This decomposition also introduces a natural graph of neighbors, with two vehicles being neighbors if their Voronoi partitions share an edge.

If we let $C_{V^i}$ represent the centroids of the Voronoi partition, then it turns out that the control law

$$u^i = -k(y^i - C_{V^i})$$

converges asymptotically to a set of critical points for the cost function, and hence provides (locally) optimal coverage. A key element of this approach is that the only communication required is with the nearest neighbors of the vehicle (since this is what is needed to determine the Voronoi decomposition). Figure 10.14 illustrates the coverage algorithm applied to a region with $\phi(q)$ being a Gaussian around the point in the upper right portion of the region.

The above formulation assumes that the collection of vehicles that is available is sufficient to cover the entire region of interest. A slightly different problem occurs when there is not enough sensor range to simultaneous view all portions of the environment that are of interest. In this case, one must selectively cover different regions of space and change those regions over time (so that no region goes unviewed forever). Several groups have considered this problem [**?**, **?**, TJJM05]

### Consensus algorithms

As a final technology in cooperative control, we briefly describe the problem of "consensus". The consensus problem is to have a group of vehicles (or more general agents) reach a common assessment or decision based on distributed information and a communications protocols. Many of the decentralized problems listed above, especially those involving assignment, can be thought of as special cases of consensus.

The consensus problem has been formulated as a coordinated control problem by Fax [FM04] and Olfati-Saber [OSM04]. A particularly simple solution to the consensus problem is to let the behavior of each agent be governed by the first order differential equation

$$\dot{x}^i = -\frac{1}{|\mathcal{N}^i|}\sum_{j=1}^{|\mathcal{N}^i|}(x^i - x^j),$$

where $x^i \in \mathbb{R}$ is the internal state of the agent. For this system, one can show that if the information flow is bidirectional (if agent $i$ is a neighbor of agent $j$, then $j$ is a neighbor of $i$), the states of the individual vehicles asymptotically converge to the average of the initial state values for any connected graph $\mathcal{G}$.

If $\mathcal{G}$ is not bidirectional (so that there are asymmetries in the information available to each agent), then the interaction above does not necessarily lead to average consensus. We define a graph to be *balanced* if the in-degree and out-degree of all nodes are equal. In the case of balanced graphs, one can once again show that any connected graph solves the average consensus problem using the interaction rules above [OSM04]. Furthermore, even if the connections are changing as a function of time, it can be shown that the average consensus is still reached.

When the behavior of the individual agents is more complicated, we can still pose the problem in a similar manner. Suppose that each agent's dynamics are governed by

$$\begin{aligned} \dot{x}^i &= Ax^i + Bu^i \\ y^i &= Cx^i \end{aligned} \tag{10.2}$$

Fax [FM04] considers a control law in which each system attempts to stabilize itself relative to its neighbors. This is accomplished by constructing an error for each system that is a weighted combination of the relative outputs of the neighbors:

$$e^i = \sum_{j\in\mathcal{N}^i} \alpha_{ij}(y^j - y^i) \tag{10.3}$$

where $\alpha_{ij}$ is the relative weight. For simplicity, we consider uniform weighting here, so that $\alpha_{ij} = 1/|\mathcal{N}^i|$ where $|\mathcal{N}^i|$ is the number of neighbors of node $i$. The results are easily extended to the more general case.

Given the error (10.3), we apply a compensator that attempts to stabilize the overall system. For simplicity, we assume here that the stabilizer is given by a constant gain

$$u^i = Ke^i, \tag{10.4}$$

with $K \in \mathbb{R}^{m\times m}$ representing the compensation (gain) matrix. In practice, one can use a dynamic compensator to improve performance, but for analysis purposes we can just assume these dynamics are included in the system dynamics (10.2).

The interconnectedness of the system, represented by the neighbor sets $\mathcal{N}_i$ can be studied using tools from graph theory. In particular, for the the case of uniform weighting of the errors, it turns out that the combined error vector $e \in \mathbb{R}^{N\cdot m}$ can be written as

$$e = (\bar{L} \otimes I)x \tag{10.5}$$

where $\otimes$ represents the Kronecker product and $\bar{L}$ is the *weighted Laplacian* associated with the (directed) graph that models the neighbors of each node. The weighted Laplacian is a standard object in graph theory and can be defined as

$$\bar{L} = D^{-1}(D - A)$$

where $D$ is a diagonal matrix whose entries are the out-degree of each node and $A$ is the adjacency matrix for the graph (see [FM04] for more detail). Using this framework, Fax showed the following:

**Theorem 10.1.** *A local controller $K$ stabilizes the formation dynamics in equation (10.2) with error (10.5) and gain $K$ if and only if it stabilizes the set of $N$ systems given by*

$$\begin{aligned} \dot{x} &= Ax + B \cdot \lambda_i \cdot (Ky) \\ y &= Cx \end{aligned} \tag{10.6}$$

*where $\{\lambda_i\}$ are the eigenvalues of the weighted graph Laplacian $\bar{L}$.*

This theorem has a very natural interpretation in terms of the Nyquist plot of dynamical system. In the standard Nyquist criterion, one checks for stability of a feedback system by plotting the open loop frequency response of the system in the complex plane and checking for net encirclements of the $-1$ point. The conditions in Theorem 1 correspond to replacing the $-1$ point with $-1/\lambda_i$ for each eigenvalue $\lambda_i$ of $\bar{L}$. This interpretation is illustrated in Figure 10.15. The results can easily be extended to consider weightings that are nonuniform.

Theorem 10.1 illustrates how the *dynamics* of the system, as represented by equation (10.2), interacts with the *information flow* of the system, as represented by the graph Laplacian. In particular, we see that it is the eigenvalues of the Laplacian that are critical for determining stability of the overall system. Additional results in this framework allow tuning of

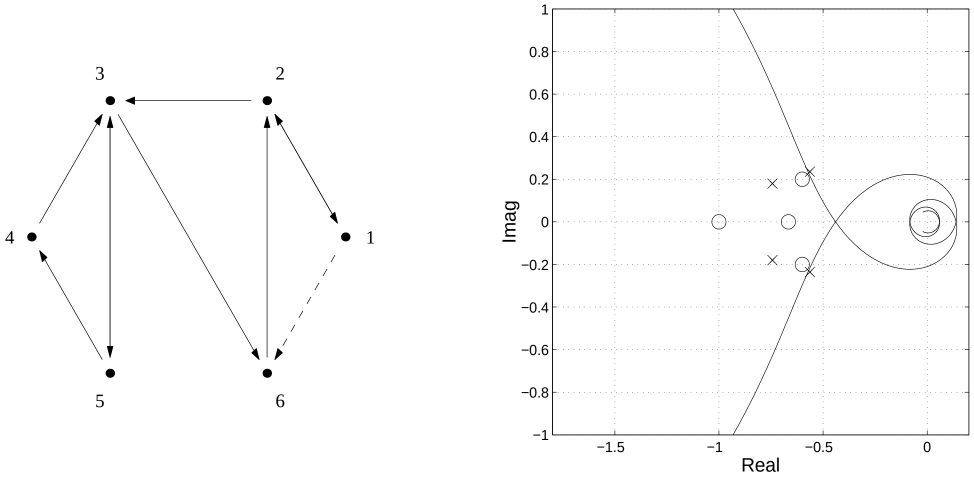


**Figure 10.15:** Interpretation of Theorem 1. The left figure shows the graph representation of the interconnected system and the right figure shows the corresponding Nyquist test. The addition of the dashed line to the graph moves the negative, inverse eigenvalues of $\bar{L}$ from the positions marked by circles to those marked by crosses.

the information flow (considered as both sensed and communicated signals) to improve the transient response of the system [FM04]. Extensions in a stochastic setting [**?**, OSM04] allow analysis of interconnected systems whose dynamics are not identical and where the graph topology changes over time.

## 10.4 Future Directions

While there has been substantial work in cooperative control over the past decade, there are still many open problems that remain to be solved. In this section we provide a brief review of some of the future opportunities in cooperative control. The topics listed here are not intended to be exhaustive, but rather to be indicative of the classes of problems which remain open. Many of these are drawn from the recent report on future directions in control, dynamics and systems [Mur03].

### Integrated control, communications and computer science

By its very nature, cooperative control involves the integration of communications and (distributed) computing systems with feedback control. In many applications, the traditional separation of computing, communications and control is no longer valid and new methods that integrate advances from the different disciplines are needed. Recent research in hybrid systems, in which continuous and logical domains are integrated, are a step in the right direction but these techniques often ignore issues associated with distributed computing and communication channels. Theories that define fundamental limits such as real-time computational complexity and performance limits of feedback systems with rate limited channels are needed.

### Verification and validation

Prescribed safety and reliability is a significant challenge for current mission-critical systems. Requirements, design, and test coverage and their quantification all significantly impact overall system quality, but software test coverage is especially significant to development costs. For certain current systems, verification and validation (V&V) can comprise over 50% of total development costs. This percentage will be even higher using current V&V strategies on emerging autonomous systems. Although traditional certification practices have historically produced sufficiently safe and reliable systems, they will not be cost effective for next-generation autonomous systems due to inherent size and complexity increases from added functionality.

New methods in high confidence software combined with advances in systems engineering and the use of feedback for active management of uncertainty provide new possibilities for fundamental research aimed at addressing these issues. These methods move beyond formal methods in computer science to incorporate dynamics and feedback as part of the system specification.

### Higher levels of decision making

The research surveyed in this paper has focused on cooperative control problems that can be formulated as optimization problems over some cost function. Many autonomous systems must make decisions for which an underlying set of continuous and discrete variables may not provide an appropriate level of abstraction for decision making. Cooperative systems that must reason about the complex interactions between the group's dynamics and the environment in which they operate may require different levels of representation of their task and their dynamics. Techniques from artificial intelligence that allow identification of strategies and tactics that can be coded as lower-level optimization-based problems are needed.

## 10.5 Conclusions

In this survey we have described some of the driving applications of cooperative control, surveyed some of the relevant technology that has been developed over the past decade and provided some possible directions for future study. Given the large and growing literature in this area, many interesting results have not been included in an attempt to capture some of the key areas of interest.

What is clear is that many of the basic problems of cooperative control have been explored and a wealth of results are available demonstrating the potential of such systems. To transition these research results to applications will require additional effort in the integration of control, communications and computer science; decision-making at higher levels of abstraction; verification and validation of distributed embedded systems; and an extensible architecture for networked control systems implementation.

The author would like to thank the US Air Force Office of Scientific Research for their past and continuing support of research in this area.

# Chapter 11
# Efficient Computation and Communications

## 11.1 Measurement Communication versus Estimate Communication

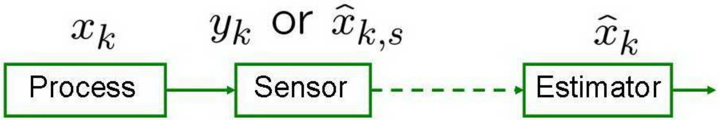


**Figure 11.1:** Estimation over a Network

Consider the following discrete-time process (Figure 11.1)

$$x_{k+1} = Ax_k + w_k. \tag{11.1}$$

A sensor measures $x_k$ and outputs

$$y_x = Cx_k + v_k \tag{11.2}$$

at each time $k$. In equation (11.1) and (11.2), $w_k$ and $v_k$ are uncorrelated zero-mean Gaussian random vectors with covariances $Q \geq 0$ and $R > 0$. The initial state $x_0$ is also assumed to be zero-mean and Gaussian with covariance $\Pi_0$.

The sensor communicates its data with a remote estimator across a network. Upon receiving the sensor data, the remote estimator computes the optimal linear estimate $\hat{x}_k$ of $x_k$.

We consider two scenarios in this section. In the first scenario, the sensor has limited computation and only sends $y_k$ to the remote estimator. We also call this *measurement communication* (or scheme one). In the second scenario, the sensor has sufficient computation and it runs a local Kalman filter to compute $\hat{x}_k^s$ and $P_k^s$, and sends $\hat{x}_k^s$ to the remote estimator. We also call this *estimate communication* (or scheme two).

Clearly $\hat{x}_k^s$ is given by $\hat{x}_k^s = \mathbb{E}[x_k | y_1, \ldots, y_k]$ and is computed from a Kalman filter.

### *Estimation over a Perfect Network*

If the communication network is perfect and does not introduce any data packet drops, it is easy to see that for scheme two, upon receiving $\hat{x}_{k,s}$, the state estimate $\hat{x}_k(2)$ is set to be equal to $\hat{x}_{k,s}$. Hence

$$\hat{x}_k(1) = \mathbb{E}[x_k | y_1, \dots, y_k] = \hat{x}_{k,s} = \hat{x}_k(2).$$

Therefore, we have the following result.

**Proposition 11.1.** *Assume the sensor has unlimited computation capability and the communication network does not introduce any packet drops. Then* $P_k(1) = P_k(2)\ \forall\ k > 0$, *and*

$$\lim_{k\to\infty} P_k(1) = \lim_{k\to\infty} P_k(2) = P_\infty.$$

*In other words, the two communication schemes produce the same estimation equality at the estimator.*

### *Estimation over a Packet-dropping Network*

Now consider the case when the communication network introduces data packet drops.

Let $\gamma_k(1)$ and $\gamma_k(2)$ be the indicator functions of whether the $y_k$ or $\hat{x}_{k,s}$ is successfully transmitted to the estimator or not, e.g., $\gamma_k(1) = 1$ means $y_k$ is received by the estimator and $\gamma_k(1) = 0$ means $y_k$ is dropped by the network and hence it will not be available at the estimator.

Consider the measurement communication scheme. When $\gamma_k(1) = 1$, it is clear that

$$\big(\hat{x}_k(1), P_k(1)\big) = \mathbf{KF}\big(\hat{x}_{k-1}(1), P_{k-1}(1), y_k\big).$$

When $\gamma_k(1) = 0$, we can write $\hat{x}_k$ and $P_k$ as

$$\begin{aligned} \hat{x}_k(1) &= A\hat{x}_{k-1}(1), \\ P_k(1) &= AP_{k-1}(1)A' + Q. \end{aligned}$$

In other words, when $\gamma_k = 0$, only the time update in the standard Kalman filter is performed. Define the function $h : \mathbb{S}^n_+ \to \mathbb{S}^n_+$ as

$$h(X) \triangleq AXA' + Q. \tag{11.3}$$

and the function $g : \mathbb{S}^n_+ \to \mathbb{S}^n_+$ as

$$g(X) \triangleq X - XC'[CXC' + R]^{-1}CX. \tag{11.4}$$

Then we can write $P_k(1)$ in a compact form as

$$P_k(1) = \begin{cases} h\big(P_{k-1}(1)\big) & \text{if } \gamma_k(1) = 0, \\ g \circ h\big(P_{k-1}(1)\big) & \text{if } \gamma_k(1) = 1. \end{cases}$$

Now consider the estimate communication scheme. When $\gamma_k(2) = 1$, it is easy to see that $\hat{x}_k(2) = \hat{x}_{k,s}$ and as a result, $P_k(2) = P_{k,s}$. When $\gamma_k(2) = 0$, it is also easy to see that $\hat{x}_k(2) = A\hat{x}_{k-1}$. Therefore

$$P_k(2) = AP_{k-1}(2)A' + Q.$$

We can also write $P_k(2)$ in a compact form as

$$P_k(2) = \begin{cases} h\big(P_{k-1}(2)\big) & \text{if } \gamma_k(2) = 0, \\ P_{k,s} & \text{if } \gamma_k(2) = 1, \end{cases}$$

where $P_{k,s} = g \circ h(P_{k-1,s})$.

Notice that, since $\gamma_k(1)$ and $\gamma_k(2)$ are random quantities, $P_k(1)$ and $P_k(2)$ are also random. Therefore instead of directly comparing $P_k(1)$ and $P_k(2)$, we compare their expected values. Before we state the main result of this section, we introduce a lemma.

**Lemma 11.1.** *For any $X \geq Y \geq 0$, the following hold:*

1. $h(X) \geq h(Y), \quad g(X) \geq g(Y)$.
2. $g(X) \leq X$.

*Proof.*

1. $h(X) \geq h(Y)$ is easy to show as $h(X)$ is quadratic in $X$. The proof for $g(X) \geq g(Y)$ can be found in Lemma 1-c at Appendix A in [**?**].
2. Since $XC'[CXC' + R]^{-1}CX \geq 0$, by definition,

$$g(X) = X - XC'[CXC' + R]^{-1}CX \leq X.$$

□

**Proposition 11.2.** *Assume the sensor has unlimited computation capability and the communication network may drop data packet. Further assume that $\gamma_k(1)$ and $\gamma_k(2)$ have the same distribution. Then*

1. $\mathbb{E}[P_k(1)] \geq \mathbb{E}[P_k(2)]$ $\forall k > 0$. *In other words, the average estimation equality at the estimator using scheme two is always better than using scheme one.*
2. $\lim_{k\to\infty} \mathbb{E}[P_k(1)] \leq M_1, \ \lim_{k\to\infty} \mathbb{E}[P_k(2)] = M_2$*, where* $M_1 \geq M_2 \geq 0$ *satisfy*

$$M_1 = \gamma g \circ h(M_1) + (1-\gamma)h(M_1), \tag{11.5}$$

$$M_2 = \gamma P_\infty + (1-\gamma)h(M_2). \tag{11.6}$$

*Proof.*

1. Since $\gamma_k(1)$ and $\gamma_k(2)$ have the same distribution, by definition of the expected value, it is sufficient to show

$$P_k(1) \geq P_k(2) \tag{11.7}$$

for any realization of the packet drop sequences $\gamma_k = \gamma_k(1) = \gamma_k(2)$. We use mathematical induction to prove Eqn (11.7).

(a) $P_0(1) = P_0(2) = P_0$.

(b) Assume $P_m(1) \geq P_m(2)$ for $0 \leq m < k$.

(c) At $m+1$:

i. If $\gamma_{m+1} = 1$, then

$$\begin{aligned} P_{m+1}(1) &= g \circ h\big(P_m(1)\big) = g \circ h \circ f_m \circ f_{m-1} \circ \cdots \circ f_1(P_0) \\ &\geq (g \circ h)^{m+1}(P_0) = P_{m+1,s} = P_{m+1}(2), \end{aligned}$$

where $f_i = g \circ h$ if $\gamma_i = 1$ and $f_i = 0$ if $\gamma_i = 0, i = 1, \ldots, m$. The inequality is from Lemma 11.1 as $h \geq g \circ h$.

ii. If $\gamma_{m+1} = 0$, then

$$P_{m+1}(1) = h\big(P_m(1)\big) \geq h\big(P_m(2)\big) = P_{m+1}(2).$$

The inequalities appeared in the induction steps are from Lemma 11.1. The three steps above complete the induction.

2. From [**?**], the operator $g_l$ is concave, hence by Jensen's Inequality, we get

$$\begin{aligned} \mathbb{E}[P_k(1)] &= \gamma \mathbb{E}\big[g \circ h\big(P_{k-1}(1)\big)\big] + (1-\gamma)\mathbb{E}\big[h\big(P_{k-1}(1)\big)\big] \\ &\leq \gamma g \circ h\big(\mathbb{E}[P_{k-1}(1)]\big) + (1-\gamma) h\big(\mathbb{E}[P_{k-1}(1)]\big). \end{aligned}$$

It is then easy to show by induction that Eqn (11.5) holds. Without loss of generality, assume $P_k(2) = P_{k,s} = P_\infty$. Then

$$\begin{aligned} \lim_{k\to\infty} \mathbb{E}[P_k(2)] &= \sum_{i=0}^{\infty} \gamma(1-\gamma)^i h^i(P_\infty) \\ &= \gamma P_\infty + (1-\gamma) h\Big(\sum_{i=0}^{\infty} \gamma(1-\gamma)^i h^i(P_\infty)\Big) \\ &= \gamma P_\infty + (1-\gamma) h\big(\lim_{k\to\infty} \mathbb{E}[P_k(2)]\big). \end{aligned}$$

Therefore

$$\lim_{k\to\infty} \mathbb{E}[P_k(2)] = M_2$$

and $M_2$ satisfies Eqn (11.6). Finally note that

$$M_2 = \lim_{k\to\infty} \mathbb{E}[P_k(2)] \leq \lim_{k\to\infty} \mathbb{E}[P_k(1)] \leq M_1.$$

□

From [**?**], there exists a critical rate $\gamma_c \in [0,1)$ such that if and only if $\gamma > \gamma_c$, Eqn (11.5) has a unique bounded solution and

$$\lim_{k\to\infty} \mathbb{E}[P_k(1)] < \infty.$$

A lower bound $\underline{\gamma}$ and an upper bound $\overline{\gamma}$ are also provided in [**?**] to bound the value of $\gamma_c$. The lower bound is given by

$$\underline{\gamma} = 1 - \frac{1}{\rho^2(A)}, \tag{11.8}$$

where $\rho(A)$ is the spectral radius of the matrix $A$. When $C^{-1}$ exists, it is proved that $\gamma_c = \underline{\gamma}$. It also turns out that as long as $\gamma > \underline{\gamma}$, Eqn (11.6) has a unique solution which can be verified by noticing

$$M_2 = \gamma P_\infty + (1-\gamma)h(M_2) = \tilde{A}M_2\tilde{A}' + (1-\gamma)Q + \gamma P_\infty,$$

where $\tilde{A} = \sqrt{1-\gamma}A$.

## 11.2 Trading Computation for Communication

(e.g., estimate error at the sensor; transmit new measurement if the error is large)

## 11.3 Local Temporary Autonomy and Shock Absorbers

The concept of *Local Temporary Autonomy* (LTA) was first proposed in the IT Convergence lab at the University of Illinois. In a networked control system, LTA can reduce an individual component's temporal dependency on other components and allow it to operate for some time even if other components fail to work.

The main idea of LTA is to introduce *Shock Absorbers* consisting of *State Estimators* placed at the controller and *Actuation Buffer* placed at the actuator.

As we see from previous chapters, networked control system offers many advantages than classic feedback control. However, many new issues arise such as random data packet delays, drops, etc., which may affect the system performance or even cause instability. The data packet delays and drops are frequently seen on a wireless communication network.

### *State Estimators*

Figure 11.2 shows one scenario where an estimator is placed at the controller side. Even when the sensor data arrive at random times due to the delay or drops, the estimator can always produce a regular state estimate. It is easy to note that this design can also tolerate temporary *sensor failure*. Hence if we need to shut down the sensor (e.g., change its battery) for a short period of time, the entire system can still work properly.

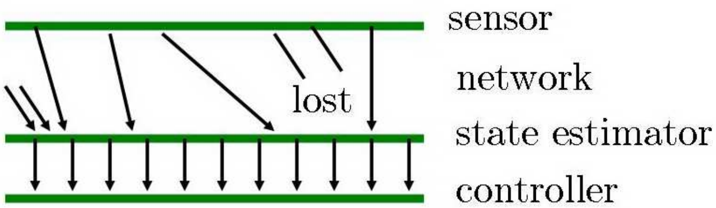


**Figure 11.2:** Use a state estimator to minimize the effect of sensor data delays or drops

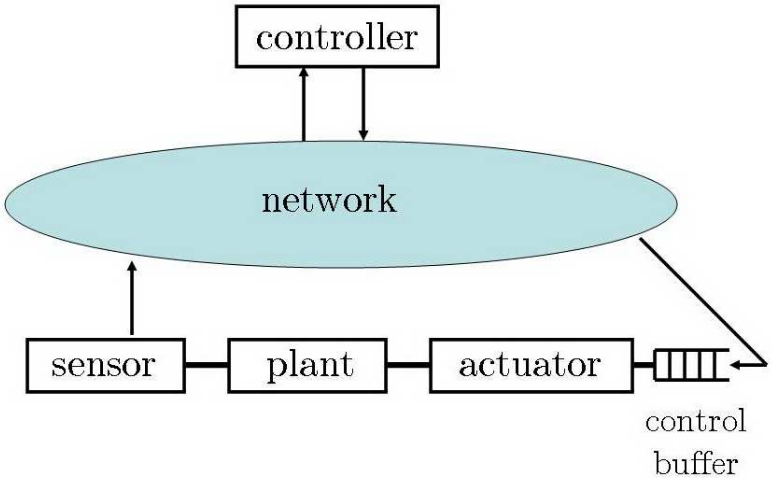


**Figure 11.3:** Use an actuation buffer to minimize the effect of control data delays or drops

### *Actuation Buffer*

While placing an estimator at the controller helps minimize the effect of sensor data delays or drops, placing an actuation buffer (or control buffer) at the actuator (Figure 11.3) helps minimize the effect of control data delays or drops. At each time, the controller computes not only the current control law to be applied, but also a sequence of future control laws using a receding horizon model predictive control approach, and sends these future control laws along with the current control law to the actuator. If the current control data is not received, the actuator executes the “current” control law which is from previous received data packet. If the current control data is received, then the entire data packet is put in the actuation buffer and the previous control data is discarded.

It was demonstrated in the IT Convergence lab that above approaches can effectively increase the system’s LTA.

## 11.4 Event-based Control: Transmit When Necessary

## 11.5 Further Reading

## 11.6 Exercise

# Chapter 12
# Sensor Networks

## 12.1 Introduction to Sensor Networks

*'Technological advances in semiconductors, storage, interfaces and networks enable a new computer class (platform) to form about every decade to serve a new need.'*

This empirical law, well backed up by historical data, was first formulated by Gordon Bell in 1972. As circuits continue to grow both smaller in size and greater in density, these new computers are also smaller, more powerful and cheaper to build, enabling their ubiquitous presence in our worlds.

Travelling far from the mainframe computers in the 1960s, we are now at the convergence of cellular phones and computers, having sailed through minicomputers, workstations, personal computers and personal digital assistants (PDA) during the journey. The latest class of computing devices, according to Bell, will provide access to streams of real time information to and from the physical world at an extremely fine level of granularity. For example, these computing devices will allow us to seek out and reserve a parking spot in a parking garage when we enter it, rather than driving endlessly from one level to the next in search of an empty space. They will give us the ability to adjust from afar the temperature in our home, office or car; help in preventing catastrophic effects of natural disasters, such as tsunamis or landslides; continuously monitor our body and transmit periodic updates to our physician, help scientist uncover the secrets of ecosystems as well as many many more. This is the goal behind the conception and development of Wireless Sensor Networks (WSNs). WSNs are ad hoc networks of devices that, in a single, small package, together have sensing, computing and communication capabilities. Sometimes called 'motes', these self-configuring devices can be deployed in any environment without a pre-designed topology, building a network that can route information via multi-hop wireless communication. Low power characteristics, combined with smart power management software and power scavenging modules, will enable long unattended deployments.

WSNs represent a paradigm shift from conventional networks. Internet technology is built around powerful devices that are pre-configured, have a fixed topology and operate in a static environment. Communication in wireless communication networks is usually one-to-one, such as cellular phone to base station, or one-to-many, such as all the broadcast media, with a point-to-point link between sender and receiver. Here communication is many-to-many, following a dynamically changing multi-hop path. Every node of the network is now a sender, a receiver and router, all at the same time. Moreover, connectivity is highly variable and, because of power savings requirements, radios need to be turned off for most the time.

## A Brief History of WSNs

'Smart Dust' is probably the most captivating definition of WSNs. The term refers to a research project sponsored by DARPA [**?**], and lead by professor Kristofer Pister, aimed at building a self-contained, millimeter-scale sensing and communication platform for massively distributed sensor networks. The target design goal was an inexpensive device of the size of a grain of sand, equipped with sensors, computational capability, bi-directional wireless communications, and a power supply, to be deployed by the hundreds. The science and engineering goal of the project was to build a complete, complex system in a tiny volume using state-of-the art technologies. The resulting prototype, shown in figure **??**, showed the feasibility of project and paved the way for the development of several other prototypes. In particular, the scientific community quickly acknowledged the bottleneck that the lack of adequate software infrastructure would create. Strict power requirements, computing and memory limitations impose a philosophical shift in the standard approach to software design. New software services have to be created to ensure reliable operation, remote management and constant health monitoring of the network, while simultaneously being constrained by the limited resources of computing power, memory and energy. On the hardware side a more user friendly device was needed to develop and test software systems based on WSNs. Also, a new, reprogrammable, reconfigurable device was needed to design and test several different classes of applications without the need to build custom hardware for each one. To address these issues a new version of wireless sensors devices was designed. It was 1998. The era of the 'mote' began.

The mote's hardware architecture resembles that of a very small computer with its microcontroller, RAM and storage, with the addition of a radio and a sensor board. On the hardware side the main differences with smart dust can be summarized as follows:

- **Input/output port**. An input/output port was included in the design to allow reprogrammability, swapping sensor boards, connection to a computer acting as a gateway between the sensor network and the outside world.
- **Storage unit.** A flash memory was installed to store data locally.
- **LEDs**. Light emitting diodes were installed for a quick feedback on the operation of the mote, for debugging purposes.

- **Radio communication.** As opposed to the smart dust node, which uses passive communication through a series of mirrors to reflect a received laser beam, the mote employs a radio frequency communication scheme.

Although the architecture has not changed much, several generations of motes have succeeded and now several companies, such as Crossbow, MoteIV, DUST Networks, Sensoria and Ember manufacture motes for commercial, military and research markets. Figure **??** shows the evolution of the Berkeley motes. State of the art encompasses extremely low power components, a digital radio, and USB connection.

### Software Architecture

#### *Operating System*

The most used operating system in the research community is the TinyOS [**?**, **?**]. The TinyOS is an open-source operating system designed for wireless embedded sensor networks. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as is required by the severe memory constraints inherent in sensor networks. TinyOS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools, all of which can be used as-is or be further refined for a custom application. TinyOS's event-driven execution model enables fine-grained power management and allows the scheduling flexibility made necessary by the unpredictable nature of the interaction of wireless communication systems with the physical world.

TinyOS has been ported to over a dozen platforms and numerous sensor boards. A wide community uses it in simulation to develop and test various algorithms and protocols. New releases have been downloaded over 10,000 times. Over 500 research groups and companies are using TinyOS on the Berkeley/Crossbow Motes. Numerous groups are actively contributing code to the sourceforge site. They working together to establish standard, interoperable network services based on direct experience and honed through competitive analysis in an open environment.

A three layer software architecture was conceived to achieve modularity, code reusability, separation of design objectives. At the bottom level is the software interacting with the platform, which comprises the core software services, such as a basic scheduler clock, radio stack, sensor drivers. The middle layer, also called middleware, includes all the software services supporting operations at the application level, such as time synchronization, localization, power management, routing services. At the highest level sits the application layer, where software components are designed to accomplish the desired functionality. The application layer needs to map specifications into constraints that are then pushed on the middleware services employed. A feedback loop is provided to the application layer to indicate if the constraints are satisfied. Chapter 2 will describe the relationship between different layers analyzing a Pursuit Evasion Game (PEG) application using Wireless Sensor Networks.

### Middleware Services

- **Security.**

  All data traffic is encrypted using hardware support provided on the node. Key management strategies for large-scale deployments can be evaluated. In addition to the key management and encryption, the system must be robust in the event of a series of attacks, and be able to operate during the attack while determining appropriate countermeasures to them. For example, if a portion of the nodes in the network is fully compromised, the attacker may use these nodes to alter, block or severely modify the running application. As a countermeasure, a trust map of the sensor network at the base station needs to be built and updated periodically. In addition, each node could build a neighbor table and rank its neighbors based on reputation and trust relation. The combination of the base station and node trust maps can help in isolating the compromised nodes, and providing reliable data aggregation.

- **Large-scale management.**

  A low-overhead, low-power, flexible network management facility will be demonstrated and evaluated. It must be easy to augment system and application code to enable management. The network operator must be able to easily express and obtain attributes and regions of interest. The protocols should scale to the level of management activity, dropping essentially to zero energy usage when management is inactive, operate even when other network layers or services are faulty, and have a very limited footprint.

- **Self-localization.**

  The many individual nodes should be able to determine their absolute or relative positions with little manual intervention, and in a robust and stealthy manner. Localization has proved to be a really challenging problem. Ranging, i.e. measuring distance between nodes, is made particularly difficult by fading, multi-path effects, as noted in the work of Whitehouse et al. [**?**]. Several algorithms are available, with different strengths and limitations [**?**].

- **Robust programming and Rapid Retasking.**

  It must be possible to reliably deliver, using the wireless channel, complete binary images of system or application code to large or focused subsets of nodes and to maintain consistency across the set even when nodes are added, die, or are intermittently connected. The Deluge [**?**] family of robust dissemination algorithms and extensions is designed to ensure integrity. It should also be possible to rapidly re-task the network among a family of possible behaviors.

- **Network communication.**

  The system capabilities above, as well as the application capabilities below, fundamentally rest upon four basic network communication capabilities: robust dissemination of information to a large collection of nodes, reliable collection of information, efficient exchange of information among physically localized groups of nodes, and routing of information from any (potentially mobile) point to any (potentially mobile) point.

- **Sensing and Identification.**

  Individual nodes have the capability to perform local sensing and signal processing. Passive vigilance is needed to ensure that the energy expended in sensing is proportional to detections, rather than time spent observing. The key concept is the sensor cascade, in which low-power, low-fidelity sensors with hardware wake-up capabilities can invoke selectively higher level, higher capacity assets. In addition, collections of nodes share processed information to refine the detection and classification.

- **Visualization.**

  It must be possible for human operators to observe the features the network has detected, as well as the health and status of the network itself. In both cases, the operator needs to be apprised of the certainty or uncertainty of the findings in order to plan responses. This information needs to be provided not only to fixed assets monitoring the network, but to handheld mobile assets moving through the network itself.

- **Tracking.**

  Identified objects need to be tracked and their estimated trajectories reported to a variety of receivers. Tracking multiple objects simultaneously requires the sensor network to correctly associate sensor readings to the movement of its respective object, group such readings and process them to produce the tracking trajectories, either in-network or remotely.

- **Asset coordination.**

  In response to a detection, a remote controlled aerial vehicle with camera mount will be directed to point of activity to gain further fidelity or autonomous unmanned ground vehicles will pursue the detected object. Such asset coordination introduces a closed loop feedback that will be utilized in several other aspects of maintaining the system, including improving localization accuracy by actively moving to and obtaining the position of nodes with high uncertainty, as well as repairing various network faults.

### Applications

The availability of WSNs technology has enabled the development of a handful of new applications, ranging from monitoring and surveillance to asset and people tracking. While most of these applications still live in the research and development, there exist products and services based on WSNs. This paragraph will illustrate a few of these applications.

- **Firebug: Design and Construction of a Wildfire Instrumentation System Using Networked Sensors [?].**

  Collecting real time data from wildfires is important for safety considerations, and allows predictive analysis of evolving fire behavior. One way to collect such data is to deploy sensors in the wildfire environment. FireBugs are small, wireless sensors (motes) based on TinyOS. The FireBug network self-organizes into edge-hub configurations for collecting real time data in wild fire environments. Hub motes act as base stations, by receiving sample data from any mote and sending commands to any mote. The FireBug system combines state-of-the-art sensor hardware running TinyOS with standard, off-the-shelf World Wide Web and database technology for allowing users to rapidly deploy FireBugs and monitor network behavior. The FireBug system is composed of a network of GPS-enabled, wireless thermal sensors, a control layer for processing sensor data, and a command center for interactively communicating with the sensor network. Each of these layers is independent of the others, communicating through well-defined interfaces.

- **Habitat Monitoring on Great Duck Island [?].**

  In the spring of 2002, the Intel Research Laboratory at Berkeley initiated a collaboration with the College of the Atlantic in Bar Harbor and the University of California at Berkeley to deploy wireless sensor networks on Great Duck Island, Maine. These networks monitor the microclimates in and around nesting burrows used by the Leach's Storm Petrel. The goal was to develop a habitat monitoring kit to enable researchers worldwide to engage in the non-intrusive and non-disruptive monitoring of sensitive wildlife and habitats.

  At the end of the field season in November 2002, well over 1 million readings had been logged from 32 motes deployed on the island. Each mote has a microcontroller, a low-power radio, memory and batteries. For habitat monitoring, sensors for temperature, humidity, barometric pressure, and mid-range infrared were added through a sensor board. Motes periodically sample and relay their sensor readings to computer base stations on the island. These in turn feed into a satellite link that allows researchers to access real-time environmental data over the Internet.

In June 2003, a second generation network with 56 nodes was deployed. The network was augmented in July 2003 with 49 additional nodes and again in August 2003 with over 60 more burrow nodes and 25 new weather station nodes. These nodes form a multi-hop network transferring their data back ”bucket brigade” style through dense forest. Some nodes are more than 1000 feet deep in the forest providing data through a low power wireless transceiver.

- **Structural Monitoring.**

  WSNs are a natural tool for distributed sensing. Being wireless, small and power efficient, they can be deployed easily without the need for wiring, they can be mounted anywhere and operate for a long time. Civil engineering applications such structural monitoring of constructions under solicitations have largely benefited from this technology. Sample applications include measuring the response of building under severe conditions, such as earthquakes, or structural health of the critical structures like the Golden Gate bridge in San Francisco, situated in a seismically unsafe area and subject to high seasonal winds. In both applications high resolution accelerometers are used together with nodes manufactured by Crossbow. Data collection is carried out wirelessly. Motes route data to a base station via multi-hop communication using neighboring nodes. In the first project a physical model of a building was shaken by forces comparable to the Canoga Park earthquake that occurred in the Los Angeles metropolitan area in 1994. Picture **??** shows the experimental setup and the shear stress on the building resulting from the simulated quake. The second project aims at monitoring vibration of the bridge, and detecting unusual behavior by wind earthquake or local damage. From a technological standpoint, these projects demonstrate the feasibility of WSNs for applications requiring high accuracy, high data rate collection and communication.

- **Traffic Monitoring [?].**

  Wireless magnetic sensor networks offer a very attractive, low-cost alternative to current technologies such as inductive loops, video cameras and radar for traffic measurement in freeways, urban street intersections and presence detection in parking lots. The actual network comprises 5 diameter sensor nodes glued on the pavement where vehicles are to be detected. The sensor nodes send their data via radio to the access point on the side of the road. The access point forwards sensor data to the Traffic Management Center via GPRS or to the roadside controller. The sensor node has a built-in magneto-resistive sensor that measures changes in the Earths magnetic field caused by the presence or passage of a vehicle in the proximity of the node. A low-power radio relays the detection data to the access point at user-selectable periodic reporting intervals or on an event driven basis. By placing two nodes a few feet apart in the direction of traffic, accurate individual vehicle speeds can be measured and reported.

- **Building Monitoring and Control.**

  Buildings, both residential and commercial, can greatly benefit from the use of sensor networks, by decreasing construction and operating costs, while improving comfort and safety. Furthermore, more than half of the cost of an HVAC system in a building is represented by installation and most of it is wiring. Wireless communications could sensibly lower that cost. Combining wireless technology with MEMS technology could reduce the cost further, allowing sensors to be embedded in products such as ceiling tiles and furniture, and enable improved control of the indoor environment. On the operating cost, WSNs could dramatically improve energy efficiency. With oil prices rising and not likely to decrease anytime soon, policy makers and researchers are working together to find ways to decrease consumption by avoiding useless waste. The United States is the bigger consumer of energy with 8.5 quadrillion British Thermal Units (BTU). Commercial and residential sectors account for about 40% of total consumption, according to a study conducted by the Energy Information Administration in March 2004). Employment of WSNs technology could potentially lead to sizeable energy savings. On the comfort and safety side, WSNs enable functions that traditionally are localized in a single point to be distributed over a wider space, with an opportunity to build more efficient systems, with more localized, precise climate control. While academic institutions are envisioning new applications, several companies, such as Carrier, Honeywell, Bosch just to name a few, are looking into integrating WSN technology into their existing core business.

- **Personal Health Monitoring.**

  The gathering of vital information from a person nowadays follows a pull process, via scheduled visits and tests. The health care cost of the American economy is huge, accounting to about 15% of GDP. The aging population will only increase this cost. Several factors contribute to the inefficiencies of this system. First of all, most of the clinical data is still on paper. Moving this information around is still a manual process. Check-ups are still sporadic, therefore rarely able to predict a problem before it occurs. It is estimated that Cardiac arrest kills 350,000 Americans per year, and only 6% of those not already in a hospital at the time they have a heart attack will survive the ordeal. Using WSN technology, each person could potentially be under continuous monitoring, and thus increasing the chance of detecting a problem at its early stage of development. What's more, patients' healthcare data could be saved, organized and retrieved automatically, which would improve the detection, prevention and care of medical problems while decreasing the overall administrative costs of the nation's health care systems.

### Closing the Loop around Sensor Networks

The applications described above can be organized in three categories depending on the use of the collected data:

- **Offline data analysis.**

  WSNs are used in this context to understand phenomena that have time and spatial components. Biologists use environmental data to understand ecosystems; civil engineers measure the stress that structures undergo in the occurrence of an earthquake, strong winds and other natural phenomena; traffic engineers collect data to better understand traffic patterns.

  These types of applications are the least challenging. Since the data is going to be analyzed offline, there is virtually no time constraint on the network for the delivery of the messages. If packets are lost, information can be re-sent. Also delay is not a main issue. The main challenges the designer has to face are data reliability, accurate design of a buffer to store the data waiting to be sent, time synchronization and sensor location information.

- **Online event detection.**

  In these types of applications, data is analyzed online to detect particular threshold phenomena. In this case, sensor data is used to make a discrete decision. In built environments, typical examples are represented by turning off unnecessary lighting, regulating A/C equipment and thermostats, and triggering an alarm if a potential threat is detected. Security systems for intrusion, anomaly and fault detection all belong to this category of applications. Remote health care systems can use WSNs to detect whether an elder person has fallen or, in the case of personal health monitoring systems, reveal anomalies in a person's physical health.

  The applications in this class are substantially more complex to design and implement. They add a real time and decision making component to the first class. The main issues here are detection accuracy and timely response. Security systems are overloaded due to high false positive detection percentage. In the decision making part, detection needs to be fast and promptly transmitted. In the sensing part, the WSN needs to be accurate by combining possibly different sensor data to provide accurate detection. At the network level, certain constraints, mostly loose, such as maximum tolerable delay, maximum packet loss, bandwidth limitation need to be satisfied. At the application level, these parameters have an important effect on application performance metrics such as the percentage of false detections, false negatives and delays in the detection and response.

- **Online estimation and control**

  This class of applications is probably the most challenging to design and implement. Here the goal is observe and control a certain dynamical phenomenon. In the most general case, measurements need to be collected and sent to one or more controllers, which will then estimate the state of the system and compute inputs to the actuators. Both measurements and inputs have very stringent time constraints, depending on the system dynamics, that the network needs to be able to satisfy. Examples of applications are Pursuit Evasion Games (PEGs), control of power grids, Scada networks, telesurgery, robocup, industrial control, manufacturing, environmental control.

The third set of applications encompasses all the previous difficulties and introduces new ones. In these kinds of applications, sensor data is collected by the sensors and sent to a controller via wireless communication. The controller, in turn, computes state estimate from sensory information, calculates the input to be given to the actuators and sends this information using the WSN multi-hop communication infrastructure. Placing a communication network in the control loop raises many issues. One of the key parameters in digital control systems design is the selection of a fixed sampling period. This is mainly a function of the system dynamics, and this places a hard constraint on the time necessary to receive observations, estimate the state, compute an input and transmit it to the actuators. All this needs to happen within one sampling interval. Computing power of modern machines, combined with usually wired, dedicated interconnection between different parts of the system guarantees that such constraints are met. When closing the loop around WSNs, the assumption of data availability does not hold anymore, as packets are randomly dropped and delayed. While system and control theory provides a wealth of analytical results, the assumptions that the theory is traditionally based upon do not hold true in this setting, and neglecting these phenomena may yield to catastrophic overall system performance. A notion of time, either global or local, is needed to order and combine possibly different sensor data for state estimation. The estimator needs to know what to do when observations are not arriving, and the controller needs to design an input using uncertain state estimates, not knowing whether its previous input has been successfully received by the actuators.

More generally, the use of networks in control systems imposes a paradigm shift in the designer's mentality. Deterministic methods need to be replaced by stochastic ones, as such is the nature of network phenomena. This argument is particularly true in wireless networks, where the use of a shared channel with random disturbances and noise cannot be modelled deterministically. Resources in the network (e.g., bandwidth, energy, power, etc.) need to be appropriately allocated in order to optimize system performance.

Put an overview of the chapter here. **LS**

## 12.2 Sensor Scheduling

Literature review here. **LS**

### Performance Optimization Subject to Resource Constraint

Consider the following system with one process whose state is to be estimated by multiple sensors

$$x_{k+1} = Ax_k + w_k, \tag{12.1}$$
$$y_x^i = H_i x_k + v_k^i,\ \ i = 1, \dots, N, \tag{12.2}$$

where $w_k$ and $v_k^i$ are uncorrelated zero-mean Gaussian random vectors with covariances $Q \geq 0$ and $R_i > 0$ for all $k$ and $i$. The initial state $x_0$ is also assumed to be zero-mean and Gaussian with covariance $\Pi_0$. Assume at each time $k$, only one of the $N$ sensors can send its measurement data to a remote estimator which computes $\hat{x}_k$, the estimate of $x_k$, based on previously received data. Denote $P_k$ as the error covariance of $\hat{x}_k$.

Let $\theta$ be a scheduling scheme that determines at each $k$, which sensor is selected to send its measurement. Clearly $\hat{x}_k$ and $P_k$ are functions of $\theta$ and they are computed as

$$\begin{aligned}\hat{x}_k(\theta) &= \mathbb{E}[x_k | \text{all data packets received up to } k],\\ P_k(\theta) &= \mathbb{E}[(x_k - \hat{x}_k)(x_k - \hat{x}_k)' | \text{all data packets received up to } k].\end{aligned}$$

From Chapter 2, when sensor $i$ is selected at time $k$, the *a priori* error covariance matrix $P_{k|k-1}$ evolves as

$$\begin{aligned}P_{k+1|k} = {} & AP_{k|k-1}A' + Q\\ & - AP_{k|k-1}H_i'[H_iP_{k|k-1}H_i' + R_i]^{-1}H_iP_{k|k-1}A'\end{aligned} \tag{12.3}$$

where the recursion starts from $P_{0|-1} = \Pi_0$. We shall simply write $P_{k|k-1}$ as $P_k$ for notational convenience. Apparently, $P_k$ depends on the schedule of the sensors. We wish to find a schedule such that $P_k$ is minimum in the steady state. Consider the following simple algorithm [**?**] that chooses a sensor $i$ at time $k$ stochastically. First we have the following result on the upper bound of the expected estimation error.

**Theorem 12.1.** *Assume the $i$-th sensor is chosen at time $k$ with probability $\pi_i$ independently at each time. Then $\mathbb{E}[P_k]$, the expected error covariance of the estimate, is upper bounded by $\Delta_k$ which is given by the following recursion*

$$\Delta_{k+1} = A\Delta_k A' + Q - \sum_{i=1}^{N} \pi_i A\Delta_k H_i'[H_i\Delta_k H_i' + R_i]^{-1} H_i \Delta_k A', \quad (12.4)$$

*where the initial condition is $\Delta_0 = \Pi_0$.*

*Proof.* Define

$$f_{H_i}(P) = APA' + Q - APH_i'[H_iPH_i' + R_i]^{-1}H_iPA'$$

and

$$f_{H_i}^k(P) = \underbrace{f_{H_i}(f_{H_i}(\cdots(f_{H_i}(P)\cdots))}_{f_{H_i}\text{ applied } k \text{ times}}.$$

From [**?**], $f_{H_i}$ is concave and increasing in $P$ when $P \geq 0$. We can also rewrite $P_k$ in equation (12.3) as

$$P_{k+1} = f_{H_i}(P_k) \text{ when sensor } i \text{ is selected.}$$

Therefore

$$\mathbb{E}[P_{k+1}] = \sum_{i=1}^{N} \pi_i \mathbb{E}[f_{H_i}(P_k)].$$

Since $f_{H_i}(P)$ is concave in $P$, using Jensen's Inequality, we immediately obtain

$$\mathbb{E}[P_{k+1}] = \sum_{i=1}^{N} \pi_i \mathbb{E}[f_{H_i}(P_k)] \leq \sum_{i=1}^{N} \pi_i f_{H_i}\big(\mathbb{E}[P_k]\big).$$

As $f_{H_i}$ is an increasing operator, we obtain the upper bound $\Delta_k$. □

The convergence of the upper bound $\Delta_k$ implies boundedness of the recursion in equation (12.3). As an example, if $A$ is stable, the recursion in equation (12.4) converges. The case when $A$ is stable (and thus the process to be estimated does not grow unbounded) is very important in a large number of practical applications of estimation. The algorithm thus consists of choosing $\pi_i$'s so as to optimize the upper bound as a means of optimizing the expected steady-state value of $P_k$ itself. The problem is solved under the constraint of probabilities being non-negative and summing up to 1. The optimization problem can be solved by a gradient search algorithm or even by brute force search for a reasonable value of $N$. After determining the probability values, the sensors are turned on and off with their corresponding probabilities. Note that the implementation does assume some shared randomness and synchronization among the sensors so that two sensors are not turned on at the same time. This can readily be achieved, e.g., through a common seed for a pseudo-random number generator available to all the sensors. Alternatively a token-based mechanism for the scheme can be implemented. Also note that the algorithm is run off-line and it has to be reapplied every time the number of sensors changes. However, if a sensor is stochastically failing with a known probability, we can model that in the algorithm.

### Resource Optimization with Guaranteed Performance

Consider a linear system

$$\begin{aligned} x_{k+1} &= Ax_k + w_k, \\ y_k &= Cx_k + v_k. \end{aligned} \tag{12.5}$$

where $w_k, v_k, x_0$ are independent Gaussian random variables, and $x_0 \sim \mathcal{N}(0,\ \Sigma)$, $w_k \sim \mathcal{N}(0,\ Q)$ and $v_k \sim \mathcal{N}(0,\ R)$. We assume that $x_k \in \mathbb{R}^n$ and $y_k = [y_{k,1}, y_{k,2}, \ldots, y_{k,m}]^T \in \mathbb{R}^m$ is the vector of the measurements from the sensors such that the element $y_{k,i}$ represents the measure of the sensor $i$ at time $k$.

Assume that the sensor nodes are battery powered. Let $E_{k,i}$ denote the remaining energy of sensor $S_i$ after time $k$ and define $E_k \triangleq [E_{k,1}, \ldots, E_{k,m}]^T$. Without introducing conservatism, we also assume that the energy cost for $S_i$ sending a measurement packet to the fusion center is 1.

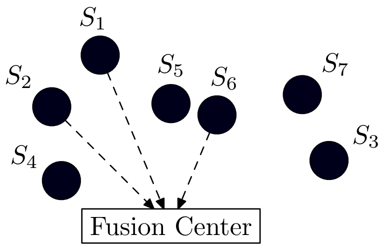


**Figure 12.1:** Sensors Selected at time $k$

Assume that the sensors start sending measurement from time 1. Let $\mathcal{S}_k,\ k = 1, 2, \ldots,$ be the set of sensors that are selected to transmit their measurements to the fusion center at time $k$. For example, in Figure 12.1, $\mathcal{S}_k = \{S_1, S_2, S_6\}$. A sensor selection schedule is defined as an infinite series of sensor selection strategy $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \ldots)$ and it is defined feasible if $E_{k,i} \geq 0,\ \forall k, i$, which means we are not using the sensors that have no power left.

Let $P_k$ denote the error covariance at the estimator at time $k$, which clearly depends on the set of data measurement received from time 1 to time $k$, and we indicate $P_k$ as $P_k(\mathcal{S})$ to underline the dependence on the sensor selection strategies.

Suppose it is required that $P_k \leq P_d$ for all $k$, where $P_d$ is a given positive definite matrix, which can be interpreted as a desired estimation accuracy level. The lifetime $L$ of the network under schedule $\mathcal{S}$ is defined to be

$$L(\mathcal{S}) \triangleq \min_k \{k : P_k(\mathcal{S}) \nleq P_d\} - 1. \tag{12.6}$$

The maximal lifetime of the network is defined as

$$\mathcal{L} \triangleq \sup_{\mathcal{S}\ feasible} L(\mathcal{S}). \tag{12.7}$$

The main goal is to find the optimal or suboptimal scheduling policy, i.e., determining $\mathcal{S}_k$ at each time $k$ such that the $L$ is maximized.

**LS** Include results from Mo et al. (sensor selection, cdc09 and ascc09).

## 12.3 Centralized Kalman Filtering Over a Static Sensor Tree

Consider the problem of state estimation over a wireless sensor network (Figure 12.2). The process dynamics is described by

$$x_{k+1} = Ax_k + w_k. \tag{12.8}$$

A wireless sensor network consisting of $N$ sensors $\{S_1, \cdots, S_N\}$ is used to measure the state. When $S_i$ takes a measurement of the state in Eqn (12.8), it returns

$$y_k^i = H_i x_k + v_k^i, \quad i = 1, \ldots, N. \tag{12.9}$$

In Eqn (12.8) and (12.9), $x_k \in \mathbb{R}^n$ is the state vector, $y_k^i \in \mathbb{R}^{m_i}$ is the observation vector for $S_i$, $w_{k-1} \in \mathbb{R}^n$ and $v_k^i \in \mathbb{R}^{m_i}$ are zero-mean white Gaussian random vectors with $\mathbb{E}[w_k w_j{}'] = \delta_{kj} Q \geq 0$, $\mathbb{E}[v_k^i v_t^i{}'] = \delta_{kt}\Pi_i > 0$, $\mathbb{E}[v_k^i v_t^j{}'] = 0\ \forall t, k$ and $i \neq j$, $\mathbb{E}[w_k v_t^i{}'] = 0\ \forall i, t, k$. We assume that $(A, \sqrt{Q})$ is controllable and $(C_{\text{all}}, A)$ is observable, where $C_{\text{all}} = [H_1; \cdots; H_N]$, i.e., the joint measurement matrix of all sensors.

Each sensor can potentially communicate via a single-hop connection with a subset of all the sensors by adjusting its transmission power. Let us introduce a fusion center which we denote as $S_0$, and consider a tree $T$ with root $S_0$ (see Figure 12.3). We suppose that there is a non-zero single-hop communication delay, which is smaller than the sampling time of the process. All sensors are synchronized in time, so the data packet transmitted from $S_i$ to $S_0$ is delayed one sample when compared with the parent node of $S_i$. We also assume that $S_i$ aggregates the previous time data packets from all its child nodes with its current time measurement into a single data packet.

Therefore only one data packet is sent from $S_i$ to its parent node at each time $k$.

Let us define the following state estimate and other quantities at $S_0$ for a given $T$:

$$\begin{aligned}
\hat{x}_k^-(T) &\triangleq \mathbb{E}[x_k|\text{all measurements up to } k-1],\\
\hat{x}_k(T) &\triangleq \mathbb{E}[x_k|\text{all measurements up to } k],\\
P_k^-(T) &\triangleq \mathbb{E}[(x_k - \hat{x}_k^-(T))(x_k - \hat{x}_k^-(T))'],\\
P_k(T) &\triangleq \mathbb{E}[(x_k - \hat{x}_k(T))(x_k - \hat{x}_k(T))'],\\
P_\infty^-(T) &\triangleq \lim_{k\to\infty} P_k^-(T), \text{if the limit exists},\\
P_\infty(T) &\triangleq \lim_{k\to\infty} P_k(T), \text{if the limit exists}.
\end{aligned}$$

We drop the dependence on $T$, i.e., we write $\hat{x}_k^-(T)$ as $\hat{x}_k^-$, etc., if there is no confusion on the underlying $T$. In this chapter, we are interested in computing $\hat{x}_k$ and $P_k$ for a given $T$.

### Optimal Estimation Over a Sensor Tree

Assume $T$ has depth $D$. Define $\mathcal{Y}_k^{k-i+1}$ as the set of all measurements available at the fusion center for time $k-i+1$ at time $k, i = 1, \cdots, D$. For

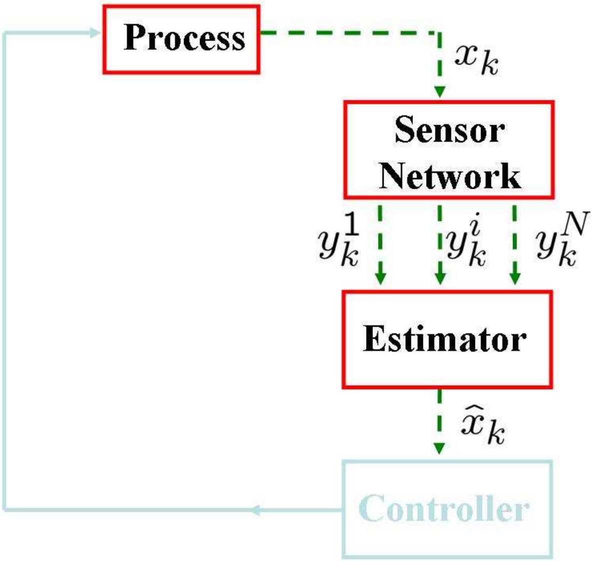


**Figure 12.2:** State Estimation Using a Wireless Sensor Network

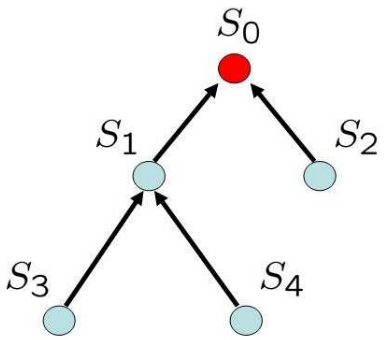


**Figure 12.3:** An Example of a Sensor Tree

the tree example in Figure 12.3, at time $k$, the fusion center has

$$\begin{aligned}\mathcal{Y}_k^k &= \{y_k^1, y_k^2\},\\ \mathcal{Y}_k^{k-1} &= \{y_{k-1}^1, y_{k-1}^2, y_{k-1}^3, y_{k-1}^4\}.\end{aligned}$$

We immediately notice that $\mathcal{Y}_{k-i}^{k-i} \subset \mathcal{Y}_k^{k-i}$, i.e., more measurements for time $k-i$ are collected at $k$ compared with at time $k-i$. For example, $\mathcal{Y}_{k-1}^{k-1} = \{y_{k-1}^1, y_{k-1}^2\}$ are the only available measurements at time $k-1$. However at time $k$, the available measurements for time $k-1$ changes to $\mathcal{Y}_k^{k-1}$. Hence we can obtain a better estimate of $x_{k-1}$ at time $k$ than at time $k-1$. This inspires us to recompute the optimal estimate of the previous states and use them as input to generate the current estimate. That is the basic idea contained in Theorem 12.2, where we recompute the optimal estimate of $x_{k-D+1}, \cdots, x_{k-1}$ at time $k$ and then make use of the updated estimates to compute the current estimate $\hat{x}_k$. Figure 12.4 shows the overall estimation scheme at time $k$.

Let $S_{i_j}$ be the node that is $j$ hops away from $S_0$. Define

$$\begin{aligned}\Gamma_j &\triangleq [H_{1_j}; H_{2_j}; \cdots], \quad j = 1, \cdots, D\\ C_i &\triangleq [\Gamma_1; \cdots; \Gamma_i], \quad i = 1, \cdots, D\\ \Upsilon_j &\triangleq \operatorname{diag}\{\Pi_{1_j}, \Pi_{2_j}, \cdots\}, \quad j = 1, \cdots, D\\ R_i &\triangleq \operatorname{diag}\{\Upsilon_1, \cdots, \Upsilon_i\}, \quad i = 1, \cdots, D.\end{aligned}$$

Intuitively, $\Gamma_j$ is the joint measurement matrix and $\Upsilon_j$ is the joint noise covariance from all sensors that are $j$ hops from the fusion center. $C_i$ is the joint measurement matrix, and $R_i$ is the joint noise covariance from all sensors that are $j$ or less than $j$ hops from the fusion center. With these definitions, the following theorem presents the optimal estimation algorithm over a sensor tree.

**Theorem 12.2.** *Consider a sensor tree $T$ with depth $D$.*

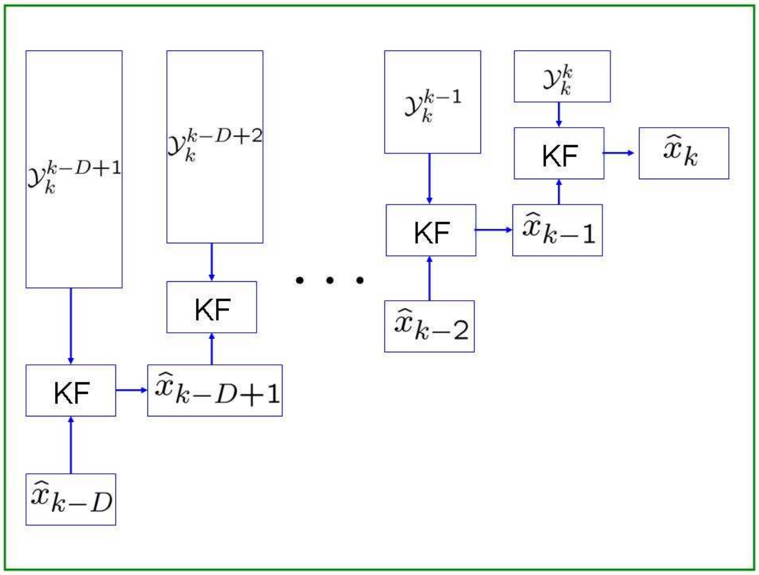


**Figure 12.4:** Kalman Filter Iterations at Time $k$

1. *$\hat{x}_k$ and $P_k$ can be computed from $D$ Kalman filters as*

$$\begin{aligned}(\hat{x}_{k-D+1}, P_{k-D+1}) &= \boldsymbol{KF}(\hat{x}_{k-D}, P_{k-D}, \mathcal{Y}_k^{k-D+1}, C_D, R_D)\\ &\vdots\\ (\hat{x}_{k-1}, P_{k-1}) &= \boldsymbol{KF}(\hat{x}_{k-2}, P_{k-2}, \mathcal{Y}_k^{k-1}, C_2, R_2)\\ (\hat{x}_k, P_k) &= \boldsymbol{KF}(\hat{x}_{k-1}, P_{k-1}, \mathcal{Y}_k^{k}, C_1, R_1).\end{aligned}$$

2. *$P_\infty^-$ and $P_\infty$ satisfy*

$$P_\infty^- = g_{C_2} \circ \cdots \circ g_{C_{D-1}}(P^*), \tag{12.10}$$

$$P_\infty = \tilde{g}_{C_1} \circ g_{C_2} \circ \cdots \circ g_{C_{D-1}}(P^*), \tag{12.11}$$

*where $P^*$ is the unique solution to $g_{C_D}(P^*) = P^*$.*

*Proof.* 1) We know that the estimate $\hat{x}_k$ is generated from the estimate of $\hat{x}_{k-1}$ together with all the available measurements at time $k$ through a traditional Kalman filter. Similarly, the estimate $\hat{x}_{k-1}$ is generated from the estimate of $\hat{x}_{k-2}$ together with all the available measurements for time $k-1$ at time $k$, etc. This recursion for $D$ steps corresponds to the $D$ Kalman filters stated in the theorem.
2) Follows directly from Kalman filter recursive equations. □

The estimation algorithm presented in Theorem 12.2 readily extends to a general graph that represents the sensor communications. The fusion center only needs to keep track of the measurement data up to previous time $k - D + 1$. Thus in a distributed setting, every node acts as a fusion center and the system robustness (against sensor failure) is increased.

### Example

We consider an integrator chain in this section. The discrete time system dynamics is given by Eqn (12.8) with

$$A = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}.$$

and with process noise covariance $Q = 0.3I$. There are two sensors available. The measurement equations are given by

$$\begin{aligned} y_k^1 &= [\ 0 \quad 1\ ]x_k + v_k^1 = H_1 x_k + v_k^1, \\ y_k^2 &= [\ 1 \quad 0\ ]x_k + v_k^2 = H_2 x_k + v_k^2, \end{aligned}$$

with covariances $\Pi_1 = 0.25$ and $\Pi_2 = 0.5$. Consider the following two sensor topologies (Figure. 12.5).

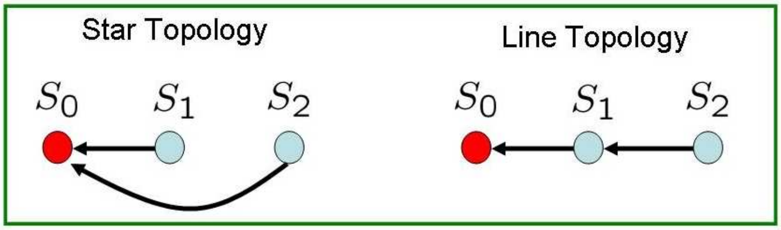


**Figure 12.5:** Integrator Chain Example

The first one is the star topology, i.e., the two sensors communicate with the fusion center directly, which corresponds to the centralized Kalman filter. The second one is a line topology (a special tree), and the measurement data from sensor two to the fusion center get delayed by one step. For this example, it is easy to calculate that

$$P^* = \begin{bmatrix} 0.1838 & 0.0103 \\ 0.0103 & 0.1822 \end{bmatrix},$$

which is the unique solution to $P^* = g_{[H_1;H_2]}(P^*)$. As a result, for the star topology,

$$P_\infty(\text{star}) = \tilde{g}_{[H_1;H_2]}(P^*) = \begin{bmatrix} 0.0825 & 0.0021 \\ 0.0021 & 0.0822 \end{bmatrix},$$

with $\mathrm{Tr}\big(P_\infty(\text{star})\big) = 0.1647$. For the line topology,

$$P_\infty(\text{line}) = \tilde{g}_{[H_1]}(P^*) = \begin{bmatrix} 0.1835 & 0.0047 \\ 0.0047 & 0.0823 \end{bmatrix},$$

with $\text{Tr}\big(P_\infty(\text{line})\big) = 0.2658$.

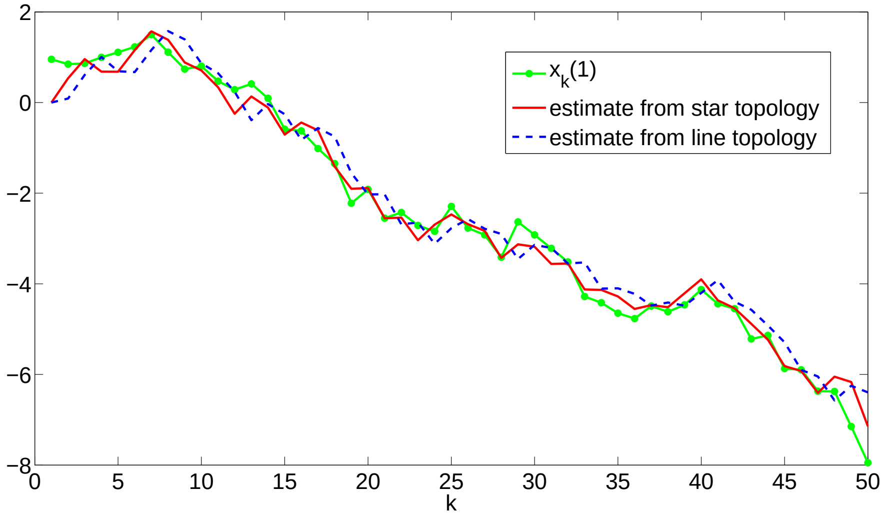


**Figure 12.6:** True State and its Estimates

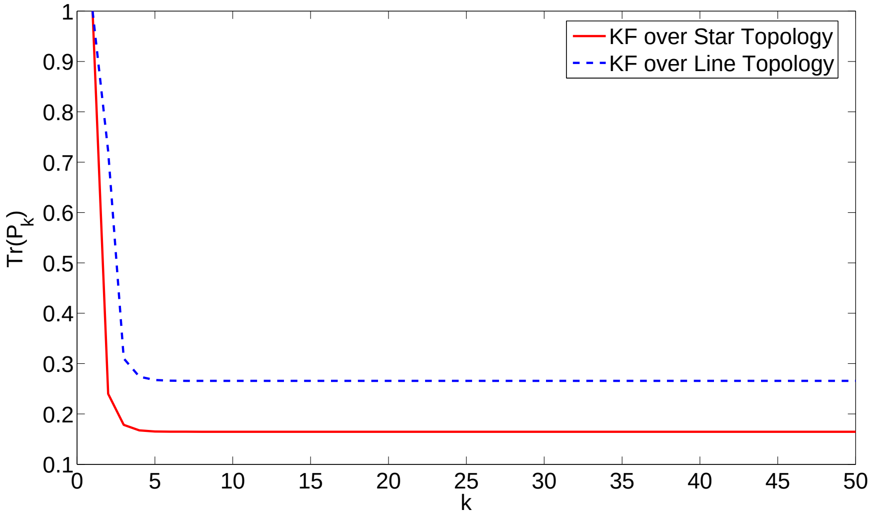


**Figure 12.7:** Error Covariances

We plot the first component of the true state and its estimates based on the two sensor topologies in Figure 12.6. We also plot the corresponding error covariance in Figure. 12.7. As those figures demonstrate, the simulations agree well with the theory developed.

## Applications

### *Sensor Tree Performance Comparison*

Consider a tree $T$ of depth $D$ with root at $S_0$. In Theorem 12.2, we have shown that the steady-state error covariance matrix can be found in an exact form as

$$P_\infty(T) = \tilde{g}_{C_1(T)} \circ g_{C_2(T)} \circ \cdots \circ g_{C_{D-1}(T)}\big(P^*(T)\big) \tag{12.12}$$

and $P^*(T)$ is the unique solution to $g_{C_D(T)}\big(P^*(T)\big) = P^*(T)$.

In many cases, we are interested in finding a tree $T$ that has some desired properties, such as it has the minimum error covariance $\overline{P}(T)$. This often involves comparing two trees $T_1$ and $T_2$. In general, since $\overline{P}(T) \in \mathbb{S}^n_+$ where only partial order exists, it may not always hold that either $\overline{P}(T_1) \le \overline{P}(T_2)$ or $\overline{P}(T_2) \le \overline{P}(T_1)$. However, in many circumstances, we are still able to compare the performance of two trees. We first prove the following lemma.

**Lemma 12.1.** *Assume $1 \le l_1 \le l_2 \le D$ and $P \in \mathbb{S}^n_+$. Then*

$$C'_{l_1}[C_{l_1} P C'_{l_1} + R_{l_1}]^{-1} C_{l_1} \le C'_{l_2}[C_{l_2} P C'_{l_2} + R_{l_2}]^{-1} C_{l_2}. \tag{12.13}$$

*Proof.* We first prove the case $l_1 = 1$ and $l_2 = 2$. Note that we write Eqn (12.13) as

$$\begin{aligned} C'_1[C_1 P C'_1 + R_1]^{-1} C_1 &\le \begin{bmatrix} C_1 \\ \Gamma_2 \end{bmatrix}' \left[ \begin{bmatrix} C_1 \\ \Gamma_2 \end{bmatrix} P \begin{bmatrix} C_1 \\ \Gamma_2 \end{bmatrix}' + R_2 \right]^{-1} \begin{bmatrix} C_1 \\ \Gamma_2 \end{bmatrix} \\ &= \begin{bmatrix} C_1 \\ \Gamma_2 \end{bmatrix}' \begin{bmatrix} B & M \\ M' & G \end{bmatrix}^{-1} \begin{bmatrix} C_1 \\ \Gamma_2 \end{bmatrix}, \end{aligned}$$

where $B = C_1 P C'_1 + R_1$, $G = \Gamma_2 P \Gamma'_2 + \Upsilon_2$, and $M = C_1 P \Gamma'_2$. Since $B > 0, G > 0$, and

$$\begin{bmatrix} B & M \\ M' & G \end{bmatrix} > 0,$$

the Schur complement $S_B \triangleq B - M G^{-1} M' > 0$. Therefore by performing block matrix inversion, we obtain

$$\begin{aligned} \begin{bmatrix} C_1 \\ \Gamma_2 \end{bmatrix}' \begin{bmatrix} B & M \\ M' & G \end{bmatrix}^{-1} \begin{bmatrix} C_1 \\ \Gamma_2 \end{bmatrix} &= \begin{bmatrix} C_1 \\ \Gamma_2 \end{bmatrix}' \begin{bmatrix} X_1 & -B^{-1} M S_B^{-1} \\ -S_B^{-1} M' B^{-1} & S_B^{-1} \end{bmatrix} \begin{bmatrix} C_1 \\ \Gamma_2 \end{bmatrix} \\ &= C'_1 B^{-1} C_1 + X_2 X'_2 \ge C'_1 B^{-1} C_1, \end{aligned}$$

where $X_1 = B^{-1} + B^{-1} M S_B^{-1} M' B^{-1}$ and $X_2 = C'_1 B^{-1} M S_B^{-\frac{1}{2}} - \Gamma'_2 S_B^{-\frac{1}{2}}$. Having proved the case $i = 1, j = 2$, the general case easily follows if we write $C_1 := C_{l_1}$ and $\Gamma_2$ is such that $C_{l_2} = [C_{l_1}; \Gamma_2]$. □

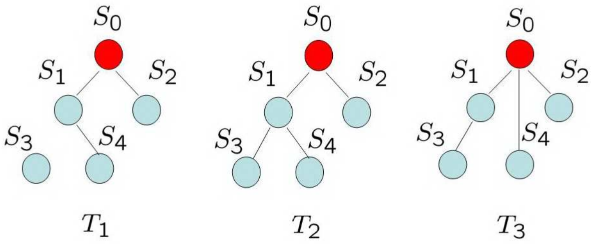


**Figure 12.8:** Comparison of Three Sensor Trees

**Corollary 12.2.1.** *For all $l = 1, \cdots, D-1$, and all $X \geq 0$, $g_{C_{l+1}}(X) \leq g_{C_l}(X)$.*

**Corollary 12.2.2.** *For all $l = 1, \cdots, D-1$, and all $X \geq 0$, $\tilde{g}_{C_{l+1}}(X) \leq \tilde{g}_{C_l}(X)$.*

We can interpret Corollary 12.2.1 and 12.2.2 in the following sense. For an estimator, the more information it has (i.e., more sensors) and the less delay the measurement data arrive, the more accurate it can estimate the process state.

For a given tree $T$, define

$$\mathcal{S}_{l-hop}(T) \triangleq \{S_i : S_i \text{ is within } l\text{−hops away from } S_0\} \tag{12.14}$$

for $l = 1, \cdots, D$. For example, in Fig. 12.8, $\mathcal{S}_{1-hop}(T) = \{S_1, S_2\}$, and $\mathcal{S}_{2-hop}(T) = \{S_1, S_2, S_3, S_4\}$.

**Theorem 12.3.** *For two trees $T_1$ and $T_2$, if $\mathcal{S}_{l-hop}(T_1) \subset \mathcal{S}_{l-hop}(T_2)\ \forall\ l = 1, \cdots, D$, then $\overline{P}(T_2) \leq \overline{P}(T_1)$.*

*Proof.* Since $\mathcal{S}_{j-hop}(T_1) \subset \mathcal{S}_{j-hop}(T_2)\ \forall\ j = 1, \cdots, D$, from Lemma 12.1, we have $g_{C_{j-1}(T_1)} \geq g_{C_{j-1}(T_2)}$ and $\tilde{g}_{C_{j-1}(T_1)} \geq \tilde{g}_{C_{j-1}(T_2)}$. Therefore the theorem follows immediately from Eqn (12.12). □

**Corollary 12.3.1.** *If $T_1 \subset T_2$, then $\overline{P}(T_2) \leq \overline{P}(T_1)$.*

These results provide an easy to compare the performance of different sensor trees. For example, consider the three sensor trees in Fig. 12.8. Apparently, $T_1 \subset T_2$, and $\mathcal{S}_{j-hop}(T_2) \subset \mathcal{S}_{j-hop}(T_3)$, $j = 1, 2$, therefore from Theorem 12.3 and Corollary 12.3.1, we immediately obtain

$$\overline{P}(T_3) \leq \overline{P}(T_2) \leq \overline{P}(T_1).$$

### *Minimum-energy Sensor Tree*

In [?], Shi et al. first considered the problem of minimizing sensor energy usage while guaranteeing a desired level of estimation quality at the fusion

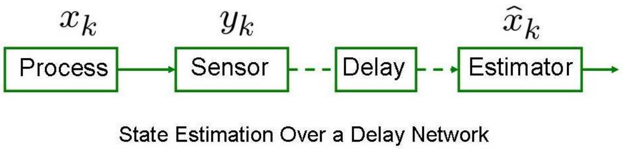


**Figure 12.9:** Estimation over a Packet-delaying Network

center. Let $e(T)$ denote the total energy cost (i.e., transmission energy and receiving energy, etc.) when the sensor communication with $S_0$ is represented by $T$. Further denote $\mathcal{T}_{\text{all}}$ as the set of all trees with depth $D$ that are rooted at $S_0$. The following problem is then considered.

$$\min_{T\in\mathcal{T}_{\text{all}}} e(T)$$

subject to

$$\overline{P}(T) \le P_{\text{desired}}$$

where $P_{\text{desired}} \ge 0$ is given. The result in Theorem 12.2 was used to guide the construction of the minimum energy sensor tree. The basic idea is as follows. If $\overline{P}(T) \nleq P_{\text{desired}}$, then $T$ is reconfigured to $T'$ by connecting a sensor that is currently two-hops away from $S_0$ directly to $S_0$. It is shown that $\overline{P}(T') \le \overline{P}(T)$, and within at most $N$ steps, $\overline{P}(T') \le P_{\text{desired}}$, i.e., $T'$ is now a feasible solution. A minimum energy subtree algorithm is then run on $T'$ to further reduce the energy cost.

### *From Packet Delay to Packet Drop*

Consider the problem of state estimation over a packet-delaying network as seen from Fig. 12.9. The process dynamics is the same as in Eqn (12.8) and sensor measurement equation is given by

$$y_k = Cx_k + v_k. \tag{12.15}$$

After taking a measurement at time $k$, the sensor sends $y_k$ to a remote estimator for generating the state estimate. We assume that the measurement data packets from the sensor are to be sent across a packet-delaying network to the estimator. Each $y_k$ is delayed by $d_k$ times, where $d_k$ is a random variable described by a probability mass function $f$, i.e.,

$$f(j) = \mathbf{Pr}[d_k = j], j = 0, 1, \cdots \tag{12.16}$$

We assume $d_{k_1}$ and $d_{k_2}$ are independent if $k_1 \ne k_2$, and the estimator discards any data $y_k$ (or $\hat{x}_k^s$) that are delayed by $D$ times or more.

Given the system and the network delay models in Eqn (12.8), and Eqn (12.15)–(12.16), we are interested in computing $\mathbf{Pr}[P_k \le M]$, the probability that $P_k$ is bounded by a given matrix $M \in \mathbb{S}_+^n$. The probabilistic

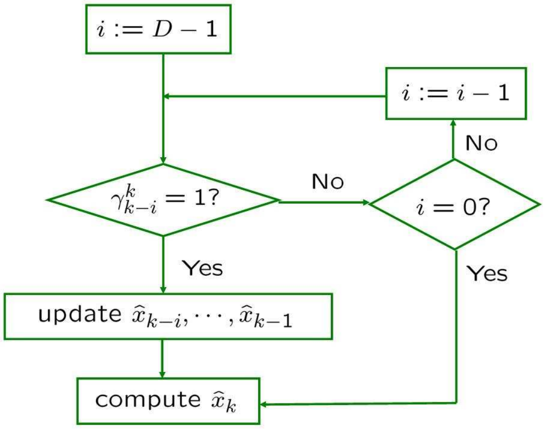


**Figure 12.10:** Recursive Kalman filtering

metric was proposed in [**?**] for state estimation over packet-dropping networks.

The recursive Kalman filtering technique from Theorem 12.2 dealing with delayed measurement provides a promising way to bridge the gap between packet drop analysis and packet delay analysis. The basic ideas is as follows. Since $y_{k-i}$ may arrive at time $k$, we can improve the estimation quality by recalculating $\hat{x}_{k-i}$ utilizing the new available measurement $y_{k-i}$. Once $\hat{x}_{k-i}$ is updated, we can update $\hat{x}_{k-i+1}$ in a similar fashion. Fig. 12.10 illustrates the idea, where $\gamma_{k-i}^{k} = 1$ or $0$ is the indicator function whether the measurement packet generated at time $k-i$ arrives at time $k$ or not.

Define $\gamma_{k-i} \triangleq \sum_{j=0}^{i} \gamma_{k-i}^{k-j}$, i.e., $\gamma_{k-i}$ indicates whether $y_{k-i}$ is received by the estimator at or before $k$, and define $\hat{\gamma}_i(D)$ as

$$\hat{\gamma}_i(D) \triangleq \begin{cases} \sum_{j=0}^{i} f(j), \text{if } 0 \le i < D, \\ \sum_{j=0}^{D-1} f(j), \text{if } i \ge D. \end{cases}$$

Then it is easy to verify that for a fixed $D$,

$$\mathbf{Pr}[\gamma_{k-i} = 1] = \hat{\gamma}_i(D). \tag{12.17}$$

Notice that now $\mathbf{Pr}[\gamma_{k-i} = 1]$ becomes a constant, thus given a stochastic description of the packet delays in Eqn (12.16), we can convert the packet delay model into a packet drop model. Similar to [**?**], we are then able to obtain similar bounds on $\mathbf{Pr}[P_k \le M]$ using the corresponding new packet arrival rate $\hat{\gamma}_i(D)$.

## 12.4 Distributed Control over Sensor Networks

LS include main ideas from the following paper.

## 12.5 Further Reading

## 12.6 Exercise

# Bibliography

[AK86] R. Akella and P. R. Kumar. Optimal control of production rate in a failure prone manufacturing system. *IEEE Transactions on Automatic Control*, AC-31(2):116–126, February 1986.

[CPR01] P. R. Chandler, M. Pachter, and S. Rasmussen. UAV cooperative control. In *Proc. American Control Conference*, pages 50–55, 2001.

[Del06] D. Del Vecchio. Discrete dynamic feedback for a class of hybrid systems on a lattice. In *IEEE International Symposium on Computer-Aided Control Systems Design*, 2006.

[DM06] W. B. Dunbar and R. M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

[DMK06] D. Del Vecchio, R. M. Murray, and Erik Klavins. Discrete state estimators for systems on a lattice. *Automatica*, 42(2):271–0285, 2006.

[ED05] M. G. Earl and R. D'Andrea. Multi-vehicle cooperative control using mixed integer linear programming. Technical Report arXiv:cs.RO/0501092, `http://arXiv.org`, 2005.

[FM04] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(5):1465–1476, 2004.

[HRW$^+$04] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin. The Stanford testbed of autonomous rotorcraft for multi-agent control (STARMAC). In *AIAA Digital Avionics Systems Conference*, 2004.

[Jin06] Z. Jin. *Coordinated Control of Networked Multi-Agent Systems*. PhD thesis, California Institute of Technology, Electrical Engineering, 2006.

[JLM03] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of grups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[JM03] Z. Jin and R. M. Murray. Stability and performance analysis with double-graph model of vehicle formations. In *Proc. American Control Conference*, 2003.

[JM04] Z. Jin and R. M. Murray. Double-graph control strategy of multi-vehicle formations. In *Proc. IEEE Control and Decision Conference*, 2004.

[KKAH04] E. King, Y. Kuwata, M. Alighanbari, and J. How. Coordination and control experiments for uav teams. *Advances in the Astronautical Sciences*, 118:145–155, 2004.

[KM04] E. Klavins and R. M. Murray. Distributed computation for cooperative control. *IEEE Pervasive Computing*, 3(1):56–65, 2004.

[Lav02] E. Lavretsky. F/a-18 autonomous formation flight control systems design. In *AIAA Conference on Guidance, Navigation, and Control*, pages AIAA 2002–4757, 2002.

[LF01] N. E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proc. IEEE Control and Decision Conference*, pages 2968–2973, 2001.

[Mur03] R. M. Murray, editor. *Control in an Information Rich World: Report of the Panel on Future Direcitons in Control, Dynamics and Systems*. SIAM, 2003. Available at `http://www.cds.caltech.edu/~murray/cdspanel`.

[OFL04] P. Ogren, E. Fiorelli, and N. E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302, 2004.

[OS06] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[OSM02] R. Olfati-Saber and R. M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *Proc. IFAC World Congress*, 2002.

[OSM04] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[RBTH02] A. Richards, J. Bellingham, M. Tillerson, and J. How. Co-ordination and control of multiple UAVs. In *AIAA Conference on Guidance, Navigation, and Control*, 2002.

[Rey87] C. W. Reynolds. Herds, and schools: A distributed behavioral model. *Computer Graphics (SIGGRAPH '87 Conference Proceedings)*, 21(4):25–34, 1987.

[SCPP03] C. Schumacher, P. Chandler, M. Pachter, and L. Pachter. UAV task assignment with timing constraint. In *AIAA Conference on Guidance, Navigation, and Control*, 2003.

[SH96] D. Swaroop and J. K. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(3):349–357, 1996.

[SKS03] D. H. Shim, H. J. Kim, and S. Sastry. A flight control system for aerial robots: Algorithms and experiments. *Control Engineering Practice*, 11:1389–1400, 2003.

[SRSP06] T. Shima, S. Rasmussen, A. Sparks, and K. Passino. Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms. *Computers and Operations Research*, 33(11):3252–3269, 2006.

[SV99] D. Schrage and G. Vachtsevanos. Software enabled control for intelligent uavs. In *Proc. IEEE International Conference on Control and Applications*, 1999.

[TFI$^+$04] A. Tiwari, J. Fung, J. M. Carson III, R. Bhattacharya, and R. M. Murray. A framework for Lyapunov certificates for multi-vehicle rendezvous problems. In *Proc. American Control Conference*, 2004.

[TJJM05] A. Tiwari, M. Jun, D. E. Jeffcoat, and R. M. Murray. The dynamic sensor coverage problem. In *Proc. IFAC World Congress*, 2005.

[TPS98] C. Tomlin, G. J. Pappas, and S. Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, 1998.

[VSR$^+$04] V. Vladimerou, A. Stubbs, J. Rubel, A. Fulford, and G.E. Dullerud. A hovercraft testbed for decentralized and cooperative control. In *Proc. American Control Conference*, 2004.

Go through and replace all citations to conference papers with citations to the archival literature (or remove the citation from the main text). **RMM**

# Index