KTH Electrical Engineering

State-based Channel Access for a Network of Control Systems

CHITHRUPA RAMESH

Doctoral Thesis
Stockholm, Sweden 2014

KTH Royal Institute of Technology
School of Electrical Engineering
Automatic Control Lab
SE-100 44 Stockholm
SWEDEN

TRITA-EE 2014:016
ISSN 1653-5146
ISBN 978-91-7595-092-1

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i reglerteknik fredagen den 25 April 2014, klockan 10:00 i sal E3, Kungliga Tekniska högskolan, Osquarsbacke 14, Stockholm.

Tryck: Universitetsservice US AB

Abstract

Wireless networked control systems use shared wireless links to communicate between sensors and controllers, and require a channel access policy to arbitrate access to the links. Existing multiple access protocols perform this role in an agnostic manner, by remaining insular to the applications that run over the network. This approach does not give satisfactory control performance guarantees. To enable the use of wireless networks in emerging industrial applications, we must be able to systematically design wireless networked control systems that provide guaranteed performances in resource-constrained networks.

In this thesis, we advocate the use of state-based channel access policies. A state-based policy uses the state of the controlled plant to influence access to the network. The state contains information about not only the plant, but also the network, due to the feedback in the system. Thus, by using the state to decide when and how frequently to transmit, a control system can adapt its contribution to the network traffic, and enable the network to adapt access to the plant state. We show that such an approach can provide better performance than existing methods. We examine two different state-based approaches that are distributed and easy to implement on wireless devices: event-based scheduling and adaptive prioritization.

Our first approach uses events to reduce the traffic in the network. We use a state-based scheduler in every plant sensor to generate non-coordinated channel access requests by selecting a few critical data packets, or events, for transmission. The network uses a contention resolution mechanism to deal with simultaneous channel access requests. We present three main contributions for this formulation. The first contribution is a structural analysis of stochastic event-based systems, where we identify a dual predictor architecture that results in separation in design of the state-based scheduler, observer and controller. The second contribution is a Markov model that describes the interactions in a network of event-based systems. The third contribution is an analysis of the stability of event-based systems, leading to a stabilizing design of event-based policies.

Our second approach uses state-based priorities to determine access to the network. We use a dominance protocol to evaluate priorities in a contention-based setting, and characterize the resulting control performance. An implementation and evaluation of this channel access mechanism on sensor nodes is also presented.

The thesis finally examines the general networked control problem of jointly optimizing measurement and control policies, when a nonlinear measurement policy is used to perform quantization, event-triggering or companding. This contribution focuses on some of the fundamental aspects of analyzing and synthesizing control systems with state-based measurement policies in a more generalized setting. We comment on the dual effect, certainty equivalence and separation properties for this problem. In particular, we show that it is optimal to apply separation and certainty equivalence to a design problem that permits a dynamic choice of the measurement and control policies.

प्रज्ञानम् ब्रह्म ।

Acknowledgements

I would like to acknowledge those who have supported me through the writing of this thesis. Foremost among them is my advisor, Karl Henrik Johansson, for giving me this incredible learning opportunity, and for making it possible for me to complete my studies. You have been a perpetual source of inspiration, in technical matters and beyond. Your research vision and experience have been a gentle guide in defining my approach and interests over the last few years. Your direct and generous style has provided me with a managerial model to aspire to. I am grateful to you for the support, all the advice and the many opportunities you have made available to me.

My co-advisor, Henrik Sandberg, has been a big support in my attempts to learn, analyze and write control theory. Thank you for generously sharing your knowledge, and for patiently entertaining all my questions in mathematics, stochastic control, and everything else. Your technical reach is astounding; I hope to aspire to at least some of your knowledge, and all of your assiduousness. I have thoroughly enjoyed discussing my research with you.

I would like to thank Emilio Frazzoli for hosting me in the ARES group with the Laboratory for Information and Decision Systems at MIT. It was an exciting experience, and I am thankful to Mukul Agarwal, Saurabh Amin and Ketan Savla for the discussions during my visit. I would also like to thank Maurice Heemels for the opportunity to visit and interact with his group at Eindhoven. I'm immensely grateful to Sanjoy Mitter and John Baras for the encouragement and the many inspiring suggestions and references.

Several of my collaborators have played an important role in my learning. I am grateful to Maben Rabi for providing me with interesting leads and directions throughout my studies. All those pointers to papers, books and authors have definitely shaped my thoughts and interests. I have learnt a lot from working with Phoebus Chen, especially from his attention to detail. I enjoyed our long discussions on routing and mesh networks. José Araujo has time and again provided me immense support with implementations and experiments on the test-bed. Many thanks to my other co-authors, Lei Bao and Dick Jenkins, for the collaboration, discussions and reviews.

The Automatic Control Lab at KTH has provided a warm and welcoming environment away from home for me. I am thankful to Pangun for convincing me to apply to this lab. I have enjoyed all the time spent with my colleagues, in Stockholm and at conferences and workshops. Special thanks to Alireza, André, Assad, Burak, Cátia, Euhanna, Farhad, José, Pangun and PG for the wonderful company over the years, and to Damiano, Iman, Jana, Jeff, Jim, Pablo, Phoebus and Ubaldo for all the good times we have shared. Conversations with Christian, Erik, Oscar and Torbjön, were always welcome breaks. It is impossible to call out to everyone I have come to know in the lab over the years, but I thank each of them for the excellent company. Thanks also to Bart, Maben, André, Pangun, Jana and Junfeng,

for reviewing various parts of this thesis. I am grateful to Aitor and Lin for helping me with experiments. Many thanks to Anneli, Hanna, Karin, Kristina and Niclas for their generous help with everything, and for the wonderful spirit they bring to our lab.

I am grateful to the Swedish Research Council, the Knut and Alice Wallenberg Foundation, the Swedish Governmental Agency for Innovation Systems (VINNOVA) through the WiComPI project and the EU project SOCRADES for financially supporting my research.

Thanks to Priti, Neha and Marianna for the support and company in Stockholm. Shilpa and Archana, thanks for the vital intercontinental link. I am indebted to G. Shridhar for the mentoring and encouragement over the years.

I would like to thank my vast family, for cheering me up everyday over skype, phone and email, especially through this last phase of my thesis. I am very fortunate to have all the grandparents, Ajji, Audha, Thatha and Thathi, around to encourage and inspire me even today. I'm deeply grateful to my parents-in-law, Jahnavi and Srinivasan, for all the support they have given me. Sharan and Shilpa, and Sunil and Sangamithra, thanks for your company and affection. I'm thankful to my dearest mother, Sujatha, and I cherish every memory of my father, Ramesh, for the unwavering support and encouragement. And to my dearest husband, Sandeep, I could not have made it without you, and without the humour and love you bring to my life. Where you are is home to me. Thank you for being with me through every step of this journey.

Chithrupa

Contents

Chapter 1

Introduction

Information is essential to feedback control. We use evidence and information to make decisions in our everyday lives, as informed decisions can outdo even the best guesswork. Similarly, a control system uses measurements to compute a control action. When applied to a physical process, the control action has the potential to modify the behaviour of the process in a desired manner. Feedback control is a hidden component of many engineering solutions, from boilers to autopilots for spacecrafts. We use feedback control to regulate a physical process, or to automate a manual decision process, with an agility, accuracy and computational capacity that far surpasses human capability. Conceptually, a control system comprises of four main components, as depicted in Figure 1.1: the physical process or plant to be controlled, a sensor to take measurements, a controller to generate the control signal and an actuator to apply the control signal to the plant. Underlying the functional blocks in a feedback loop are the information channels that enable feedback control; a dedicated sensing link that conducts measurements to the controller and a dedicated actuation link that conducts control signals to the actuator.

Our ability to gather information necessary for decision-making is an important factor that determines the use of feedback control to engineer the world around us. If we could measure and collate information regarding the moisture content of soil, we could irrigate fields more efficiently and potentially increase our agricultural output. With knowledge of our vital signs, we might be able to train better and recover more efficiently, leading to fewer sports injuries or better fitness and even weight loss strategies. By learning the energy consumption of all the devices in our homes, we might be able to reduce our energy consumption by choosing to charge or run these devices during non-peak hours. All these applications require extensive information gathering from numerous, possibly mobile, sources that need not be located near each other. The impossibility of providing a dedicated, wired link from a sensor at each source location, had relegated such applications to the realm of futuristic dreams.

Fortunately, we have found new ways to harvest and collate information. Wireless sensor nodes have given us the ability to embed a little intelligence, some

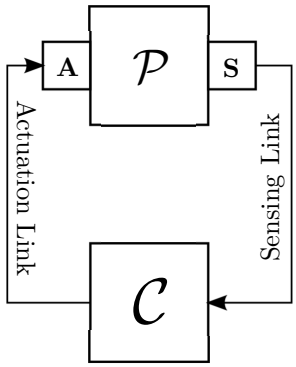


Figure 1.1: A control system, at its simplest, comprises of a plant ($\mathcal{P}$), sensor ($\mathbf{S}$), controller ($\mathcal{C}$) and actuator ($\mathbf{A}$). The sensing and actuation links transmit information that enables feedback control. In classical control, these are inherently assumed to be dedicated links.

sensing capability and a wireless transceiver in almost any object around us (Akyildiz et al., 2002; Chong and Kumar, 2003). Wireless and internet connectivity now encompasses the globe, and our computing capability and storage capacity are better and cheaper than ever before. The Internet-of-Things makes every device capable of networking, and even physical objects can be addressable and reachable via communications (Atzori et al., 2010). These technologies promise ubiquitous and inexpensive computation, communication and sensing abilities, enabling us to harvest information about everything around us. Favourably, these technologies are almost ready and already in use. A plethora of applications from military and biomedical sensing, environmental and building monitoring, and home surveillance use it to monitor the world around us.

Now, paucity of information is no longer a constraint for control. The deluge of data can be used to make decisions and compute control actions. However, by advancing from monitoring to actuation, we arrive at a new paradigm. Future applications may now anticipate the ability to interact with, and expand the capabilities of, the physical world through computation, communication, and control. The idea of designing a non-interacting system or physical component appears outdated. Every system or physical component must be able to be integrated with information relevant to its functioning, and capable of reacting to this information. Thus, physical systems come together with information networks to perform sensing and actuation, resulting in a *cyber-physical system* (Lee, 2008; Rajkumar et al., 2010).

Many exciting applications have been conceived using this paradigm. Intelligent transport systems envision sensors fitted on automobiles that can transmit road, wind or traffic conditions to other vehicles. Such information can also be collated to generate route guidance and traffic warnings to all motorists. Smart bio-sensors can be used to constantly monitor critical patients and release medication, through

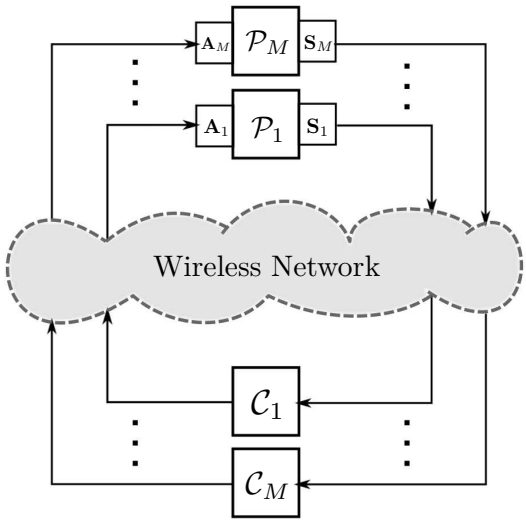


Figure 1.2: In wireless NCSs, the sensing and actuation links from all the control systems are aggregated into a shared wireless network. A wireline NCS uses a communication bus in place of the wireless network.

bio-patches, when necessary. This has the potential to bring down medical care costs in an ageing society and enable independent living for those with manageable conditions. Assisted living technology and environmental control can allow near-normal interaction with smart homes and offices for the disabled. Future retail environments and public areas can be designed as smart spaces, capable of interacting with visitors and providing automated customer service. Fewer constraints on sensing, communication and actuation can also result in big improvements in robotics, factory automation and process control.

To execute this vision to its utmost potential, many of these applications require the exchange of information within a control system to occur over a wireless network. This is a challenging problem. The wireless medium is notorious for its unreliability, as anyone who has experienced dropped calls, or a slow WiFi connection, is well aware of. So, how do we design control systems capable of providing performance guarantees to run over wireless networks? We examine this question in the next section.

1.1 Wireless Networked Control Systems

A network of control systems with wireless-in-the-loop is depicted in Figure 1.2. The sensing and actuation links of a wireless *networked control system* (NCS) are accommodated within a shared wireless network, which is also used by the other

control systems in the network. A wireline NCS uses a communication bus in place of the wireless network. The communication bus provides a wired medium, but it is not a dedicated link, as it is shared between all the control systems in the network.

There are many challenges in designing a wireless NCS. However, the benefits of wireless outweigh these challenges. Wireless networks eliminate costs incurred in the installation and maintenance of wiring. Wireless sensor networks are easy to deploy, making it possible to modify existing systems by adding sensors on the fly. Consequently, they offer possibilities of better sensing, which could lead to better control. Wireless is sometimes the only choice, for example, in a network of mobile agents, or in areas where infrastructure is difficult to install. A wireless system is an enabler of our information-driven world, where decisions can be taken anywhere and reach everywhere. There is also a compulsion to follow the trend; if wireless sensors will soon be ubiquitous, how can control systems remain wired?

Thus, future applications have created a need for wireless in the feedback loop, and the technology to fulfill that need already exists. Then, can we plug wireless nodes into existing control systems, and achieve working solutions? No. There is a disconnect between classical control theory and wireless communication. Classical control theory focuses on every aspect necessary to accomplish control action, barring the informational transactions within the closed-loop. This is because dedicated, near-ideal transmission links appear transparent to a control system. As we replace these dedicated connections with shared links, the imperfections of the communication media can no longer be ignored. Wireless transmissions are susceptible to much higher packet losses and delays than wireline transmissions, due to the nature of the wireless medium. What should be done when the measurement packet does not arrive or the control packet is not delivered? Wireless bandwidth is scarce. How do we transmit real-valued measurements and controls through such links? Wireless networks are interference-constrained. How do we accommodate simultaneous transmission requests from different control systems in the network? Thus, control theory must grow to encompass a networked view of systems, as required to deal with wireline and wireless control systems.

In the last decade, networked control theory has taken on this challenge, and provided us with solutions to some of the above issues. The EU project SOCRADES (Service-Oriented Cross-layer infRAstructure for Distributed smart Embedded devices) used some of these solutions to integrate wireless nodes into a process control setup. As part of this project, wireless control experiments were conducted in a mining plant at Boliden in Sweden. The froth-flotation process in this plant is shown in Figure 1.3. The design of the control system had to overcome a harsh radio environment that introduced delays and packet losses in the wireless communication between the nodes.

However, some key challenges with using a shared medium remain unsolved. A wireless link cannot support simultaneous transmissions from different users. The network requires a channel access policy to ensure that no more than one control system transmits data at a given time, over a shared link. This policy must also play a role in determining how often each control system in the network transmits.

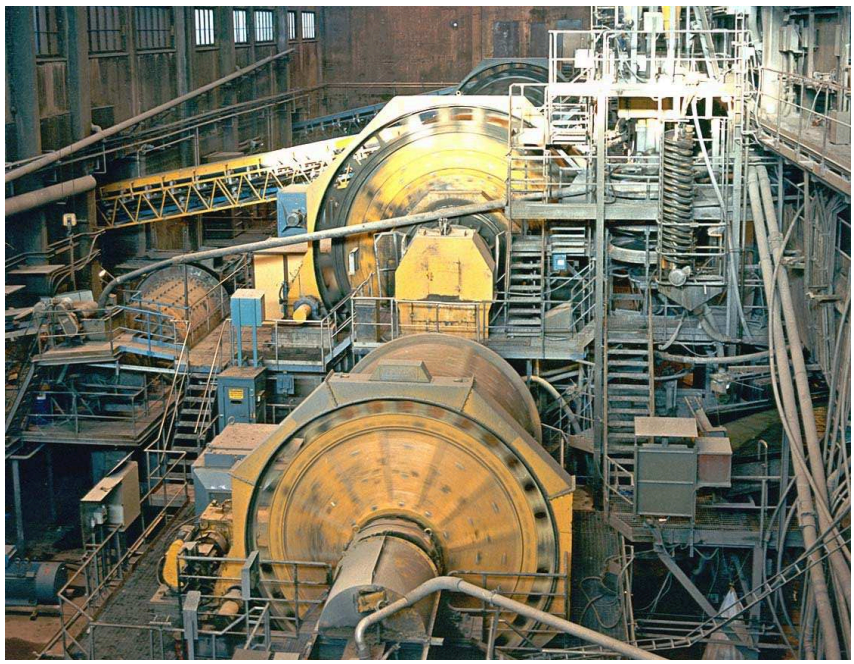


Figure 1.3: Wireless sensors were introduced in the froth-flotation processes at the mining plant in Boliden. This is a real example of wireless networked control, carried out in the EU project SOCRADES. (Courtesy of Boliden)

This is because a wireless link can only transmit a limited amount of data at a given time. If a control system monopolizes a shared link, there is clearly no way to accommodate any other user in this network. In other words, the finite capacity of a wireless link corresponds to a scarcity of a shared resource and the fundamental challenge here is to identify an efficient resource allocation policy. So, how should we allocate a share of the common transmission medium between the network users? Furthermore, can we design a resource allocation policy that delivers a performance guarantee for a control system using a wireless network?

Any solution we identify must satisfy other general constraints imposed by future applications and the underlying technology. A wireless communication protocol must be robust to fluctuations in the medium and easily adapt to it. It must not be fragile to a sudden breakdown of the wireless link. Sensor networks are characterized by high node densities, low data rates and energy constraints. In addition, a network of control systems is likely to have small and equal packet sizes. However, packet generation rates need not be low, relative to the data rates supported by the network. We consider large networks, which are typically organized without much hierarchy, due to similar energy constraints on all the nodes. Network planning and deployment is a time- and energy-consuming task, and impractical for such net-

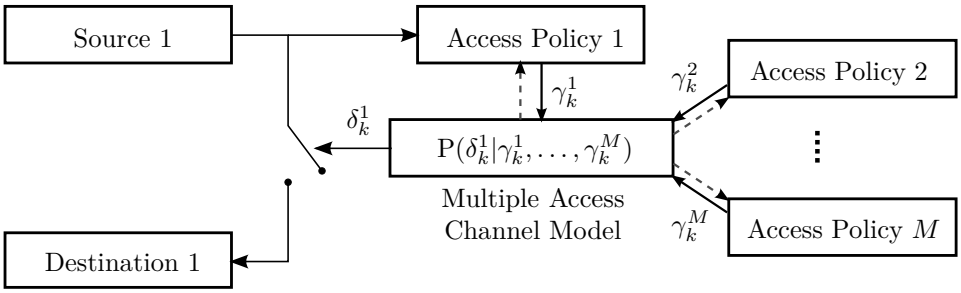


Figure 1.4: A packet-based model for an instantaneous, memoryless multiple access channel. The channel access policy associated with each source-destination pair using the shared channel, generates an access request γ_k^j, for $1 \leq j \leq M$. The source-destination pairs $j \in \{2, \ldots, M\}$ are represented by their channel access policies alone. The channel is turned 'on' or 'off' by the corresponding δ_k^j, as determined by all the access requests.

works. This is especially true for applications which involve a frequently changing network, possibly due to the presence of mobile agents or other moving objects.

1.2 Motivating Examples

We now present a model for the interactions in a shared link between multiple users. With the help of a number of examples, we illustrate the need for the channel access policy presented in our model and the unsuitability of existing channel access solutions. Following this, we present the state-based channel access policy explored in this thesis.

1.2.1 A Packet-based Multiple Access Channel Model

In Figure 1.4, we consider M source-destination pairs that communicate over a shared link. Each source-destination pair uses a channel access policy to request access to the channel. The figure depicts only the first source-destination pair. The other pairs using the shared medium are represented by their channel access policies alone. Each channel access policy generates an access request $\gamma_k^j \in \mathbb{Z}^+$, for $1 \leq j \leq M$, where k denotes the discrete time index. The discrete-valued γ_k^j conveys the need to transmit, and its range is determined by the transmission protocol and the wireless transceiver. The channel between each source-destination pair is represented by a switch, which is controlled by the associated channel access indicator $\delta_k^j \in \{0, 1\}$. This indicator δ_k^j is determined by all the access requests $\{\gamma_k^1, \ldots, \gamma_k^M\}$ in a manner described by the probability kernel in the figure. The probability kernel $\mathrm{P}(\delta_k|\gamma_k^1, \ldots, \gamma_k^M)$ models the interaction of the channel access requests in the shared

wireless medium as instantaneous, memoryless and randomized. This probability kernel represents our packet-based model for the multiple access channel between the source-destination pair depicted in Figure 1.4.

We use an instantaneous interaction model, but this need not result in any loss in generality. At first glance, such a model may not seem to capture the delay in accessing the network due to contention from other users. However, any delay induced by the multiple access network is simply attributed to the channel access policy, and not to the interaction in the channel. Thus, the delay due to multiple access for a successfully transmitted packet can be measured by the number of channel access requests[1] issued before δ_k^j attains a value of 1. If the channel access policy imposes a bound on the maximum number of attempts to transmit a given packet, and transmission fails for each of these attempts, then the packet is discarded. In such a case, the packet is considered as 'lost', and the delay is considered to be unbounded for this packet.

Our interaction model is memoryless as the probability kernel does not depend on past channel access requests or outcomes. However, the channel access policy may use past outcomes to determine future access requests. Thus, this modelling choice does not impose any limitations. Similarly, our choice of a randomized interaction model ensures no loss of generality. As an added benefit, it can be used to model physical losses from the medium, hidden terminals, unsynchronized clocks and other such phenomena unique to the wireless medium.

Let us consider a simple example to illustrate the need for a channel access policy. We allow two systems to use a broadcast network, such as a wireless link, without an explicit access mechanism. Through this example, we learn the limitations of the broadcast medium.

Example 1.1

We use a laboratory tank process, consisting of a coupled upper and lower tank, with wireless sensing and actuation links. The single process experimental setup is depicted by the illustration to the left in Figure 1.5. We wish to regulate the level of water in the upper tank. The links use a shared medium, but no channel access policy; they simply transmit when they generate a packet. We observe the effect of interference from both the links on a reference tracking experiment. Next, we conduct the same experiment with two similar processes, in place of one, to observe the effect of four interfering links. The double process experimental setup is depicted by the illustration to the right in Figure 1.5.

The results obtained from these real experiments are shown in Figure 1.6. To evaluate the above scenarios, we compute the average probability of a packet loss. We find that interference from two links causes 14% of the packets to be lost, whereas interference from four links causes as much as 45% of the packets to be lost. As expected, the performance of reference tracking with four interfering links is worse than with two.

[1]The source-destination pair is considered to operate on a much slower time scale than the channel access policy itself, thus allowing for multiple transmission requests per packet.

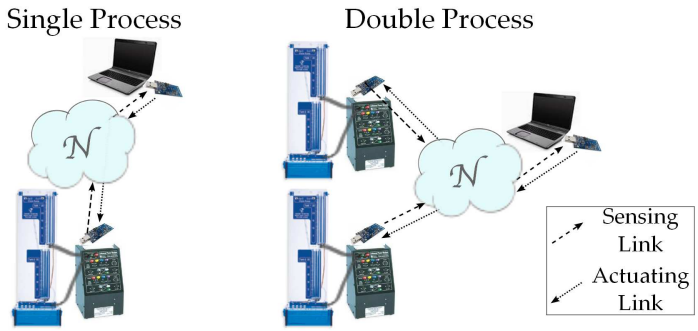


Figure 1.5: The setup to the left depicts a single coupled tank process which uses a wireless network to communicate with the controller. In the setup on the right side, two such processes share the network. The sensor nodes broadcast when they have data to transmit, without an explicit multiple access protocol in Example 1.1, or using random access in Example 1.3.

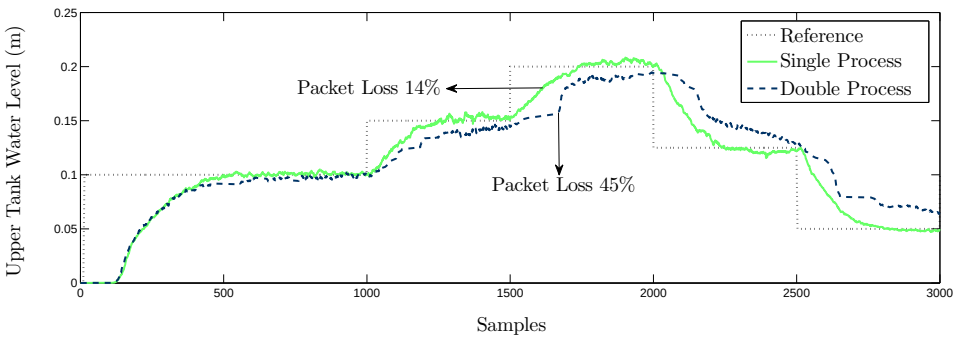


Figure 1.6: A comparison of two experiments: the solid line corresponds to the results of the reference tracking experiment with a single process using the broadcast medium, whereas the dashed line corresponds to the same experiment with two processes sharing the medium. In the second experiment, one of the processes suffers a packet loss of 45%, and consequently, its control performance is worse.

Interference from just four links can cause almost half the packets to be lost in a broadcast medium. Thus, a shared network requires a policy to arbitrate channel access among its users.

The channel access policy is a randomized strategy that generates an access request, and serves two main roles. Its first role is to prevent interference by arbitrating access to the shared network. It accomplishes this by typically spacing apart transmissions in time or frequency. However, any shared medium can only

support a given amount of network traffic, and preventing interference in high traffic networks is a futile task. Thus, the policy must also regulate network access by curtailing transmissions from users to prevent congestion. Through this second role, the channel access policy regulates the contribution of each node to the total traffic in the network.

The next two examples illustrate two commonly used channel access policies that accomplish these two roles in different ways. Through these examples, we examine the unsuitability of existing channel access solutions. In the following example, each user is allotted a slot in a transmission schedule. This constitutes a channel access policy called Time Division Multiple Access (TDMA) and the resulting access mechanism is typically non-randomized. However, we can still use the model in Figure 1.4 to describe such a policy. The transmission schedule can be static or dynamic; the order of transmission is fixed and does not change with time in a static schedule, whereas the order of transmission may be altered with time in a dynamic schedule. The following example illustrates the drawbacks of both approaches.

Example 1.2

We consider a network of wireless tank processes as in Figure 1.5, wherein each node is allotted a time slot to transmit data according to a given schedule. Each data packet in this network takes T_{tx} seconds to transmit, and each wireless tank process is sampled every T_{s} seconds. Then, the maximum number of processes that can be supported by this network is given by $M = T_{\text{s}}/T_{\text{tx}}$.

Now, consider a static schedule for a network of M processes. Once the water level in the tank has been regulated, the nodes do not have much to convey until the next reference change. However, each node is still allotted a slot every T_{s} seconds. This is inefficient, and reduces the number of users the network can support. A static schedule for more than M processes can only be achieved by increasing time between transmitted samples. Thus, some data samples must be discarded irrespective of how important they are to the control system.

Next, consider a dynamic schedule for a network of $2M$ processes. A static schedule cannot support $2M$ processes at the same sampling rate, and thus, a dynamic schedule must be used. Now, slots can be allotted to nodes that require the channel the most. However, the nodes that require access must first be identified. A new schedule must be drawn up and all $2M$ nodes must be notified of the change. In a wireless network, information collection and distribution is subject to the vagaries of the wireless medium, and some data may be lost. If request notifications are lost, the corresponding nodes do not get to access the network. If the schedule change notification is lost, two or more nodes may access the channel simultaneously, resulting in a collision.

Example 1.2 shows that static schedules are not scalable with the number of nodes in the network. Static allocations result in a waste of scarce resources, while dynamic allocations are difficult to implement over a lossy medium. A centralized scheduler,

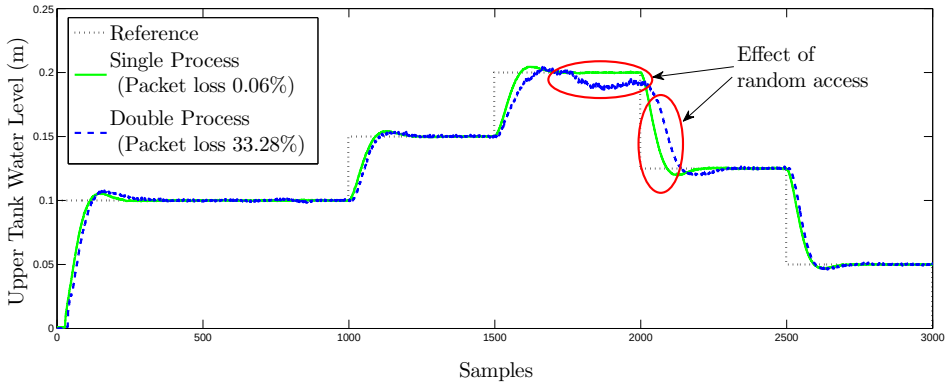


Figure 1.7: Effect of random access: When two processes share the medium and use CSMA/CA, one of the processes suffers a packet loss of 33%. This reflects on its closed-loop performance, and the reference tracking is much noisier. Also, it does not track the reference for a small region indicated above, and there is a delay compared to the single process.

which is susceptible to failure, or an energy intensive consensus algorithm must be used to draw up dynamic schedules.

In the third example, we use a policy that determines access to the network in a distributed, randomized way. This facilitates an easy deployment on wireless nodes. The distributed nature of the access policy implies that two or more nodes in the network may decide to transmit at the same time, resulting in a collision, wherein all packets are lost. In this example, the nodes use the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol, to illustrate the effect of collisions and *random access* on a control system.

Example 1.3

We return to our experimental setup from Example 1.1, with channel access determined by CSMA/CA. The results of a reference tracking experiment conducted on the single and double process setup are depicted in Figure 1.7. Firstly, by comparing with the results in Figure 1.6, note the remarkable improvement in performance over the experiments in Example 1.1, due to using a multiple access mechanism. However, random access results in random losses due to collisions. After a number of random transmission attempts that result in collisions, an unsuccessful packet is simply discarded irrespective of how important it is to the control system. The effect of such random losses is particularly visible when there are four interfering links. The reference tracking is noisier and delayed as well, compared to the performance obtained with two interfering links.

Despite the ease of implementation, the randomness of channel access makes this method unsuitable for critical control systems. Notice that both the access policies

presented above were *agnostic*, or blind, to the application layer. The agnostic approach ensures modularity in design, and is well-suited to serving heterogeneous networks. However, such an approach cannot deliver a satisfactory performance guarantee for control systems. In this thesis, we therefore seek an approach that guarantees stability for a network of control systems over a shared medium, while being easy to deploy, ad hoc and scalable.

1.2.2 State-based Channel Access

We propose a *state-based* channel access policy, in place of agnostic methods, with an emphasis on distributed solutions that are well-suited to wireless nodes. A state-based policy uses the state of the plant being controlled to influence the probability of a successful transmission. The state contains information about the plant, as well as the network, due to feedback control. Thus, by using the state to decide when and how frequently to transmit, a control system can adapt its contribution to the network traffic, and enable the network to adapt its output to the plant state. However, directly using the state of the plant to determine an access probability may result in a mechanism that is difficult to implement and analyze. Instead, we envision a state-based access policy as a combination of a *state-based directive*, and a *contention resolution mechanism* (CRM) that determines channel access in a distributed manner. Our approach is motivated by an understanding of the two roles played by a channel access policy: a CRM works by resolving contention between simultaneous channel access requests, thus spreading traffic that arrives in bursts. However, CRMs typically attempt to transmit a packet a few times, beyond which the packet is discarded. We appropriate this latter role of discarding packets to the *state-based directive*, as it can take an informed decision on the relevance of the packet to the control system and its impact on the network.

In the next two examples, we illustrate two different realizations of a state-based channel access policy. These examples establish that a state-based access policy can provide better performance guarantees than agnostic methods. The first example presents an event-based approach, where the state-based directive is a binary notification of an event, and the CRM is a randomized access policy. The second example is a priority-based approach, where the state-based directive is a discrete-valued priority associated with the current data packet. The control systems in the network exchange and evaluate priorities in a distributed manner using a *tournament*.

Example 1.4

We consider a heterogenous network of 20 first-order stochastic plants, indexed by $j \in \{1, \ldots, 20\}$. There are three different types of plants, representing different physical processes, sampled at different rates. These plants use a shared wireless link between their sensors and controllers, while the actuation links are dedicated connections. The channel access policy comprises of an event-triggering policy and a CRM. The event-triggering policy selects a data packet for transmission if the state satisfies the criterion $(x_k^j)^2 > \epsilon_j$ for each plant in the network. The CRM permits

Table 1.8: A comparison of control costs with and without an event-based policy

	Event-based Access	Random Access
Control Cost for plant type 1	23.5785	45.3074
Control Cost for plant type 2	8.3489	10.0028
Control Cost for plant type 3	5.3803	6.1213

Table 1.9: A comparison of control costs with and without state-based priorities

	Prioritized Access	Random Access
Average Control Cost	0.2576	0.3524

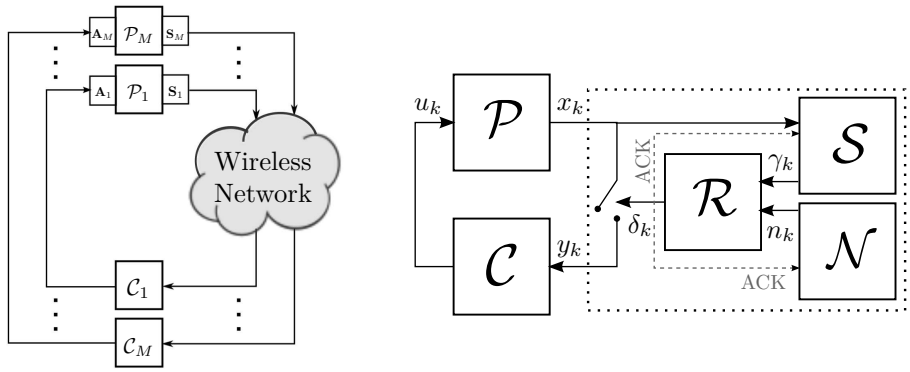
three attempts to transmit this packet, depending on the outcome of a Bernoulli process, with probabilities $\{1, 0.75, 0.5\}$, respectively. The controllers implement a state feedback law, using an estimate of the state when the state is not available. The control cost is a quadratic function of the state and the control signal, for a finite number of time instances.

In Table 1.8, we compare the control costs obtained with and without an event-triggering policy, i.e., with $\epsilon_j = 2.5$ and $\epsilon_j = 0$, respectively, for all j. There is a marked improvement with an event-triggering policy due to fewer collisions when using the CRM.

Example 1.5

We consider a homogenous network with 20 first-order linear stochastic plants. These plants use a shared wireless link between their sensors and controllers, while the actuation links are dedicated connections. The channel access policy comprises of a state-based prioritization policy and a mechanism to compare priorities in a distributed manner, called tournaments. Each plant generates a packet to transmit, and these packets vie for $N_{\mathrm{T}} = 10$ transmission slots. The prioritization policy uses a quadratic function of the innovations, which is the new information in the current measurement, and discretizes this value to a 8-bit priority field. The winners of the tournaments have the highest priority values. Thus, the probability of a successful transmission varies with the priority of the data packet. On an average, each plant in this network can be shown to transmit with probability 0.4403.

The controllers implement a state feedback law, using an estimate of the state when the state is not available. The control cost is a quadratic function of the state and the control signal, for a finite number of time instances. In Table 1.9, we compare the control costs obtained with the state-based prioritization policy against a random access policy with the same average probability of transmission.



(a) NCSs with wireless sensing link (b) State-based channel access on the sensing link

Figure 1.10: To the left, a network of control systems with a shared wireless sensing link and individual, dedicated actuation links. To the right, a control system in this network that uses a state-based policy to access the shared link. The block $\mathcal{N}$ is an aggregated representation of all the other network users.

Clearly, the prioritization policy results in lower costs than random access, because data packets with large associated costs are transmitted with a higher probability.

Thus, state-based channel access can significantly improve control performance, and potentially provide better guarantees, for wireless NCSs. In the next section, we formulate the problem of state-based channel access and identify some important issues we deal with in this thesis.

1.3 Problem Formulation

We consider a network of M control systems, consisting of a plant $\mathcal{P}_j$ and a controller $\mathcal{C}_j$ each, for $1 \leq j \leq M$. The control systems share access to a common wireless link between their sensors and controllers. We assume that the communication between the controllers and actuators occurs over a point-to-point link, not a shared network, as depicted in Figure 1.10(a).

From the perspective of a single control loop, this system can be modelled as shown in Figure 1.10(b). We drop the index j in this figure for simplicity. The block $\mathcal{N}$ represents the network as seen by this loop and the block $\mathcal{R}$ denotes the CRM, which determines whether the control loop or the rest of the network gets to access the shared medium. Each of the blocks in the model are explained below.

Plant: We consider a linear, discrete-time stochastic plant $\mathcal{P}$, with a state $x_k \in \mathbb{R}^n$. The state is described by the recursive update equation $x_{k+1} = Ax_k + Bu_k + w_k$, where $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are the system and input matrices, respectively.

The process noise w_k is an independent and identically distributed (i.i.d.) zero-mean Gaussian process with covariance matrix R_w. The initial state x_0 is a zero-mean Gaussian with covariance matrix R_0.

State-based Scheduler: There is a local scheduler $\mathcal{S}$, situated in the sensor node of each control system, which generates a discrete-valued channel access request γ_k. The state-based channel access policy, denoted by f, is applied to input arguments derived from the information pattern at the scheduler, $\mathbb{I}_k^s$. This information pattern consists of the system variables known to the scheduler, such as the state of the plant, previous access requests, channel outputs and applied controls. A successful transmission results in an acknowledgement (ACK) from the receiver, enabling the scheduler to keep track of previous channel outputs. The state-based channel access policy can be realized as an event-triggering policy, with $\gamma_k \in \{0, 1\}$ as an event indicator, or as a prioritization mechanism, with $\gamma_k \in \{0, \ldots, \gamma_{\max}\}$ as the priority values for some suitably chosen $\gamma_{\max} > 0$.

Network: The block $\mathcal{N}$ represents all the other traffic sources in the shared network. The network traffic indicator is denoted n_k, and its description depends on the realization of the channel access mechanism. The network can consist of heterogenous sources, and thus, n_k need not be generated by an information pattern similar to $\mathbb{I}_k^S$.

CRM: The CRM is a randomized strategy that resolves contention between multiple simultaneous channel access requests. When the contention is resolved in favour of our control loop, $\delta_k = 1$, and otherwise 0. Thus, the channel access indicator is given by $\delta_k = \mathcal{R}(\gamma_k, n_k)$. Notice that the policy $\mathcal{R}$ is a combination of the CRM and the interaction between various access requests, denoted by the probability kernel in Figure 1.4.

Controller: The channel output or the measurement across the network is denoted by y_k. When $\delta_k = 1$, the full state is available to the controller $\mathcal{C}$. When $\delta_k = 0$, the controller is only aware that it has not received a packet. The control law g generates the control signal u_k when applied to input arguments derived from the information pattern at the controller $\mathbb{I}_k^c = \left\{ \boldsymbol{y}_0^k, \boldsymbol{\delta}_0^k, \boldsymbol{u}_0^{k-1} \right\}$. The bold font denotes a set of variables such as $\boldsymbol{a}_t^T = \{a_t, a_{t+1}, \ldots, a_T\}$.

The controller tries to minimize the objective function, defined over a horizon N, and given by

$$J = \mathbb{E}\left[x_N^\top Q_0 x_N + \sum_{s=0}^{N-1} (x_s^\top Q_1 x_s + u_s^\top Q_2 u_s) \right] ,$$

where the weighting matrices Q_0 and Q_1 are non-negative, and Q_2 is positive definite.

Observer: Since the state is not always available to the controller, we sometimes need an observer ($\mathcal{O}$) to generate an estimate of the state. This estimate is given by $\hat{x}_{k|k} = \mathbb{E}[x_k|\mathbb{I}_k^c]$. The resulting estimation error is defined as $\tilde{x}_{k|k} \triangleq x_k - \hat{x}_{k|k}$, with error covariance $P_k = \mathbb{E}[\tilde{x}_{k|k}\tilde{x}_{k|k}^\top|\mathbb{I}_k^c]$.

In this thesis, we seek state-based channel access methods that are capable of delivering performance guarantees for wireless NCSs. To this end, we address the following questions:

Q1: Structural Analysis: Different multiple access architectures arise when we have different information patterns at the scheduler. What is the impact of the information pattern on the design and performance of a stochastic control system?

Q2: Realizations of State-based Access: In what ways can we use these information patterns to design channel access methods? How do we find realizations for the desired functionalities of the blocks $\mathcal{S}$ and $\mathcal{R}$?

Q3: Network Modelling: How do we model the interactions between different control systems in a multiple access channel? Can we use such a model to characterize the access probability of a state-based method?

Q4: Stability Analysis: Can we comment on the stability properties of a network of wireless NCSs, given the channel access policy and the corresponding model for network interactions?

Q5: Design: What choices of the scheduling policy f, the control policy g and the CRM $\mathcal{R}$ must be made to achieve a certain performance or guarantee a property for the control system and the network?

By designing a state-based channel access policy for a stochastic system, we are, in effect, choosing a multiple access channel for our control system. Our work presents some ways of doing this, while obtaining performance guarantees for each control system and the entire network itself.

1.4 Inherent Limitations of the Problem Formulation

By choosing the above problem formulation, we have inherently limited the scope of our research. We briefly motivate some of the choices and assumptions underlying the formulation presented above.

Stochastic plant: We consider a stochastic plant in our setup for two main reasons. Firstly, a wireless channel is modelled as a stochastic process, and any control system using a wireless network will consequently be a stochastic process, irrespective of the plant model. Secondly, analyzing network traffic requires probabilities, and a stochastic plant model will readily supply this. A benefit of the stochastic

model is that stochastic control offers a suitable framework to interpret the results we obtain.

Linear Quadratic Gaussian (LQG) Cost: The linear plant with Gaussian random variables and a quadratic cost offers us the simplest scenario to understand the implications of multiple access for control systems. This class of problems often leads to well-defined and well-understood solutions.

Dedicated Actuation Links: This assumption allows us to focus on the implications of multiple access on the sensing link. It is a likely scenario of operation in process control, as actuators require cables drawn from a power source and are thus wired to the controllers (Willig et al., 2005). This architecture also arises when the controller is implemented at the actuator. Even otherwise, a controller is likely to have more processing power and may even be centrally located along with the other controllers in the network. This would permit them to coordinate a schedule for transmissions on the actuation links, thus avoiding the need for distributed non-coordinated multiple access decisions at each controller.

Lossless transmissions: To focus on the implications of multiple access, we assume that the physical medium itself does not lose packets. Many of our results can be extended to include simple packet loss models.

Lossless ACKs: Most CRMs and network protocols provide ACKs when packets are received successfully. The ACK directly follows a transmission slot, in most protocols. Under fair conditions, a link may not see drastic changes within short intervals of time. Hence, ACKs are assumed to be delivered without losses.

Real-valued Transmissions: The wireless channels we consider cannot support real-valued transmissions. However, most network protocols provide sufficiently large payloads to allow us to neglect the effect of quantization. In addition, this assumption allows us to focus on the implications of multiple access.

1.5 Thesis Outline

The rest of the thesis is organized as follows. In Chapter 2, we provide a review of existing literature pertaining to stochastic control, multiple access methods and NCSs. We also cover recent developments in event-based control in some detail. In the multiple access section of this chapter, we provide a framework to answer the questions posed in this thesis, by introducing different classes of channel access, including state-based access methods.

The next three chapters outline an event-based realization of a state-based channel access method. These chapters contain the main results of this thesis. Chapter 3 contains a structural analysis of event-based systems, and answers **Q1**. It also presents a control system architecture with design solutions related to **Q2** and **Q5**.

Chapter 4 contains a model for the interactions, resulting from event-triggered policies and the CRM, between the control systems in a shared network. This model uses an assumption from Bianchi (2000) to decouple the network-induced correlation between the control systems, and the assumption and resulting model are validated through simulations for various network configurations. This chapter provides answers to **Q3**.

In Chapter 5, we provide sufficient conditions for Lyapunov mean square stability of a network of event-based systems. We also use this analysis to identify a design procedure for event-triggering policies that guarantee stability of the plant and network. This chapter provides answers to **Q4** and **Q5**.

Chapter 6 provides an example of a state-based prioritization mechanism, which is an alternate realization of a state-based channel access mechanism. We evaluate the access probability and establish that a state-based access policy can provide better performance guarantees than agnostic methods. This chapter answers parts of **Q2**, **Q3** and **Q5**.

In Chapter 7, we examine the general stochastic problem of jointly optimizing measurement and control policies, when the measurement policy is nonlinear. Many such examples can be found in networked control. We present a general formulation for all these problems, and identify stochastic properties of the resulting control system that permit the measurement and control properties to be separately designed, with no loss in optimality. This work explores **Q1** in a general context.

Chapter 8 provides conclusions and some notes for future work.

1.6 Contributions

This thesis is based on the following publications. The order of the authors indicates the degree of contribution, where the first author performed most of the work. Chapter 3 is based on the following publications:

> C. Ramesh, H. Sandberg and K. H. Johansson. Design of State-based Schedulers for a Network of Control Loops, *IEEE Transactions on Automatic Control*, 58(8):1962-1975, 2013.
>
> C. Ramesh, H. Sandberg, L. Bao and K. H. Johansson, On the Dual Effect in State-based Scheduling of Networked Control Systems, in *Proceedings of the American Control Conference*, pages 2216-2221, 2011.
>
> C. Ramesh, H. Sandberg and K. H. Johansson, LQG and Medium Access Control, in *Proceedings of the 1st IFAC Workshop on Estimation and Control of Networked Systems*, pages 328-333, 2009.

Chapter 4 is based on the following publications:

> C. Ramesh, H. Sandberg and K. H. Johansson, Performance Analysis of a Network of Event-based Systems, *Submitted for journal publication*, 2014.

C. Ramesh, H. Sandberg and K. H. Johansson, Steady State Performance Analysis of Multiple State-based Schedulers with CSMA, in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 4729-4734, 2011.

Chapter 5 is based on the following publications:

C. Ramesh, H. Sandberg and K. H. Johansson, Stability Analysis and Design of a Network of Event-based Systems, *Submitted for journal publication*, 2014.

C. Ramesh, H. Sandberg and K. H. Johansson, Stability Analysis of Multiple State-based Schedulers with CSMA, in *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 7205-7211, 2012.

Chapter 6 is based on the following publications:

C. Ramesh, D. Jenkins, J. Araujo, H. Sandberg and K. H. Johansson, State-based Priorities for Tournaments in Wireless Networked Control Systems, 2014. To be submitted.

C. Ramesh, H. Sandberg and K. H. Johansson, Multiple access with attention-based tournaments for monitoring over wireless networks, in *Proceedings of the European Control Conference*, pages 4302-4307, 2009.

Chapter 7 is based on the following publication:

M. Rabi, C. Ramesh and K. H. Johansson, Separated Design of Encoder and Controller for Networked Linear Quadratic Optimal Control, 2014. To be submitted.

The following publications do not form a part of this thesis, but contributed to the network perspective of this work.

P. Chen, C. Ramesh and K. H. Johansson, Reducing Packet Loss Bursts in a Wireless Mesh Network for Stochastic Bounds on Estimation Error, in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 3130-3135, 2011.

P. Chen, C. Ramesh and K. H. Johansson, Network Estimation and Packet Delivery Prediction for Control over Wireless Mesh Networks, in *Proceedings of the 18th IFAC World Congress*, pages 6573-6579, 2011. (With corresponding Tech. Rep. TRITA-EE:043)

Chapter 2

Background

In this chapter, we present a summary of the main tools and methods that are used frequently in the rest of the thesis, along with an overview of related work in networked control systems (NCSs). The work presented in this thesis draws on many existing concepts and methods in classical control theory, particularly stochastic control. In addition, channel access and resource allocation are also well established topics in computer networks, and many of the protocols and policies we refer to in this thesis have been in use since the early days of computer networks. We present a short review of these topics. Then, we model some basic types of multiple access protocols, and comment on their applicability to NCSs. There is also a growing body of research related to various aspects of state-based channel access. In recent years, event-based systems have garnered much interest as a means to reduce congestion in NCSs. Prioritized access has also seen some new developments. We present an overview of some of the related work in NCSs, with a particular emphasis on event-based systems. We begin with an introduction to stochastic control.

2.1 Stochastic Control

Control systems use the principle of feedback primarily to counter uncertainties that cannot be accounted for using a deterministic model of the physical process, such as disturbances and model errors. However, controller design often requires even disturbances to be modelled, and stochastic processes offer us a natural way to accomplish this (Åström, 1970). Dynamical systems subject to stochastic disturbances form the main focus of stochastic control theory. This theory is particularly apt to study the impact of information and its availability on control systems. In this section, we consider the general control problem of minimizing the expected value of a quadratic cost criterion in the state and control variables of a linear multiple input multiple output plant. The controller observes the plant state through a dynamical nonlinear measurement process. This simple setup is sufficient to present some of the distinguishing aspects of stochastic control.

Consider a linear plant $\mathcal{P}$, described by

$$x_{k+1} = Ax_k + Bu_k + w_k \ , \tag{2.1}$$

where $x_k \in \mathbb{R}^n$ denotes the state, $u_k \in \mathbb{R}^p$ denotes the control signal, and $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times p}$ denote the state and input matrices. The process noise $w_k \in \mathbb{R}^n$ is a zero-mean i.i.d. Gaussian process with covariance $R_w \in \mathbb{R}^{n\times n}$. The initial state is also a zero-mean and i.i.d. Gaussian with covariance $R_0 \in \mathbb{R}^{n\times n}$.

The measurement y_k available to the controller is a dynamic, nonlinear function of the information available to the measurement process. In classical control, the measurement process is typically a noisy, static function of the state as measured by a sensor, such as $y_k = Cx_k + w_k$. The measurement process in our setup refers to a combination of a classical sensor measurement, a strategy designed to compensate other effects such as those induced by a transmission channel, and the channel itself. A compensation strategy is, in general, a dynamic policy, and in the case of network compensation, it is often a nonlinear policy. An arbitrarily varying channel is also typically modelled as a dynamical process. Thus, we consider the information available at the sensor to be the set of all the states and the past measurements and controls, and is denoted as $\mathbb{I}_k^s = \{\boldsymbol{x}_0^k, \boldsymbol{y}_0^{k-1}, \boldsymbol{u}_0^{k-1}\}$. Since this is a generic formulation, we let the measurements belong to an undefined but measurable space Υ, which is determined by the nature of the compensation policy and channel. The measurements are generated by the causal policy f_k, defined as the map

$$f_k : \mathbb{R}^{n(k+1)} \times \Upsilon^k \times \mathbb{R}^{pk} \to \Upsilon \ , \tag{2.2}$$

taking $(\boldsymbol{x}_0^k, \boldsymbol{y}_0^{k-1}, \boldsymbol{u}_0^{k-1}) \mapsto y_k$. We define the measurement strategy as the set of measurement policies chosen over the control horizon N and denote it as $\boldsymbol{f} = \{f_0, \ldots, f_{N-1}\}$.

The controller is also assumed to have perfect recall of all received measurements, implying that it remembers all the received measurements and past actions. Thus, the information pattern at the controller is given by $\mathbb{I}_k^c = \{\boldsymbol{y}_0^k, \boldsymbol{u}_0^{k-1}\}$. The control signal u_k is generated by a causal policy g_k, defined as the map

$$g_k : \Upsilon^{k+1} \times \mathbb{R}^{p(k+1)} \to \mathbb{R}^p \ , \tag{2.3}$$

taking $(\boldsymbol{y}_0^k, \boldsymbol{u}_0^{k-1}) \mapsto u_k$. The control strategy is denoted as $\boldsymbol{g} = \{g_0, \ldots, g_{N-1}\}$.

For a given measurement strategy $\boldsymbol{f}$, the control strategy $\boldsymbol{g}$ may be chosen to minimize the classical finite horizon linear quadratic Gaussian (LQG) cost, given by

$$J_{\mathrm{LQG}} = \mathbb{E}\left[\sum_{t=0}^{N-1}(x_t^\top Q_1 x_t + u_t^\top Q_2 u_t) + x_N^\top Q_0 x_N\right] . \tag{2.4}$$

This results in the following controller-only optimization problem:

$$\text{Problem 1} \ : \min_{\boldsymbol{g}} J_{\mathrm{LQG}}$$

Sometimes, the measurement strategy must also be chosen along with the control strategy. In such cases, we denote the net cost as J_{LQC} as there is an additive communication or measurement penalty term in the cost, such that $J_{\mathrm{LQC}} = J_{\mathrm{LQG}} + \mathbb{E}\sum_{t=0}^{N-1} \lambda c(y_t))$, for some scalar $\lambda > 0$ and $c : \Upsilon \to \mathbb{R}_0^+$. Then, we have the following joint measurement and control optimization problem:

$$\text{Problem 2} \ : \min_{\boldsymbol{f},\, \boldsymbol{g}} J_{\mathrm{LQC}}$$

In the rest of this section, we use Problem 1 to illustrate that probing incentives lead to a dual role for the control signals. We use Problem 2 to illustrate the impact of information patterns on decision makers in a control loop. Then, we consider the simpler case of static measurement policies, and introduce the certainty equivalence principle for such systems.

2.1.1 Probing and Dual Effect in Problem 1

We use the formulation of Problem 1 to examine the consequences of a nonlinear measurement policy. When the information available at the controller is limited, the control signal sometimes has a dual role of influencing the information available to the controller in the future, in addition to its original role of regulating the state. The benefit obtained by influencing future information is termed as a *probing incentive* (Feldbaum, 1961).

In general, any measurable function induces a distribution on its output. More precisely, let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be measurable spaces where $\mathcal{A}$ and $\mathcal{B}$ denote the σ-algebras on the sets X and Y, respectively. Also, let $h : X \to Y$ be measurable, i.e., $h^{-1}(B) \in \mathcal{A}$ whenever $B \in \mathcal{B}$, where h^{-1} is the inverse image of B under h. A probability measure P_X on X induces a probability measure $\mathrm{P}_X \circ h^{-1}$ on Y through h. In the context of stochastic control, the measurement and control strategies induce a probability measure on the control signals and other system variables in the feedback loop. Let us look at optimal estimation and control to understand the consequences of this.

First, let the primitive random variables of our control system, comprising of the initial state x_0 and process noise w_k for $k \geq 0$, be defined on a suitable probability triple $[\Omega, \mathcal{F}, \mathrm{P}]$[1]. Then, the control strategy, the measurement strategy, and the plant dynamics[2], induce a distribution on the system variables. We make this dependence explicit by using the notation $z_k(\boldsymbol{\omega}; \mathcal{P}, \boldsymbol{g}, \boldsymbol{f})$ for any system variable z_k, where ω denotes a realization of Ω. We can also write the expectation of a function of a system variable, such as $h(z_k)$, as $\mathbb{E}^{\mathcal{P},\boldsymbol{g},\boldsymbol{f}}[h(z_k)]$ to emphasize the strategies and functions that induce this expectation. We use this notation sparingly, in the

[1]The probability tripe $[\Omega, \mathcal{F}, \mathrm{P}]$ comprises of the event-space Ω, the σ-algebra of subsets on Ω, $\mathcal{F}$, and the probability measure on $\mathcal{F}$, P.

[2]For a given plant $\mathcal{P}$ such as in (2.1), we denote the plant dynamics also by $\mathcal{P}$, with a slight abuse of notation.

interest of brevity, and we only indicate a dependence on the policies that are consequential to the discussion at hand.

The impact of a nonlinear measurement policy is made clear by contrasting with the case of linear measurements. So, let us consider the classical setup, comprising of a linear stochastic plant with linear measurements. The measurements are given by $\bar{y}_k = C_k x_k + v_k$, where v_k is a zero-mean Gaussian process with covariance R_v, and the measurement noise $\boldsymbol{v}_0^k$, the initial state x_0, and the process noise $\boldsymbol{w}_0^k$ are independent. Then, the estimate $\hat{x}_{k|k} \triangleq \mathbb{E}[x_k|\mathbb{I}_k^c]$ and the estimation error $\tilde{x}_{k|k} \triangleq x_k - \hat{x}_{k|k}$ obey

$$\begin{aligned}\hat{x}_{k|k} &= A\hat{x}_{k-1|k-1} + Bu_{k-1} + K_k e_k \,,\\ \tilde{x}_{k|k} &= (I - K_k C_k)(A\tilde{x}_{k-1|k-1} + w_{k-1}) - K_k v_k \,,\end{aligned}$$

where the innovations process is defined as $e_k \triangleq \bar{y}_k - C_k(A\hat{x}_{k-1|k-1} + Bu_{k-1})$ (Kailath et al., 2000; Åström, 1970). The filter gain K_k is given by

$$K_k = (AP_{k-1|k-1}A^\top + R_w)C_k^\top (C(AP_{k-1|k-1}A^\top + R_w)C^\top + R_v)^{-1} \,,$$

and the estimation error covariance $P_{k|k} = \mathbb{E}[\tilde{x}_{k|k}\tilde{x}_{k|k}^\top|\mathbb{I}_k^c]$ is given by

$$P_{k|k} = AP_{k-1|k-1}A^\top + R_w - K_k(C(AP_{k-1|k-1}A^\top + R_w)C^\top + R_v)K_k^\top \,.$$

Notice that the estimation error $\tilde{x}_{k|k}$ and the innovations process e_k do not depend on the applied controls $\boldsymbol{u}_0^{k-1}$ and the policies $\boldsymbol{g}_0^{k-1}$ due to the linearity of the terms involved.

When the measurement policy is not linear, $\tilde{x}_{k|k}$ and e_k may depend on the control signals and policies. In such cases, the estimation error covariance is also influenced by the control policies. The controller, in such a setup, has an incentive to alter the statistics of the estimation error at future time instants, and thus improve the information available to it. This was termed as a *probing incentive* by Feldbaum (1961). We now define the dual effect by defining when it is not present.

Definition 2.1. *The control signals in the system, comprising of a linear stochastic plant* (2.1) *and a measurement strategy consisting of nonlinear policies* (2.2)*, are said to have* no dual effect of second order *if and only if the filtering and prediction errors for any two admissible control strategies* (2.3) *such as* $\bar{\boldsymbol{g}}, \widetilde{\boldsymbol{g}}$*, are equal for any two times* $k \geq \ell$*, i.e.,*

$$\begin{aligned}&\mathbb{E}^{\mathcal{P},\bar{\boldsymbol{g}},\boldsymbol{f}}\left[\left(x_k(\boldsymbol{\omega};\mathcal{P},\bar{\boldsymbol{g}},\boldsymbol{f}) - \mathbb{E}^{\mathcal{P},\bar{\boldsymbol{g}},\boldsymbol{f}}\left[x_k(\boldsymbol{\omega};\mathcal{P},\bar{\boldsymbol{g}},\boldsymbol{f})|\boldsymbol{y}_0^\ell\right]\right)^2\middle|\boldsymbol{y}_0^\ell\right] =\\ &\qquad\mathbb{E}^{\mathcal{P},\widetilde{\boldsymbol{g}},\boldsymbol{f}}\left[\left(x_k(\boldsymbol{\omega};\mathcal{P},\widetilde{\boldsymbol{g}},\boldsymbol{f}) - \mathbb{E}^{\mathcal{P},\widetilde{\boldsymbol{g}},\boldsymbol{f}}\left[x_k(\boldsymbol{\omega};\mathcal{P},\widetilde{\boldsymbol{g}},\boldsymbol{f})|\boldsymbol{y}_0^\ell\right]\right)^2\middle|\boldsymbol{y}_0^\ell\right] \,. \qquad (2.5)\end{aligned}$$

Notice that the above definition requires the filtrations generated by the system variables in both control systems to be equal. This definition is stated in a more rigorous manner in Chapter 7. A similar notion of a neutral system has been implied by Nayyar et al. (2014). Also, if we set one of the control strategies, say $\widetilde{g}_k = 0$ for all $k \geq 0$ in the above definition, we obtain the definition given by Bar-Shalom and Tse (1974). Finally, the dual effect can be present in any moment of the estimation error, not just the second central moment as described above. However, the second-order dual effect has an important consequence for optimal control design with respect to a quadratic cost, which we describe in a later section. We now examine the optimal control policy for the LQG cost (2.4) to see how the probing incentive introduces a dual effect.

The optimal control policy in the classical setup is found using dynamic programming (Bellman, 1954). This method uses a backwards induction principle, and solves for the optimal control by minimizing a cost-to-go function at each time instant. In the absence of complete state information, the minimization of the cost-to-go is assumed to result in a solution of the form $\mathbb{E}[x_k^\top S_k x_k | \boldsymbol{y}_0^k, \boldsymbol{u}_0^{k-1}] + s_k$ for the cost-to-go, where S_k and s_k are positive semi-definite matrices of the appropriate dimensions. This assumption is usually proven using induction. We attempt to use the same method for our problem setup, but find that a similar solution is difficult to obtain in general.

At time N, the cost-to-go can always be assumed to have the form $V_N = \mathbb{E}[x_N^\top S_N x_N | \boldsymbol{y}_0^N, \boldsymbol{u}_0^{N-1}] + s_N$, where $S_N = Q_0$ and $s_N = 0$, with no loss in generality. At time $N-1$, we have

$$
\begin{aligned}
V_{N-1} = & \min_{u_{N-1}} \mathbb{E}[x_{N-1}^\top Q_1 x_{N-1} + u_{N-1}^\top Q_2 u_{N-1} + V_N | \boldsymbol{y}_0^{N-1}, \boldsymbol{u}_0^{N-2}] \\
= & \min_{u_{N-1}} \mathbb{E}\Big[x_{N-1}^\top (Q_1 + A^\top S_N A) x_{N-1} + u_{N-1}^\top (Q_2 + B^\top S_N B) u_{N-1} \\
& \underbrace{\qquad + x_{N-1}^\top A^\top S_N B u_{N-1} + u_{N-1}^\top B^\top S_N A x_{N-1} | \boldsymbol{y}_0^{N-1}, \boldsymbol{u}_0^{N-2} \Big]}_{\text{To be minimized}} \\
& + \underbrace{\mathbb{E}[w_{N-1}^\top S_N w_{N-1}]}_{\text{because } w_{N-1} \text{ is independent}} .
\end{aligned}
$$

The last equality is obtained by substituting for x_N using the state equation (2.1), and evaluating terms related to $\mathbb{E}[w_{N-1}]$ to zero. The cost-to-go can be written as

$$
\begin{aligned}
V_{N-1} &= \mathbb{E}[x_{N-1}^\top S_{N-1} x_{N-1} | \boldsymbol{y}_0^{N-1}, \boldsymbol{u}_0^{N-2}] + s_{N-1} \\
S_{N-1} &= Q_1 + A^\top S_N A - A^\top S_N B (Q_2 + B^\top S_N B)^{-1} B^\top S_N A \\
s_{N-1} &= \mathsf{tr}\{S_N R_w\} + \mathsf{tr}\{A^\top S_N B (Q_2 + B^\top S_N B)^{-1} B^\top S_N A P_{N-1|N-1}\} ,
\end{aligned}
$$

where the estimation error covariance is given by

$$
P_{N-1|N-1} = \mathbb{E}[\tilde{x}_{N-1|N-1}^\top \tilde{x}_{N-1|N-1} | \boldsymbol{y}_0^{N-1}, \boldsymbol{u}_0^{N-2}] .
$$

This is similar to the classical setup. However, things become more interesting at $N-2$. Now, we have

$$\begin{aligned} V_{N-2} &= \min_{u_{N-2}} \mathbb{E}[x_{N-2}^\top Q_1 x_{N-2} + u_{N-2}^\top Q_2 u_{N-2} + V_{N-1} | \boldsymbol{y}_0^{N-2}, \boldsymbol{u}_0^{N-3}] \\ &= \min_{u_{N-2}} \mathbb{E}\Big[x_{N-2}^\top (Q_1 + A^\top S_{N-1} A) x_{N-2} + u_{N-2}^\top (Q_2 + B^\top S_{N-1} B) u_{N-2} \\ &\qquad + x_{N-2}^\top A^\top S_{N-1} B u_{N-2} + u_{N-2}^\top B^\top S_{N-1} A x_{N-2} \\ &\qquad \underbrace{+ \mathsf{tr}\{A^\top S_N B (Q_2 + B^\top S_N B)^{-1} B^\top S_N A P_{N-1|N-1}\} | \boldsymbol{y}_0^{N-2}, \boldsymbol{u}_0^{N-3} \Big]}_{\text{To be minimized}} \\ &\qquad + \mathsf{tr}\{S_N R_w\} + \mathsf{tr}\{S_{N-1} R_w\} + 0 \,. \end{aligned} \tag{2.6}$$

Notice that there is one more term to be minimized with respect to u_{N-2}. This is because the estimation error covariance $P_{N-1|N-1}$ may depend on the past applied controls, including u_{N-2}, when a nonlinear measurement policy is used. This minimization problem is in general non-convex due to the additional term, and a closed-form solution to the cost-to-go of the form $\mathbb{E}[x_{N-2}^\top S_{N-2} x_{N-2} | \boldsymbol{y}_0^{N-2}, \boldsymbol{u}_0^{N-3}] + s_{N-2}$ may not be possible to find. This also illustrates the dual roles of the control; the first role is to minimize the cost-to-go by *regulating* the state, and the second role is to *improve* the estimation error at future time instants.

We return to this proof shortly to examine how controller design becomes simplified when there is no dual effect.

2.1.2 Information Patterns in Problem 2

Dynamic programming was put forward by Bellman (1954) to study optimal sequential decisions, particularly in stochastic control. As the tool gained popularity, there were many who were interested in understanding dynamic programming from an information perspective. The optimal solutions obtained using dynamic programming were a consequence of the information available to the decision makers. In Problem 1 for example, if the controller 'forgets' past measurements, or control actions, will the optimal policy be different? These were the kind of questions examined by Chernoff (1963). Striebel (1965) and Witsenhausen (1971) took this further by examining 'classical' information patterns that resulted in the simple and elegant solutions found in Markov decision processes using dynamic programming. Non-classical information patterns, in general, introduce non-linearities and simple solutions found by identifying recursive sufficient statistics may not exist for these problems, as pointed out by Witsenhausen (1971). Partially nested information patterns, identified by Ho and Chu (1972), constitute the last special class we consider. This information pattern restores much of the simplicity of results obtained with classical information patterns for certain non-classical information patterns.

We now use the formulation of Problem 2 to examine the consequences of having multiple decision makers and information patterns in the feedback loop. Notice that in Problem 2, we are required to choose both the control and measurement policies at each time instant, and the information available to make these choices is not the same[3]. We now examine three information patterns and understand their consequences for control design.

- Classical Information Pattern (Witsenhausen, 1971): An information pattern is said to be classical if all decision makers receive the same information pattern independent of the time index k and have perfect recall. This implies that, for any design, the σ-fields determined in probability space by data available at successive stages are nested.
- Non-classical Information Pattern (Witsenhausen, 1971): An information pattern is non-classical if it is not strictly classical. Examples include delayed sharing patterns with delays greater than one and forgetful decision makers, among many others.
- Partially Nested Information Pattern (Ho, 1980): An information pattern is said to be partially nested, if each decision maker knows the information available to all the decision makers whose decisions affect the measurements available to it.

The last two definitions are better understood by considering the following static team decision problem.

Example 2.1
Let us consider a static team decision problem with six decision makers. The information available to the i^{th} agent is denoted z_i, and the action associated with it is denoted u_i, for $1 \leq i \leq 6$. Let ξ represent the random process, say a scalar for simplicity, that drives this system. The information structure can be inferred from the following output equations:

$$
\begin{aligned}
z_1 &= H_1 \xi \\
z_2 &= H_2 \xi \\
z_3 &= \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 0 \\ D_{31} \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ D_{32} \end{bmatrix} u_2 \\
z_4 &= \begin{bmatrix} H_2 \\ H_4 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ D_{42} \end{bmatrix} u_2
\end{aligned}
$$

[3]Problem 1 can also be seen to have multiple decision makers by treating each control policy in the finite horizon strategy as a decision maker. In fact, each control policy is chosen with different information sets.

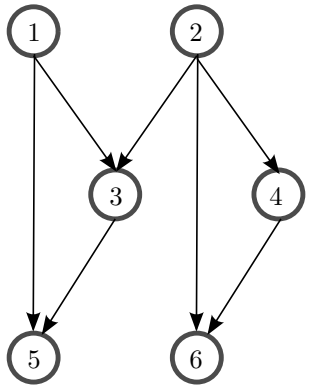


Figure 2.1: An information structure diagram for the static team decision problem in Example 2.1 with a partially nested information structure. This diagram depicts the dependence relationship between different agents. One agent's action may affect another agent's information, as indicated by the arrow. The same diagram can also represent a non-classical information pattern if some measurements are not available to successive agents.

$$z_5 = \begin{bmatrix} H_1 \\ H_2 \\ H_3 \\ H_5 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 0 \\ D_{31} \\ D_{51} \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ D_{32} \\ 0 \end{bmatrix} u_2 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ D_{53} \end{bmatrix} u_3$$

$$z_6 = \begin{bmatrix} H_2 \\ H_4 \\ H_6 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ D_{42} \\ D_{62} \end{bmatrix} u_2 + \begin{bmatrix} 0 \\ 0 \\ D_{64} \end{bmatrix} u_4 \ .$$

The information structure in Figure 2.1 captures some of the above interactions. However, the equations tell us more. For example, the above equations indicate that agent 3 has access to z_3 to generate its action u_3. Also, the decisions of agents 1 and 2 affect agent 3. However, through z_3, agent 3 also has access to z_1 and z_2, which is the information used by agents 1 and 2 in generating u_1 and u_2, respectively. Thus, it can anticipate u_1 and u_2 in a cooperative setup. A similar argument can be made for agents 4, 5 and 6. Thus, the above information pattern is nested.

Now consider what happens if $z_5 = H_5\xi + D_{51}u_1 + D_{53}u_3$. The information pattern is now non-classical, as the partial nestedness property no longer holds for agent 5. Witsenhausen (1968) illustrated, in his celebrated counterexample, that the optimal decision policies for such problems may be nonlinear even when the system is linear, and the measurements are a linear function of the state. He surmised that this was due to a signalling incentive; certain quadratic costs provide agents 1, 2 and 3 an incentive to signal the values of their measurements to agent 5. Furthermore,

if the last term $D_{53}u_3$ were to be replaced by some nonlinear function $f_{53}(u_3)$, this could provide agents 1 and 2 a probing incentive to improve the information available to agent 5 about u_3. This could help agent 5 interpret u_3 and learn the state ξ better. It is the lack of complete information at agent 5 that induces both the signalling and probing incentives. We can now understand the advantage of a partially nested information pattern. With full availability of necessary information, there are no signalling or probing incentives, and thus, the control policies are often as simple as those obtained with a classical information pattern.

Note that any control problem can be expressed as a static team decision problem by aggregating all the primitive random variables into one random vector, such as $\xi = [x_0 \ w_0 \ \dots \ w_{N-1} \ v_0 \ \dots \ v_{N-1}]^\top$. Thus, the above discussion is just as relevant to dynamical systems, as long as one treats the controller, or any other decision maker in the feedback loop, at each stage as a unique decision maker.

Now, returning to Problem 2, we see that the measurement policy f_k influences the information at the controller $\mathbb{I}_k^c$ through the measurement y_k, which is the action associated with the measurement policy. However, the controller does not have access to all the information available to the measurement policy, i.e., $\{\mathbb{I}_k^s \cup y_k\} \supset \mathbb{I}_k^c$. Thus, the resulting information pattern is non-classical. There is a signalling incentive on the link between the measurement policy and the controller, which is really the purpose of the measurement policy. The lack of complete information at the controller induces a probing incentive in the controller at previous stages. This probing incentive makes Problem 2 difficult to solve, in general.

2.1.3 Certainty Equivalence and Separation

We have now seen a number of problems where the full state is not available at the controller. In these cases, one can always compute the deterministic optimal controller, defined as the controller that minimizes the performance cost when it knows the exact instantiations of all random variables involved in the cost. Let the deterministic optimal control signal be denoted by $\bar{u}_k$. Then, a reasonable approach might be to use $\mathbb{E}[\bar{u}_k|\mathbb{I}_k^c]$ as the control signal, as explained by van de Water and Willems (1981). The certainty equivalence principle is said to hold when there is no loss of optimality in doing so. Certainty equivalence was proven to hold for the classical setup with linear, noisy measurements. A commonly accepted notion of certainty equivalence is given by Bar-Shalom and Tse (1974), which we present below.

Definition 2.2 (Certainty Equivalent Controller)**.** *A certainty equivalent controller uses the deterministic optimal controller, with the state x_k replaced by the estimate $\hat{x}_{k|k} = \mathbb{E}[x_k|\mathbb{I}_k^c]$.*

Definition 2.3 (Certainty Equivalence Principle)**.** *The certainty equivalence principle holds if the closed-loop optimal controller has the same form as the certainty equivalent controller.*

We also state a related property of stochastic control systems, the separation property. In a design problem with many decision makers, sometimes the optimization problem permits a separation in design of each decision policy, with no loss in optimality. In the context of optimal LQG with partial state information, this implies a separation in design of the observer and controller. Here, the form of the optimal controller can be different from that of the deterministic optimal controller. However, the control signal is still derived using the estimate of the state alone.

Definition 2.4 (Separation Principle)**.** *The separation principle holds if the closed-loop optimal control depends on the data only through the estimate* $\hat{x}_{k|k}$*.*

Separation was initially suggested by Joseph and Tou (1961) for the classical LQG setup, in the sense described above. In Chapter 7, we use the more general notion of separation, implying that each decision maker in the feedback loop can be designed independently, with no loss of optimality, when the separation principle holds.

Bar-Shalom and Tse (1974) show that if a static nonlinear measurement policy, such as $h_k(x_k)$, results in no dual effect of the second order, then the certainty equivalence principle holds for the given control system. We restate their result in the following theorem.

Theorem 2.1 (Bar-Shalom and Tse (1974))**.** *The optimal stochastic control for the system with linear dynamics (2.1), a static nonlinear measurement policy such as* $y_k = h_k(x_k, v_k)$*, for some independent Gaussian process* v_k*, and cost (2.4), has the certainty equivalence property if the control has no dual effect of second order. The minimizing control policy is then given by* $u_k = -L_k\hat{x}_{k|k}$*, and the resulting control cost is given by*

$$J_0 = \hat{x}_0^\top S_0 \hat{x}_0 + \mathit{tr}\{S_0 P_0\} + \sum_{n=0}^{N-1} \mathit{tr}\{S_{n+1}R_w + (L_n^\top (Q_2 + B^\top S_{n+1} B)L_n)P_{n|n}\} \,, \tag{2.7}$$

where the optimal control gain L_k *is given by*

$$L_k = (Q_2 + B^\top S_{k+1} B)^{-1} B^\top S_{k+1} A \,, \tag{2.8}$$

and the solution to the discrete-time Riccati equation S_k *is given by*

$$\begin{aligned} S_k &= Q_1 + A^\top S_{k+1} A - A^\top S_{k+1} B (Q_2 + B^\top S_{k+1} B)^{-1} B^\top S_{k+1} A \,, \\ s_k &= \mathit{tr}\{S_{k+1} R_w\} + \mathbb{E}[s_{k+1} | \boldsymbol{z}_0^k, \boldsymbol{u}_0^{k-1}] \\ &\quad + \mathit{tr}\{A^\top S_{k+1} B (Q_2 + B^\top S_{k+1} B)^{-1} B^\top S_{k+1} A P_{k|k}\} \,, \end{aligned} \tag{2.9}$$

Proof. We return to the dynamic programming computation in Section 2.1.1. We now show that a recursive solution to the optimization problem at any time step can be found when the control signal has no dual effect of order 2. We also derive

the form of the optimal controller and the resulting control cost, when there is no dual effect of the controls.

At time $N-2$, we noted earlier, in (2.6), that the presence of an additional term containing the estimation error covariance $P_{N-1|N-1}$ does not permit us to find a linear control policy that is optimal for the given cost-to-go V_{N-2}. However, if there is no dual effect of the second order, the estimation error covariance will not be induced by the past control policies. Consequently, this additional term drops out of the minimization problem. Then, the cost-to-go can be expressed in the desired form as $V_{N-2} = \mathbb{E}[x_{N-2}^\top S_{N-2} x_{N-2} | \boldsymbol{y}_0^{N-2}, \boldsymbol{u}_0^{N-3}] + s_{N-2}$. With similar arguments, this can be shown to be true for any $k \geq 0$, using induction. The solution to the minimization problem at any time k is then given by

$$u_k = -L_k \hat{x}_{k|k} \ , \tag{2.10}$$

where the optimal control gain L_k is given in (2.8). Furthermore, the matrix S_k in (2.9) is positive semi-definite and not a function of the applied controls $\boldsymbol{u}_0^{k-1}$. The scalar s_k is not a function of the applied controls $\boldsymbol{u}_0^{k-1}$ if and only if $P_{k|k}$ has no dual effect (Bar-Shalom and Tse, 1974). Since the optimal control signal (2.10) is a function of only the estimate $\hat{x}_{k|k}$, the certainty equivalence principle holds.

Using the above equations, we can find an expression for the control cost as $J_0 = \mathbb{E}[V_0] = \mathbb{E}[\mathbb{E}[x_0^\top S_0 x_0 | z_0] + s_0]$. This can be rewritten as

$$\begin{aligned} J_0 = \hat{x}_0^\top S_0 \hat{x}_0 + \mathsf{tr}\{S_0 P_0\} + \mathbb{E}[\sum_{s=0}^{N-1} \mathsf{tr}\{S_{s+1} R_w\}] \\ + \mathbb{E}[\sum_{s=0}^{N-1} \mathsf{tr}\{A^\top S_{s+1} B (Q_2 + B^\top S_{s+1} B)^{-1} B^\top S_{S+1} A P_{S|s}\}] \ , \end{aligned}$$

where the above equation was obtained by substituting for s_0. Using (2.9) and (2.10) in the above equation, we obtain the expression given in (2.7). □

2.2 Channel Access

In this section, we introduce various multiple access techniques drawn from computer networks, and present some protocols used in control networks over the last few years. When point-to-point channels are not available, broadcast channels must be used, with techniques to minimize interference from other users who share the same channel. Access schemes for such broadcast channels are known as *Multiple Access Protocols* (Rom and Sidi, 1990). These protocols are implemented in the Medium Access Control (MAC) layer, which forms a sub-layer to the second layer of the OSI network model (Tanenbaum, 2002), see Figure 2.2.

Numerous multiple access protocols have been suggested for different applications until now. These are broadly classifiable as centralized or distributed protocols, based on the hierarchies built into the protocols. Decentralized protocols, where all

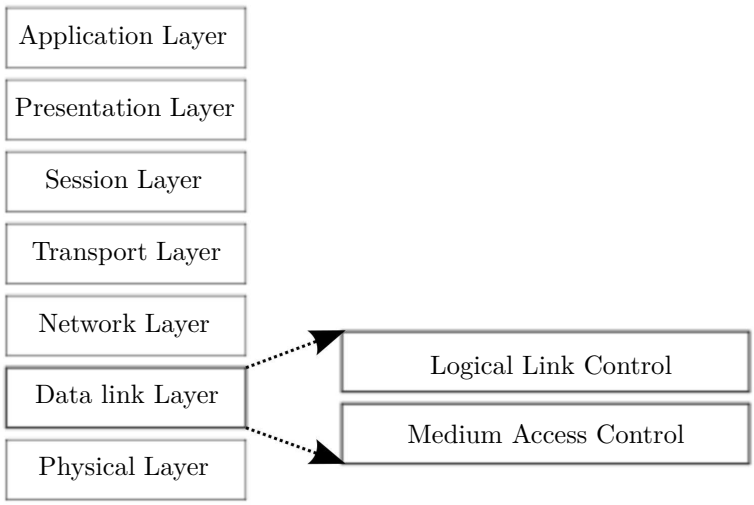


Figure 2.2: The multiple access protocol is implemented in the MAC layer of the network stack.

nodes implement the same set of rules, with no centralized coordinator which has its own set of rules, are the ones that we focus on in this thesis. Protocols are also classified as contention-based or conflict-free protocols with static or dynamic allocations. We described these terms in the following subsections.

2.2.1 Contention-free MACs

Contention-free methods assign a schedule for the users, which determines the order of network access, as illustrated in Figure 2.3. The schedule can only accommodate a fixed number of nodes commensurate to the capacity of the network. The biggest advantage of contention-free protocols is that they are conflict-free; they ensure that a transmission is always successful, when the physical medium does not cause any losses. This is achieved by allocating the channel to the users in a static or dynamic manner. The channel resources can be divided among the users in time, frequency or using codes.

We use Time Division Multiple Access (TDMA) as a typical example of conflict-free protocols in this thesis. In a network that uses TDMA, time is divided into slots, which are allotted to different users as per the schedule. When the schedule is static, this protocol typically results in a fixed delay, which depends on the size of the network. If the resources required by each user are fixed, then such a protocol is not scalable (refer to Example 1.2), despite its desirous property of guaranteeing transmissions. Furthermore, a static schedule is never altered; a transmission interval remains allotted to a network user, even if the user has no current need to transmit. This is inefficient, and reduces the number of users the

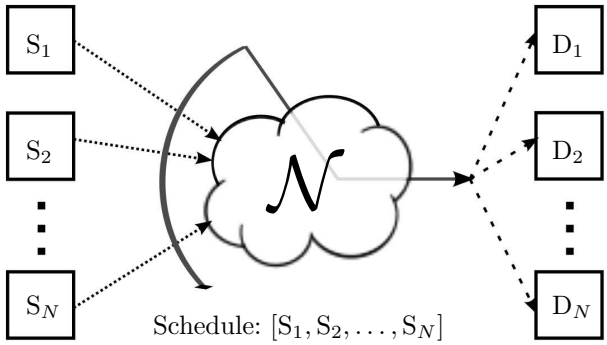


Figure 2.3: The network resources are partitioned and allocated using a schedule, which determines the order of access, in a contention-free MAC. The allocations can be static, when the schedule does not vary with time, or dynamic, when it does.

network can support. Thus, static allocations require over-provisioning of resources.

Dynamic resource allocations solve this problem. A new schedule is issued every few frames, based on the requirements of the control system. However, the requirements of all the control systems in the network must be collected to design a schedule, which must be distributed to all the systems as well. In a wireless network, information collection and distribution is subject to the vagaries of the wireless medium, and some data may be lost. Furthermore, these operations constitute a significant overhead in an energy and resource-constrained network. It is not easy to design cooperative, coordinated, dynamic protocols that are sufficiently robust to the wireless medium (Akyildiz et al., 1999; Gummalla and Limb, 2000; Ramaswami and Parhi, 1989; Goldsmith and Wicker, 2002).

2.2.2 Contention-based MACs

In a contention-based protocol, a transmitting user is not always successful, as illustrated in Figure 2.4. These protocols typically use a contention resolution mechanism (CRM). The CRM provides a distributed, randomized way to determine access to the network, in such a manner so as to eventually be successful in transmitting all the messages. In finite time, however, the CRM is not always successful in transmitting all messages. This is because packets transmitted using a contention-based protocol sometimes result in a collision. None of the packets involved are transmitted successfully as they interfere with each other. Collisions use up channel resources, and along with *random access*, cause significant deterioration in the performance of a control system. The CRM can be static or dynamic, and we look at this classification in more detail at the end of this section. Despite the obvious disadvantage of a possible collision, or the lack of a transmission guarantee, these protocols are popular in practice, as they are easy to deploy in an ad hoc manner. We look at some examples of these protocols below.

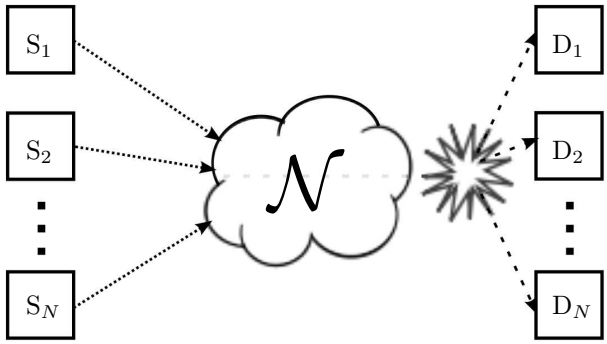


Figure 2.4: The nodes attempt to access the network resources, and any resulting contention is resolved using the protocol. The contention resolution can be successful, resulting in a transmission, or unsuccessful, resulting in a collision.

Aloha: This was the first random access protocol family introduced in networking literature (Abramson, 1970). *Pure Aloha* uses a very simple idea: a node attempts a transmission as soon as it generates a data packet. If the transmission is unsuccessful due to a collision with other packets, the data packet is scheduled for retransmission at a random time in the future, independent of other users. This protocol fares poorly in a broadcast medium, for the same reasons illustrated in Example 1.1. A broadcast medium is interference constrained, and simultaneous transmissions cannot be supported by such media. A modified version of this protocol, called *Slotted Aloha* results in a slightly higher throughput.

Aloha protocols exhibit poor performance due to the 'impolite' behaviour of users, who do not wait for an idle channel before commencing transmission. In contrast, Carrier Sensing Multiple Access (CSMA) protocols sense the channel, and only transmit when the channel is idle. If the channel is busy, they resort to different mechanisms for retransmission. Some of these protocols are described below.

p-persistent CSMA: In this protocol, a user who finds the channel busy waits and transmits as soon as the channel becomes idle with a persistence probability p. With probability $1-p$, the user delays transmission by τ seconds. The name of the protocol derives from the persistence of the node in transmitting once it senses a busy channel. In *Non-persistent CSMA*, for example, a user who finds the channel busy schedules a transmission to a random time in the future. In contrast, in 1-*persistent CSMA*, a user who finds the channel busy waits persistently and transmits as soon as the channel becomes idle. A tradeoff between these two strategies gave rise to p-persistent CSMA. These protocols were first proposed and analyzed by Kleinrock et al. (1983). We use this protocol in the thesis as an abstraction of the more complicated variant described below.

CSMA/CA: CA stands for collision avoidance. In this protocol, a user who

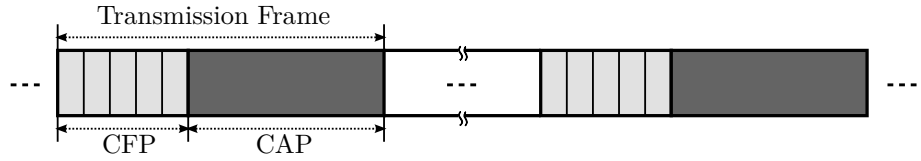


Figure 2.5: In a Hybrid MAC, the transmission time is divided into frames, and each frame consists of a contention-free period and a contention-access period.

finds the channel busy uses a more sophisticated retransmission mechanism. A popular scheme is the exponential backoff mechanism, which is described below. At each packet transmission instant, a backoff time is uniformly chosen in the range $(0, w-1)$, where w is the maximum value of the backoff time, which is called the contention window. With exponential backoff, the contention window is doubled with the number of failed transmissions. This allows the user to increase its MAC delay and waiting time proportional to the perceived traffic load of the network. Thus, the transmission delay here is very small for low traffic, and scales upwards as the traffic increases.

2.2.3 Hybrid MAC

The problem with contention-based protocols is the reduction in throughput with high traffic. So, we now examine some protocols that combine the advantages of both contention-free and contention-based methods with a hybrid MAC. In these protocols, the transmission time is divided into frames, which are further divided into a contention-free period and a contention-access period, as shown in Figure 2.5. The slots in the contention-free period are reserved for nodes which request for them using the contention-access period. Nodes may also request a contention-free slot for multiple frames. A node is also permitted to use only the contention-access period, if desired.

2.2.4 Protocols

We briefly outline the MAC layer of a few protocols relevant to control systems.

1. **CAN Bus:** This protocol is used by the smart distributed system, DeviceNet and CAN Kingdom. CAN is a serial communication protocol, which offers good performance for time-critical industrial applications (Robert Bosch GmbH, 1991; Johansson et al., 2005). Messages are allotted different static priorities, which are used to arbitrate access to the common bus. The arbitration is implemented using a bit-dominance strategy, which is described below. A node with a packet to transmit, attempts to secure the transmission slot

by winning a tournament slot. In the tournament slot, nodes transmit their priority bits, starting with the most significant bit. Other nodes listen to the network during their recessive bits, and drop out of contention when they hear a dominant bit, as it indicates that a node of higher priority desires access to the channel. The last node remaining secures the transmission slot. This protocol guarantees that the allotted priorities are observed during contention resolution, but it is hard to implement this protocol in a wireless network. An adaptation for wireless networks is discussed in Pereira et al. (2007), and a modification to the priority allocation is presented in Chapter 6.

2. **Token Bus:** This protocol is used by process field bus (PROFIBUS), manufacturing automation protocol (MAP), ControlNet and fiber distributed data interface (FDDI). The nodes in a token bus network are arranged logically into a ring, and each node knows the network address of its predecessor and successor in the ring. The node with the token is permitted to transmit until the end of the data packet, or until it runs out of time, whichever occurs earlier. Then, the token is forwarded to the next node in the logical ring. This mechanism guarantees a maximum waiting time before transmission, and makes the network deterministic. However, the token is susceptible to the vagaries of the wireless medium, such as packet losses and hidden terminals, and is hence not popular in wireless networks.

3. **Distributed Coordination Function:** This protocol is implemented in the IEEE 802.11 standard and uses CSMA/CA with exponential backoff. However, it results in random access, which could significantly deteriorate the performance of a closed-loop system (Liu and Goldsmith, 2004). It is also hard to analyze, as shown by Bianchi (2000).

4. **Beacon-enabled Hybrid MAC:** This name is used to denote the MAC layer specified by the IEEE 802.15.4 standard (IEEE, 2006) for wireless sensor networks. It uses Slotted CSMA/CA in the contention-access period, and TDMA in the contention-free period. The PAN coordinator is responsible for allocating available slots in the contention-free period, also known as GTS slots.

5. **WirelessHART Hybrid MAC:** This is another example of a hybrid MAC, as specified by the WirelessHART protocol (HART Communication Foundation, 2007). Here, the transmission time is divided into slots, and the MAC uses TDMA. However, each slot can be allotted to more than one node, and contention is permitted within the slot using CSMA/CA.

Time scales of the MAC and the control system: A natural unit of time for the description of the CRM is the time slot. In some protocols such as TDMA or p-persistent CSMA, the slot length can be equal to the time taken to transmit a

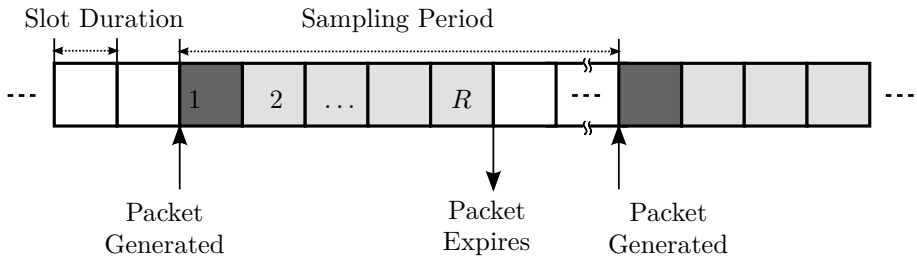


Figure 2.6: A packet is generated at the start of the time slot. The transmission time for a packet is equal to the slot time. A data packet is either delivered or discarded before the expiry instant. The sampling period is much greater than the slot duration, and the expiry instant occurs before the next sampling instant. Thus, there is no overlap in system and MAC time scales.

packet. In other protocols, such as CSMA/CA, the slot length is much smaller than the packet transmission time. In most of this thesis, we use a p-persistent MAC as an abstraction of the CRM, and accordingly assume that the control network time slot is equal to the transmission time of a single data packet. At the application layer, a more natural unit of time for the description of discrete time control theory is the sampling time of the physical process. In the rest of this thesis, we assume that the sampling time is much larger than the slot time, as shown in Figure 2.6. Furthermore, the entire operation of the channel access protocol, including all retransmission attempts, must be completed before the expiry instant. Thus, the data packet is delivered or lost by the expiry instant, which occurs prior to the next sampling instant. We do not consider an overlap in the time scales of the channel access protocol and the physical process.

2.2.5 Congestion Control

The multiple access methods presented above work by spreading simultaneous channel access attempts in different ways. However, as the network traffic increases, these methods are no longer effective, leading to network congestion. Users see a drastic reduction in quality of service, which may occur in the form of longer delays in accessing the network, or more collisions, depending on the access method used.

A congestion control or avoidance algorithm dynamically adapts the traffic contribution of each node to the total network traffic, and can be used to obtain a guaranteed quality of service. These algorithms typically use an indirect measure of the traffic in the network, such as the delay or the probability of collision, as an indicator of congestion. A class of optimal algorithms interpret this measure as the network price for using the resource, and devise an update rule for the traffic rate from the node based on the network price. For example, an increase in delay would require an appropriate reduction in data rates from the node, whereas a reduction

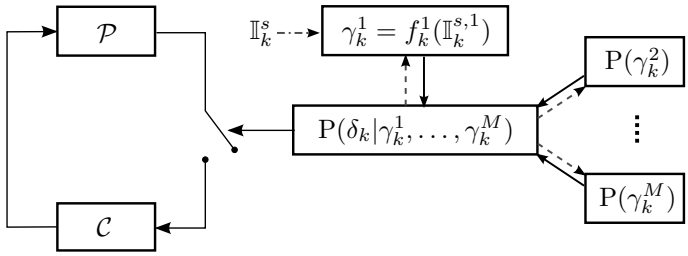


Figure 2.7: A model for a multiple access channel on the sensing link of a control system. The channel access policy f_k^j associated with each plant and controller using the shared channel, generates an access request γ_k^j, based on an information pattern $\mathbb{I}_k^{s,j}$, for $1 \leq j \leq M$. The source-destination pairs $j \in \{2, \ldots, M\}$ are represented by their channel access policies alone.

in delay could allow for an increase in the rate.

2.2.6 Channel Access Architectures

We have presented quite a number of channel access mechanisms in this section. Now, let us return to the model we presented in Figure 1.4, and see if it can be used to represent some of these mechanisms. For this purpose, we use the generalized model in Figure 2.7, as applied to a system with a linear stochastic plant $\mathcal{P}$ as source and a controller $\mathcal{C}$ as destination. Note that the channel requests γ_k are now generated using the information pattern $\mathbb{I}_k^s$ defined in Section 2.1. The information pattern helps us to identify three channel access architectures for NCSs:

Static Channel Access Mechanisms: Static protocols are random access methods with a fixed channel access probability. The access probability is independent of the current data or the past history of transmissions. The channel access policy f uses the information pattern $\mathbb{I}_k^{s,S} = \{\alpha_k\}$, where α_k is a binary random variable, independent of the initial state x_0 and the process noise $\boldsymbol{w}_0^{k-1}$. This information pattern turns the scheduling policy in this method into a binary random number generator, such as a coin flip. An example of such a mechanism could be any static contention-free mechanism, or p-persistent CSMA with no retransmissions.

Dynamic Channel Access Mechanisms: Dynamic protocols are random access methods with a channel access probability that evolves over time. The access probability is still independent of the current transmission data, but depends on the past history of transmission instants. The scheduling criterion f uses the information pattern $\mathbb{I}_k^{s,D} = \left\{\boldsymbol{\gamma}_0^{k-1}, \boldsymbol{\delta}_0^{k-1}\right\}$, where δ denotes the channel access indicator to the channel access request γ from this plant, as defined in Section 1.3. This information pattern induces memory into the scheduling policy. An example of such a mechanism could be p-persistent CSMA with retransmissions, or CSMA/CA.

Adaptive Channel Access Mechanisms: Adaptive protocols are random access methods with a channel access probability that depends on the current data packet, and possibly, evolves over time as well. The scheduling criterion f uses the information pattern $\mathbb{I}_k^{s,A} = \left\{\boldsymbol{x}_0^k, \boldsymbol{y}_0^{k-1}, \boldsymbol{\gamma}_0^{k-1}, \boldsymbol{\delta}_0^{k-1}, \boldsymbol{u}_0^{k-1}\right\}$, where x denotes the state of the plant ($\mathcal{P}$) and y denotes the measurement available across the network. This information pattern results in a state-based channel access mechanism, as discussed in the problem formulation in Chapter 1. An example of such a mechanism could be any of the methods presented in the following chapters of this thesis.

Both static and dynamic channel access mechanisms induce no dual effect (see Definition 2.1), and thus, we can use the results of Bar-Shalom and Tse (1974) to show that certainty equivalence holds. The optimal control costs can also be evaluated (Ramesh, 2011). An analysis of the structural properties of control systems with adaptive or state-based channel access mechanisms is presented in Chapter 3.

2.3 Networked Control Systems

In this section, we present an overview of research in NCSs. Control systems have seen rapid changes in the last forty years. Hybrid systems evolved with the introduction of computers and digital technology, and NCSs emerged as the need for networks in control settings arose. Murray et al. (2003) list NCSs as one of the future research directions in control, and provide an interesting overview of the motivations and origins of NCSs.

Control theory was already in use in the design of large industrial manufacturing systems by the early 70s. Such applications required modular, easily scalable, distributed control systems with integrated diagnostics. Point-to-point wiring was expensive to install and maintain for such large systems. Thus, these systems had to use a common-bus architecture, making these the first applications of closed-loop control through a shared network (Halevi and Ray, 1988a,b). Research on NCSs began as early as in the 70s, and continued through the late 80s and early 90s. The focus was then on common bus architectures, and the need for a suitable protocol to arbitrate access to the shared bus. These protocols introduced a varying delay in the closed-loop system, which were compared, and suitable protocols were identified (Ray, 1987; Lian et al., 2001). The effect of delays in a control system was studied by Nilsson (1998), and the stability of control systems under real-time delays was analyzed by Zhang et al. (2001). A network quality of service (QoS) was defined in terms of delay, with modifications to network protocols proposed to guarantee a certain QoS. Some of these methods included rate adaptations (Hong, 1995) and the prioritization of real-time traffic over other categories of traffic. Other methods included the use of deadbands to reduce the communication load in shared networks (Yook et al., 2002; Otanez et al., 2002).

Wireless networks introduce additional problems such as packet losses and data-rate constraints. These have been the focus of recent studies in NCSs (Hespanha et al., 2007). However, the original problems plaguing shared networks remain even

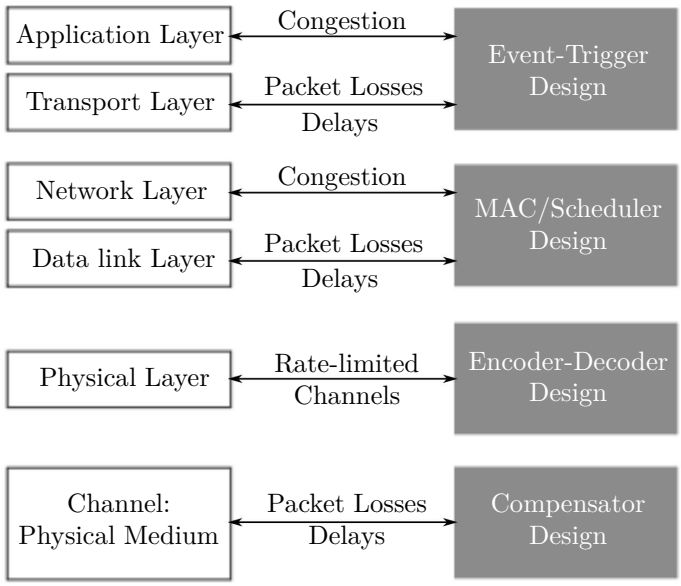


Figure 2.8: The network protocol is implemented as a stack consisting of some of the layers depicted above. We relate the problems tackled in NCSs literature to the layers in the stack. Interference from the physical medium, rate-limitations from the data channel and congestion from higher layers result in packet losses and delays.

today, as wireless networks are interference constrained. The medium cannot support simultaneous transmissions, requiring mechanisms that arbitrate channel access. Wireless networks make it hard to implement some of the older token-based or prioritized channel access methods, which were the chosen solutions for control systems on common bus architectures (Lian et al., 2001). Thus, there is a need to re-examine some of this work and to identify channel access methods suited to wireless NCSs.

In the rest of this section, we present a review of relevant work culled from the long history of NCSs. In our presentation of this work below, we look at the impact of each layer in a typical protocol stack on NCSs, as shown in Figure 2.8.

2.3.1 Control Design for Lossy Networks

The wireless medium is inherently lossy, and may cause packets to be dropped along the sensor link or the control link. Network protocols such as TCP have been designed to cope with such failures through the use of a retransmission policy, which ensures that the packet is eventually delivered to the destination. However, this property may not be very useful for a closed-loop system, as these systems are not delay tolerant. In NCSs, a packet that is delivered beyond an acceptable delay

is treated as a lost packet. The impact of such packet losses on optimal control and estimation has been well studied by Matveev and Savkin (2003), Smith and Seiler (2003), Schenato et al. (2007), Gupta et al. (2007), and others. Most of these studies model packet losses on the sensing and actuation links with Bernoulli processes. Some of these studies also consider packet loss distributions with correlations (Gupta et al., 2007).

These results are quite relevant to channel access methods as well. Many commonly used channel access mechanisms can be modelled using a Bernoulli process or a Markovian process, as shown in Figure 2.7. Then, the appropriate results from the packet loss literature can be applied to these systems. Another relevant result is the one on separation. Many of the above authors have established that the separation principle holds under both i.i.d. and correlated packet drop sequences, as long as the applied control input is made available to the observer. This is possible with a network protocol that returns an acknowledgement of a packet delivery. When such an acknowledgement is not available, the applied control signal is unknown. Then, the separation principle no longer holds, and the optimal control policy is hard to find (Schenato et al., 2007). Other results in this area include derivations of a critical probability of packet loss, below which the estimation error at the observer does not remain bounded, and also of upper and lower bounds on the achievable error covariance matrices.

2.3.2 Encoder Design for Limited Data-Rate Channels

A channel has a finite bandwidth, and the bandwidth constraint for wireless channels is more severe than that for wired channels. Thus, sensing and actuation links are permitted only limited data rates. Consequently, it is not possible to transmit a real-valued state or control signal on these links. The real values must be quantized at the source and decoded at the receiver, to reconstruct the original values. These tasks are performed by an encoder and a decoder, respectively. Many studies analyzing the design of an optimal encoder and decoder for a closed-loop system were carried out by Bansal and Başar (1989), Borkar and Mitter (1997), Tatikonda et al. (2004), Nair et al. (2007) and others.

An important result in this area is that certainty equivalence holds with a state-based encoder and decoder in the feedback loop. Bao et al. (2011) have commented on the importance of *side information* in realizing an architecture where the certainty equivalence principle holds. This relates to the availability of the applied control signal at the observer. Other results include designs for the optimal encoder-decoder pair.

2.3.3 MAC Design for Delayed and Congested Links

A MAC protocol determines the channel access strategy for multiple nodes sharing the same medium. Both contention-free and contention-based MACs introduce a delay in the closed-loop system. In addition, the retransmission scheme used by

a contention-based MAC determines the traffic contributed to the network by the MAC. A limited retransmission policy could reduce the reliability, or the probability of a successful transmission. Excessive retransmissions can increase the traffic in the network, leading to congestion, and consequently, zero throughput from the MAC (Rom and Sidi, 1990).

Scheduling strategies for real-time computer networks and systems have been an active area of research since the early 70s (Liu and Layland, 1973; Horn, 1974; Kay and Lauder, 1988), and some of the accomplishments in this area are well summarized by Buttazzo (2004). However, computer networks or processor scheduling do not present the same challenges in scheduling as controlling a physical process over a wireless network. In NCSs, the most important drawback of MACs was considered to be the time-varying delay they introduced into closed-loop systems. The impact of delays on a closed-loop system was studied by Nilsson (1998), who considered different time-varying delays on the sensor and control links. The delays are assumed to be independent random variables with known probability distributions. However, with the use of time-stamped data in the network, the delays are known when the measurement arrives. Nilsson derives the optimal control policy, and shows that it is a delay-dependent function of the current state and the past control. He also extends this to correlated delays across links.

Much of the earlier work presents a comparison between various MAC protocols, and evaluates their suitability to control applications, such as in Lian et al. (2001) and Ray (1987). Lian et al. (2001) compare CAN, Fieldbus and the DCF of IEEE 802.11, and conclude that CAN is better suited to networks with short and prioritized messages, while token bus is better suited to networks with large messages. However, neither of these can be adapted easily to wireless networks. A significant effort in this direction has been recently made by Pereira et al. (2007). They implement a CAN-like protocol for wireless nodes, called WiDOM. Another similar effort is Blackburst by Gotzhein and Kuhn (2008), which has been adapted to obtain a CAN-like protocol for wireless nodes. An important difference from CAN is that, due to the wireless medium, these protocols are not error free. They are prone to mis-detection of winners, hidden terminals, etc. that can result in collisions and loss in throughput. These aspects of the wireless medium alter any token bus or CAN-like protocol and make it randomized, irrespective of the design intent. We present and analyze a contention-based approach to CAN bus in Chapter 6.

The idea of using the state or measurement of a physical system to determine channel access has been prevalent for some time now (Walsh et al., 1999; Otanez et al., 2002; Yook et al., 2002). Walsh and Ye (2001) demonstrated that dynamic scheduling can result in significant benefits over static scheduling for NCSs. Deadbands around the nominal value were used to limit the use of the channel in (Otanez et al., 2002; Yook et al., 2002). We discuss this work, and extensions to it in Section 2.3.4. The deviation in the state from the nominal value was used to determine a priority in Try-Once-Discard (Walsh et al., 1999). Maximum error first is the prioritization principle used in this protocol, to guarantee input-to-state stability for deterministic systems with disturbances. The implementation of the original idea

was centralized, and required a network coordinator to collect and compare errors from the various physical processes in the network. This contention-free implementation has been extended to include effects of packet losses by (Tabbara and Nesić, 2008). Recently, a distributed implementation for this protocol has been conceived and implemented by (Christmann et al., 2014).

A related problem that deals with scheduling algorithms for networked plants with multiple sensors has seen many contributions over the years (Herring and Melsa, 1974; Baras and Bensoussan, 1989). These algorithms are often high-level application-specific algorithms, designed to cooperatively attain the best estimation or control performance in NCSs (Hall and Llinas, 1997; Savkin et al., 2000). It is difficult to classify these approaches as a multiple access protocol, as they often do not specify an implementation. This is especially true for wireless networks. Even so, many of these ideas are interesting from the point of view of the deterministic or stochastic policies they consider. Some sensor-scheduling algorithms for Markov models are considered by Gupta et al. (2006), and more recently by Farokhi and Johansson (2014) and Farokhi (2014).

The design of a channel access mechanism for wireless NCSs is still an open problem. Designing a multiple access protocol that provides sufficient reliability for control systems in the network, while retaining ease of implementation on wireless systems, is a challenge (Willig, 2008; Åkerberg et al., 2011). An important aspect of this problem is the ability of the channel access mechanism to adapt to the traffic in the network, and even reduce data rates when there is congestion. We present one approach to obtain such a solution in this thesis.

2.3.4 Event-based System Design for Congested Networks

Digital control systems often use the time-triggered paradigm, where a measurement is periodically sent to the controller to generate a control signal. Event-based systems provide an alternative, wherein only measurements that qualify as 'events' are sent to the controller. These systems could result in fewer transmissions (Åström and Bernhardsson, 1999; Otanez et al., 2002; Yook et al., 2002), which is an important consideration when multiple closed-loop systems use a shared network to communicate with their respective controllers. Event-based systems have been in use for very long, due to their ease of implementation. Accelerometers and gyros with pulse feedback were event-based. Relay systems with on-off control, satellite thrusters and sigma-delta modulators are event-based. Event-based control appears to be used even in biological systems (Wilson, 1999). A detailed overview of early event-based systems can be found in Åström (2008). However, it is only in recent years that an interest in the theoretical foundations of event-based systems has picked up. Early work showed that the same control performance can be achieved using fewer samples with event-based systems, for a single system (Tomovic and Bekey, 1966; Åström and Bernhardsson, 1999). Various event-triggering policies have been proposed and analyzed for different problem formulations, but mainly for stochastic and deterministic setups. We review some contributions from these

areas below. A more detailed review of these contributions can be found in Henningsson (2012) and also in Heemels et al. (2012).

Stochastic Event-based Sampling: Åström and Bernhardsson (2002) have shown that event-based sampling can be interpreted as Lesbegue sampling, which may be a useful alternative over periodic or Riemann sampling. An extension of this work was presented in Rabi (2006), where optimal event-triggering rules were shown to be derived by solving optimal stopping time problems. Furthermore, the joint choice of stopping times and control signals were derived mainly for first-order systems. This line of work has been pursued by Meng and Chen (2012) for second-order systems, and by Rabi et al. (2012) for a Markov state process with hard communication constraints.

An optimal method for reducing the communication load for remote estimation was presented in Xu and Hespanha (2004a). A smart sensor chooses when to transmit information to a remote observer, so as to minimize a long term average cost consisting of a weighted estimation error term and a communication penalty term. They show that the optimal communication policy is deterministic and stationary. The same authors also show that threshold rules perform quite well, compared to other sampling rules, in Xu and Hespanha (2004b). Interestingly, many of the optimal sampling rules in Rabi (2006) detect a level crossing in the plant state or output. These are the kind of policies considered in this thesis as well.

Henningsson et al. (2008) consider two sporadic event-triggering rules that implement a minimum inter-sampling constraint to facilitate implementability. They compare the performances obtained with periodic systems, and conclude that event-based systems can result in better control performance and reduced communication costs. Henningsson (2011) uses path constraints to model a mixed continuous and discrete time stochastic system. By converting these to constraints on the moments of the trajectory, the author derives upper and lower bounds on the control costs for a system that meets these constraints. Molin and Hirche (2009) deal with the optimal closed-loop design for event-based systems, which involves a joint selection of the event-triggering policy and control law. This is related to the work presented in Chapter 7. These authors present similar results in (Molin and Hirche, 2010, 2013). Lipsa and Martins (2011) use majorization to show that a symmetric event-triggering policy is optimal for remote estimation with first-order systems. Their work rests on the characterization of jointly optimal paging and registration policies by Hajek (2002) and Hajek et al. (2008). Xia et al. (2013) have used a Markov model to analyze an event-triggering policy based on the estimation error.

Deterministic Event-based Sampling: Tabuada (2007) used the input-to-state stability framework to identify an event-based sampling rule for nonlinear continuous time systems. A relative error criterion determines the event-triggering instants in this approach. An extension to output-based triggering was presented by Donkers and Heemels (2010, 2012). Wang and Lemmon (2008) use an exponentially decaying reference to bound a Lyapunov function, and thereby determine event-triggered sampling instants. Yu and Antsaklis (2011) present a dynamic output-based event triggering rule for passive systems. To facilitate easy scheduling of

event-triggered samples, a self-triggered approach has been proposed by Lemmon et al. (2007) and Anta and Tabuada (2010). In this approach, the control signal for the current time instant as well as the next sampling instant are identified in such a manner so as to guarantee stability. A model predictive control approach to this problem is presented by Henriksson et al. (2012), where the authors codesign the controller and the time interval to the next sample. They also present stability conditions and an off-line version of their optimal controller in the form of a look-up table.

Network of Event-based Systems: Although event-based systems were proposed to reduce congestion, most of the early work in this area dealt with a single plant. It is only in recent years that networks of event-based systems have been analyzed. Much of the work focussing on the design of event-based systems for a shared network does not explicitly deal with the problem of multiple access (Wang and Lemmon, 2011; Molin and Hirche, 2012). Others use protocols such as the CAN bus for wired networks (Anta and Tabuada, 2009), or dynamic real-time scheduling for multiple tasks on a single processor (Tabuada, 2007). These protocols are not well-suited to wireless networks, as we have discussed earlier (Akyildiz et al., 1999; Gummalla and Limb, 2000).

There had been a growing interest in analyzing a network of event-based systems with random access. This is not easy to accomplish as the multiple access channel introduces correlations between different streams of data packets (Cervin and Henningsson, 2008; Rabi and Johansson, 2009a). Some of the earlier work in this area includes an empirical analysis of event-based systems with CSMA/CA by Cervin and Henningsson (2008), which highlights the difficulties in analyzing such a network. The analysis by Rabi and Johansson (2009a) assumes independent packet losses in a network that uses Aloha. An extension of this analysis to certain CSMA-type protocols with contention-windows was provided by Rabi et al. (2010). A simple steady state model was presented by Henningsson and Cervin (2010), but with an idealized multiple access protocol that results in no collisions. More recently, event-based systems which use Aloha and Slotted Aloha have been analyzed by Blind and Allgöwer (2011a,b), but with an event-triggering policy that is not adapted to the network.

Many recent papers explicitly include a multiple access protocol in their analysis of event-based systems. An interesting work in this direction, by Weimer et al. (2012), also considers a policy to select the transmit and receive options of the radio transceiver in sensor nodes, in the context of networked estimation. A related approach to control radio modes using an event-triggering policy is presented by Cardoso de Castro et al. (2012a,b). Araujo et al. (2014) consider the codesign of a dynamic scheduling policy for the allocation of the contention-free slots in the IEEE 802.15.4 MAC, along with the event-triggering policy for a network of deterministic nonlinear systems.

2.4 Summary

In this chapter, we have reviewed some aspects of stochastic control theory and multiple access protocols that we use in the rest of this thesis. The problem formulation we used to explore stochastic control in Section 2.1 recurs in Chapters 3 and 7, along with the non-classical information pattern. We use the method of dynamic programming highlighted in this chapter, along with the concepts of dual effect, certainty equivalence and separation, in these later chapters. We will also use, and compare with, some of the multiple access protocols presented in this chapter, particularly p-persistent CSMA. We have also presented a review of related work in NCSs.

In this thesis, our aim is to design a channel access policy for a network of control systems that provides a control-specific guarantee. In doing so, we also examine the design of the control policy and the observer, when needed, in the feedback loop. Alongside, we model the interference from other network users, and explore the design of channel access policies that adapt to the traffic in the network.

Chapter 3

Structural Analysis

In the next three chapters, we present analysis and design methods for a network of event-based systems. Event-based sampling is an alternative to periodic sampling, where the state of the plant determines the next sampling instant. The resulting samples are often aperiodic. We look at event-based sampling from a network perspective, as a means to reduce transmissions within a network. Within this context, an event-triggering policy is a state-based channel access mechanism, comprising of a state-based scheduler and a contention resolution mechanism (CRM). As opposed to *agnostic* channel access methods, the state-based scheduler is capable of influencing the randomness of channel access in favour of the state of the plant in the control system. It accomplishes this by using the state to select packets to send to the CRM. Consequently, each plant sends fewer, but more important packets to the CRM for transmission across the network. This results in benefits for the entire network as well, due to the reduction in network traffic.

With the reduced traffic, the CRM is more likely to be able to successfully handle all the transmission requests, within the maximum number of retransmissions permitted by the protocol. Thus, the state-based scheduler takes on the role of regulating data flow from the source, while limiting the CRM's role to that of resolving contention between simultaneous channel access requests, by spreading traffic that arrives in bursts. This implies that the state has a bearing in deciding when to discard a packet. Furthermore, random decisions in the CRM do not impact channel access itself, only intersample delay.

By introducing a state-based channel access mechanism on the sensing link, we have introduced a nonlinear measurement with a dynamic measurement policy that depends on the state. This may result in many unexpected properties for a stochastic control system, such as those outlined in Chapter 2. Thus, we begin our analysis by examining the impact of a state-based scheduler on the control system. In this chapter, we seek to find structural properties that facilitate a simple and modular design for a control system in this network.

3.1 Contributions and Related Work

There are two main contributions in this chapter. The first contribution is an analysis of the impact of having a state-based scheduler in the closed-loop. Primarily, a state-based scheduler permits the information available at the controller to be altered with the plant state. This information is not entirely random, like in the case of packet losses due to a noisy channel (Schenato et al., 2007; Gupta et al., 2007), and it can result in a sharply asymmetrical estimation error, unlike in the case of encoder design over limited data rate channels (Tatikonda et al., 2004; Nair et al., 2007). It seems reasonable to ask if we can use the controller to move the plant state across the threshold and force a transmission. If this is possible, the controller plays two roles: the first one being to control the plant, and the second one being to control the information available at the next time step. This relates to the classical concept of a dual effect, as described by Feldbaum (1961) and expounded by Åström and Wittenmark (1995). The answer to this question determines the ease of optimal controller design, as the certainty equivalence principle might not hold if there is a dual effect (Bar-Shalom and Tse, 1974). We examine our system and find that there is a dual effect with a state-based scheduler in the closed-loop, and the certainty equivalence principle does not hold. Hence, the optimal state-based scheduler, estimator and controller designs are coupled. A restriction on the input arguments to the state-based scheduler, such that these arguments are no longer a function of the past control actions, renders the setup free of a dual effect, and enables the certainty equivalence principle to hold. These results can be seen as an interpretation, within the state-based scheduler setup, of the classical work on information patterns (Ho, 1980), dual effect, certainty equivalence and separation by Witsenhausen (1971) and Bar-Shalom and Tse (1974), and on adaptive control by Feldbaum (1961), Åström and Wittenmark (1995) and many others (Filatov and Unbehauen, 2000).

The second contribution of this chapter is on the *dual predictor architecture*, which is our proposed solution to the state-based scheduler design problem. In this architecture, the state-based scheduler thresholds the squared difference of the innovation contained in the latest measurement to the estimator across the network. This results in an optimal certainty equivalent controller, and a simple observer which generates the minimum mean-squared error (MMSE) estimate. Tuning parameters in the state-based scheduler in this architecture based on the current network traffic can result in a scheduling law that guarantees a probabilistic performance. This is not easy to show, in general, as the performance analysis of a closed-loop system with a state-based scheduler in a multiple access network is a difficult problem (Cervin and Henningsson, 2008; Rabi and Johansson, 2009a). However, we illustrate the guaranteed performance using simulations, and thus claim that the state-based scheduler we propose results in a network-aware event-triggering mechanism.

A similar strategy for the selective transmission of important packets has previously been proposed from the more general perspective of reducing network traffic

(Otanez et al., 2002). This approach has driven the design of event-based sampling systems (Rabi, 2006; Tabuada, 2007), which have been shown to outperform periodically sampled systems under certain conditions (Åström and Bernhardsson, 2002; Cervin and Henningsson, 2008; Rabi and Johansson, 2009a). More recently, such systems have been analyzed for estimation over networks (Lipsa and Martins, 2011; Battistelli et al., 2012), but the extensions to the control setting remain incomplete. We approach the networked control system (NCS) problem from a different perspective, but one that leads to a *network-aware* design of event triggering methods. By this, we mean that the event triggering function will be chosen to achieve a certain control performance with the current network traffic, and also, by taking into account the impact it will have on the resulting network traffic.

The rest of the chapter is organized as follows. In Section 3.2, we present the problem formulation. In Section 3.3, we derive theoretical results for the case when full state information is available, with and without exogenous network traffic. In Section 3.4, we present the dual predictor architecture. We look at an extension to output-based systems in Section 3.5. We present a counterexample to validate our results on the dual effect, along with other examples that illustrates our notion of network-aware event-triggering, in Section 3.6.

3.2 Preliminaries

Consider a network of control systems, where the communication between the individual sensors and controllers of different control loops occurs through a shared network, as shown in Figure 3.1. We present a problem setup for this network, along with a few important definitions, in this section.

3.2.1 Problem Formulation

We consider a network of M control loops, as shown in Figure 3.2(a). Each control loop, for $j \in \{1, \ldots, M\}$, consists of a plant $\mathcal{P}^j$, a state-based scheduler $\mathcal{S}^j$ and a controller $\mathcal{C}^j$. The loops share access to a common medium on the sensor link. A closed-loop system in this network can be modelled as shown in Figure 3.2(b), with the index j dropped for simplicity. The block $\mathcal{N}$ represents the network as seen by this loop, and the block $\mathcal{R}$ denotes the Contention Resolution Mechanism (CRM), which determines access to the network. Each of the blocks in Figure 3.2(b) is explained below.

Plant: The plant $\mathcal{P}$ has state dynamics given by

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{3.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and w_k is i.i.d. zero-mean Gaussian with covariance matrix R_w. The initial state x_0 is zero-mean Gaussian with covariance matrix R_0.

State-Based Scheduler: There is a local scheduler $\mathcal{S}$, situated in the sensor node, between the plant and the controller, which decides if the state is to be sent

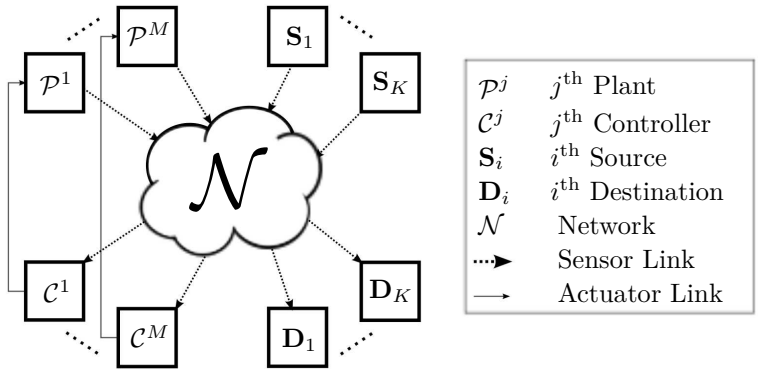
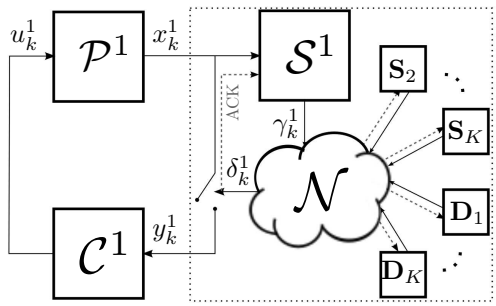
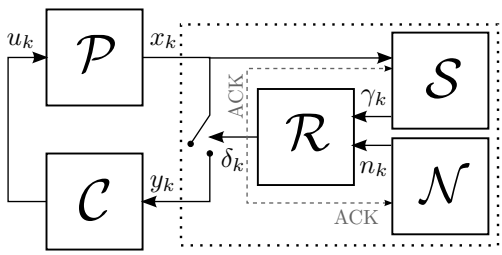


Figure 3.1: A network of M control loops, with each loop consisting of a plant $\mathcal{P}^j$ and a controller $\mathcal{C}^j$ for $j \in \{1, \ldots, M\}$. The loops share access to a common medium on the sensor link, along with K other communication flows from generic source-destination pairs. The controllers and actuators communicate over dedicated networks, not shared links.



(a) A multiple access scenario for NCSs



(b) The MA model for each closed-loop system

Figure 3.2: A plant ($\mathcal{P}^1$), state-based scheduler ($\mathcal{S}^1$) and controller ($\mathcal{C}^1$) share the network ($\mathcal{N}$) with $M-1$ other closed-loop systems with state-based schedulers ($\mathcal{S}^j, j \in \{2, \ldots, M\}$), and K generic devices ($\mathbf{D}^i, i \in \{1, \ldots, K\}$), in (a). A model, from the perspective of a single closed-loop system in the network, is depicted in (b).

across the network or not. The scheduler output is denoted γ_k, where $\gamma_k \in \{0,1\}$. It takes a value 1 when the state x_k is scheduled to be sent and 0 otherwise. The scheduling criterion is denoted by the policy f, which is defined on the information pattern of the scheduler $\mathbb{I}_k^s$, and is given by

$$\gamma_k = f_k(\boldsymbol{u}_0^{k-1}, \omega_k^s) \,, \tag{3.2}$$

where f_k is not a constant function of $\boldsymbol{u}_0^{k-1}$, i.e., $\exists\ \boldsymbol{u}_1, \boldsymbol{u}_2$ such that $f(\boldsymbol{u}_1, \cdot) \neq f(\boldsymbol{u}_2, \cdot)$. The scheduling policy f_k is also a function of $\omega_k^s \in \Omega_k^s$, and Ω_k^s is the σ-algebra generated by the information set at the scheduler, given by $\mathbb{I}_k^s = \{\boldsymbol{x}_0^k, \boldsymbol{y}_0^{k-1}, \boldsymbol{\gamma}_0^{k-1}, \boldsymbol{\delta}_0^{k-1}\}$. Here, we use bold font to denote a sequence of variables such as $\boldsymbol{a}_t^T = \{a_t, a_{t+1}, \ldots, a_T\}$. Note that an explicit acknowledgement (ACK) of a successful transmission is required for δ_k to be available to the scheduler. The scheduler output γ_k is now a function of the state, as suggested by the epithet 'state-based scheduler'.

Network: The network $\mathcal{N}$ generates exogenous traffic, as is indicated by $n_k \in \{0,1\}$. It takes a value 1 when the network traffic attempts to access the channel, and 0 otherwise. The network traffic is considered to be stochastic, as it could be generated by another control loop, or by any other communicating node in the network. Thus, n_k is a binary random variable, which is not required to be i.i.d. We say that there is no exogenous network traffic if $n_k \equiv 0$, for all k.

CRM: The CRM block $\mathcal{R}$ resolves contention between multiple simultaneous channel access requests, i.e., when $\gamma_k = 1$ and $n_k = 1$. If the CRM resolves the contention in favour of our control loop, $\delta_k = 1$, and otherwise 0. The CRM can be modelled as the medium access control (MAC) channel response $\mathcal{R}$, with MAC output δ_k given by

$$\delta_k = \mathcal{R}(\gamma_k, n_k) \,. \tag{3.3}$$

For brevity, we also define $\bar{\delta}_k = 1 - \delta_k$, which takes a value 1 when the packet is not transmitted. The MAC channel response $\mathcal{R}$ is modelled as a discrete memoryless channel at the sampling time scale, requiring the CRM to resolve contention with respect to this packet before the next sampling instant. This translates to a limitation on the sampling rates supported by the model.

Measurement: The measurement across the network is denoted y_k. It is a non-linear function of the state x_k, and is given by

$$y_k = \delta_k x_k = \begin{cases} x_k & \delta_k = 1 \,, \\ \emptyset & \delta_k = 0 \,, \end{cases} \tag{3.4}$$

where $\emptyset$ indicates an erasure. A successful transmission results in the full state being sent to the controller. However, even non-transmissions convey information as the scheduler output δ_k can be treated as a noisy and coarsely quantized measurement of the state x_k.

Controller: The control law g denotes an admissible policy for the finite horizon N defined on the information pattern of the controller, $\mathbb{I}_k^c$, and is given by

$$u_k = g_k(\omega_k^c) \,, \tag{3.5}$$

where, $\omega_k^c \in \Omega_k^c$, and Ω_k^c is the σ-algebra generated by the information pattern $\mathbb{I}_k^c = \left\{ \boldsymbol{y}_0^k, \boldsymbol{\delta}_0^k, \boldsymbol{u}_0^{k-1} \right\}$. The objective function, defined over a horizon N is given by

$$J(f,g) = \mathbb{E}\left[x_N^\top Q_0 x_N + \sum_{s=0}^{N-1} (x_s^\top Q_1 x_s + u_s^\top Q_2 u_s) \right] , \tag{3.6}$$

where Q_0 and Q_1 are positive semi-definite weighting matrices and Q_2 is positive definite.
In the rest of the chapter, we address the following questions:

1. For a NCS with a state-based scheduler, what is the optimal control policy which minimizes the cost J in (3.6)?

2. Can we find a simple, but sub-optimal, closed-loop system architecture for the given NCS?

To answer the first question, we need to examine whether the system exhibits a dual effect. This also requires us to check if we can find an equivalent system, in the sense of Witsenhausen, for which certainty equivalence holds. The second question requires us to identify restrictions on the scheduling policy f, which can ensure separation of the scheduler, controller and observer.

3.2.2 Definitions and Properties

We present a few definitions and properties that are used in the rest of the chapter.

Definition 3.1 (Uncontrolled Process)**.** *An auxiliary uncontrolled process ($\bar{\mathcal{P}}$) can be defined for any closed-loop system, by removing the effect of the applied control signals from the state. The resulting uncontrolled state is denoted $\bar{x}_k$, and given by*

$$\bar{x}_k = A^k x_0 + \sum_{\ell=1}^{k} A^{\ell-1} w_{k-\ell} \, . \tag{3.7}$$

Last Received Packet Index: The time index of the last received packet is denoted τ_k at time k (illustrated in Figure 3.3), and for $-1 \leq \tau_k \leq k$, it is given by

$$\tau_k = \max\{t : \delta_t = 1, \text{ for } -1 \leq t \leq k \, , \delta_{-1} = 1\} \, . \tag{3.8}$$

An iterative relationship for τ_k can be found as

$$\tau_k = \bar{\delta}_k \tau_{k-1} + \delta_k k, \quad \tau_{-1} = -1 \, . \tag{3.9}$$

If a packet arrives at current time k, the last received packet index $\tau_k = k$. But, if there is no packet at time k, then the last received packet index is the same as the last received packet index from time $k-1$, i.e., $\tau_k = \tau_{k-1}$. This implies that $\tau_k \in \{-1, \ldots, k\}$.

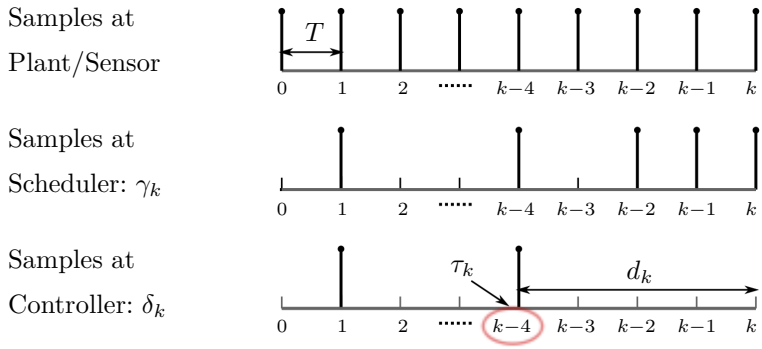


Figure 3.3: An illustration of the delay since the last received packet (d_k) and the index of the last received packet (τ_k).

Dual Effect and Certainty Equivalence: The dual effect and certainty equivalence properties were defined in Chapter 2, in Definitions 2.1 and 2.3, respectively. We reuse those definitions in this chapter.

Correlated Network Noise: We state a property of feedback systems with state-based schedulers that share a contention-based multiple access network. Even if the initial states and disturbances of all the plants in the network are independent, the contention-based MAC introduces a correlation between the traffic sources, as noted in Cervin and Henningsson (2008) and Rabi and Johansson (2009a).

Lemma 3.1. *For a closed-loop system defined by (3.1)–(3.5), the exogenous network traffic indicated by n_k is correlated to the state of the plant x_k.*

Proof. The MAC output δ_{k-1} is a function of the state x_{k-1} and the indicator of network traffic n_{k-1}, from (3.2) and (3.3). The control signal u_{k-1} is a function of the MAC output δ_{k-1} from (3.5), and is applied through feedback to the plant. Thus, x_k and γ_k are correlated to δ_{k-1}. Similarly, the network traffic from other closed-loop systems (and its indicator n_k) is correlated to δ_{k-1}, and consequently, x_k. □

3.3 Optimal Controller Design

We present the main results of this chapter in this section. We first analyze the effects of a state-based scheduler on a control loop with no exogenous network traffic, i.e., $n_k \equiv 0$. As a consequence of this, the MAC output is equal to the scheduler output, i.e., $\delta_k = \gamma_k$. We show that there is a dual effect of the control signal, and that the scheduling policy must be restricted from using the past control inputs for the certainty equivalence principle to hold. We illustrate this for a second-order system with a state-based scheduler in Figure 3.4, and show that the controller

is not oblivious to the scheduler boundaries. We extend our results to the case with exogenous network traffic.

3.3.1 Dual Effect with State-based Scheduling

We observe that the estimation error is a function of the applied controls, and that it does not satisfy the condition for no dual effect in (2.5). Thus, we have the following result.

Theorem 3.2. *For the closed-loop system defined by (3.1)–(3.5), with no exogenous network traffic, i.e., $n_k \equiv 0$, the control signal has a dual effect of order $r = 2$.*

Proof. We examine the estimation error, and show that it is not equivalent to the estimation error generated by the uncontrolled process $\bar{\mathcal{P}}$ (from Definition 3.1) in place of $\mathcal{P}$. Thus, we prove that the estimation error covariance is a function of the applied controls $\boldsymbol{u}_0^{k-1}$.

From (3.4), we know that a successful transmission results in the full state being sent to the controller, whereas a non-transmission conveys only a single bit of information (δ_k is binary) about the state to the controller. Thus, the estimate, $\hat{x}_{k|k} \triangleq \mathbb{E}[x_k|\mathbb{I}_k^c]$, is given by

$$\hat{x}_{k|k} = \delta_k x_k + \bar{\delta}_k \, \mathbb{E}[x_k|\mathbb{I}_k^c, \delta_k = 0] \; .$$

The variable δ_k cannot be removed from the above expression due to the asymmetry in the resolution of the received information with and without a transmission. The scheduler outcome, and consequently δ_k, are influenced by the applied control inputs $\boldsymbol{u}_0^{k-1}$ in a state-based scheduler such as (3.2). The estimation error, defined as $\tilde{x}_{k|k} \triangleq x_k - \mathbb{E}[x_k|\mathbb{I}_k^c]$, is given by

$$\tilde{x}_{k|k} = (x_k - \mathbb{E}[x_k|\mathbb{I}_k^c, \delta_k = 0]) \cdot \bar{\delta}_k \; , \tag{3.10}$$

and thus depends on δ_k. The estimation error when there is no transmission is defined as $\tilde{x}^0_{k|k} \triangleq x_k - \mathbb{E}[x_k|\mathbb{I}_k^c, \delta_k = 0]$, and is given by

$$\begin{aligned}
\tilde{x}^0_{k|k} &= A^k x_0 + \sum_{\ell=1}^{k} A^{\ell-1}(Bu_{k-\ell} + w_{k-\ell}) \\
&\quad - \mathbb{E}[A^k x_0 + \sum_{\ell=1}^{k} A^{\ell-1}(Bu_{k-\ell} + w_{k-\ell})|\mathbb{I}_k^c, \delta_k = 0] \\
&= \bar{x}_k - \mathbb{E}[\bar{x}_k|\mathbb{I}_k^c, \delta_k = 0] \; ,
\end{aligned}$$

where $\bar{x}_k$ is the state of the uncontrolled process (see Definition 3.1). As shown above, the additive terms containing the past applied controls can indeed be removed with knowledge of the applied controls at the estimator. However, δ_k and $\bar{\delta}_k$,

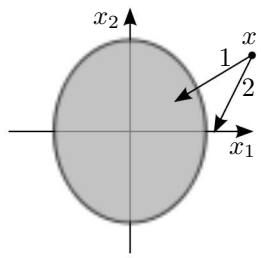


Figure 3.4: For states $x \in \mathbb{R}^2$, we define a state-based scheduling policy f that generates an event when the state lies outside the shaded region. The resulting estimation error within the shaded region is non-zero, whereas it is zero outside this region. The dual role of the controller, shown in Theorem 3.2, arises from the incentive to move some states along path 2 (and remain outside of the shaded region), as compared to path 1 (this only reduces the variance of the state).

the second factor of the product in (3.10), remain a function of the applied controls, and cannot be generated by the uncontrolled process alone.

Thus, the estimation error is always dependent on the applied controls and this distinguishes the current problem from other problems, such as the classical LQG setup. The error covariance, $P_{k|k} \triangleq \mathbb{E}[\tilde{x}_{k|k}\tilde{x}_{k|k}^{\top}|\mathbb{I}_k^c]$, is given by

$$P_{k|k} = \bar{\delta}_k \cdot (\mathbb{E}[\tilde{x}_{k|k}\tilde{x}_{k|k}^{\top}|\mathbb{I}_k^c, \delta_k = 0]) \ . \tag{3.11}$$

The covariance $P_{k|k}$ is zero if the scheduling criterion in (3.2) is fulfilled, and non-zero otherwise. Through δ_k, $P_{k|k}$ is a function of the past controls. Hence, $P_{k|k}$ does not satisfy the condition (2.5) required to have no dual effect. Thus, the system (3.1)–(3.5) exhibits a dual effect of order $r = 2$. □

In this setup, there is an incentive for the control policy to modify the estimation error along with controlling the plant, as illustrated in Figure 3.4. Thus, the controller might choose to keep the state out of the shaded region to improve the estimation error for future time steps, even if this results in an increased variance of the state.

3.3.2 Witsenhausen Equivalence

Suppose that every state-based scheduler f, from (3.2), can be transformed into an innovations-based scheduler $\tilde{f}$, such as

$$\gamma_k = \tilde{f}_k(\tilde{\omega}_k^s) \ , \tag{3.12}$$

where, $\tilde{\omega}_k^s \in \tilde{\Omega}_k^s$, $\tilde{\Omega}_k^s$ is the σ-algebra generated by the information pattern $\tilde{\mathbb{I}}_k^S = \{x_0, \boldsymbol{w}_0^{k-1}\}$, and the policy $\tilde{f}_k$ is defined in advance, and can be realized without any knowledge of the control policies used in the system. The output of such a

scheduler is only a function of the innovations, and not a function of the applied controls $\boldsymbol{u}_0^{k-1}$. This distinguishes an innovations-based scheduler from a state-based scheduler (3.2). The innovations-based scheduler does not result in a dual effect of the control signal, as we show below. Even so, we cannot replace a state-based scheduler (3.2) in a closed-loop system with an innovations-based scheduler (3.12), unless it results in an equivalent control design. We now examine the question of equivalent designs, following Witsenhausen (1971).

Definition 3.2. *An equivalent design, in the sense of Witsenhausen, g_{eq} for the optimal controller g^*, which minimizes the cost criterion (3.6) for the system defined by (3.1)–(3.5), satisfies the equivalence relation given by*

$$\boldsymbol{u}^* = \Upsilon(\boldsymbol{\omega}, g^*) = \Upsilon(\boldsymbol{\omega}, g_{eq}) \,, \tag{3.13}$$

where Υ is obtained by recursive substitution for the control signals in the system equations with the respective control policy and the primitive random variables $\boldsymbol{\omega}_k = [x_0, \boldsymbol{w}_0^{k-1}]$.

For brevity, we adopt the following notation. Let $\{\mathcal{P}, f_1, g_1\}$ denote a system with a plant $\mathcal{P}$ given by (3.1), with f_1 as the given scheduler and g_1 as the optimal controller for the cost in (3.6). We now note the following result.

Theorem 3.3. *Let there be no exogenous network traffic, i.e., $n_k \equiv 0$. For any state-based scheduler f and innovations-based scheduler $\tilde{f}$ that result in the same schedules, the corresponding optimal designs, g^* and $\tilde{g}$, respectively, are not equivalent in the sense of Witsenhausen.*

Proof. Definition 3.2 requires the control signals obtained using the policies g^* and $\tilde{g}$ to be equal. In this proof, we find the optimal control policies for $\tilde{g}$ and g^*, and show that they do not result in the same control signals.

For the optimal control policy, which minimizes the quadratic cost J in (3.6), to be certainty equivalent, we need to find a solution to the Bellman equation (Åström, 1970), which is a one-step minimization of the form

$$V_k = \min_{u_k} \; \mathbb{E}[x_k^\top Q_1 x_k + u_k^\top Q_2 u_k + V_{k+1} | \mathbb{I}_k^c] \,. \tag{3.14}$$

In general, without defining a structure for the estimator, the solution to the functional is given in the form of

$$V_k = \mathbb{E}\left[x_k^\top S_k x_k | \mathbb{I}_k^c\right] + s_k \,, \tag{3.15}$$

where S_k is a positive semi-definite matrix and both S_k and s_k are not functions of the applied control signals $\boldsymbol{u}_0^{k-1}$, see Bar-Shalom and Tse (1974). We now prove that a solution of this form can be found for $\{\mathcal{P}, \tilde{f}, \tilde{g}\}$, but not for $\{\mathcal{P}, f, g^*\}$.

First consider the system $\{\mathcal{P}, \tilde{f}, \tilde{g}\}$. We denote the state and control signals of this system as $\tilde{x}_k$ and $\tilde{u}_k$. At time N, the functional has a trivial solution with

$S_N = Q_0$ and $s_N = 0$. This solution can be propagated backwards, in the absence of a dual effect. To show this, we use the principle of induction, and assume that a solution of the form (3.15) holds at time $k+1$. Then, at time k, we have

$$\begin{aligned} V_k &= \min_{u_k} \mathbb{E}[\tilde{x}_k^\top Q_1 \tilde{x}_k + \tilde{u}_k^\top Q_2 \tilde{u}_k + \tilde{x}_{k+1}^\top S_{k+1} \tilde{x}_{k+1} + s_{k+1} | \mathbb{I}_k^c] \\ &= \min_{u_k} \mathbb{E}[\tilde{x}_k^\top (Q_1 + A^\top S_{k+1} A) \tilde{x}_k | \mathbb{I}_k^c] + \mathsf{tr}\{S_{k+1} R_w\} + \mathbb{E}[s_{k+1} | \mathbb{I}_k^c] \\ &\quad + \tilde{u}_k^\top (Q_2 + B^\top S_{k+1} B) \tilde{u}_k + \hat{\tilde{x}}_{k|k}^\top A^\top S_{k+1} B \tilde{u}_k + \tilde{u}_k^\top B^\top S_{k+1} A \hat{\tilde{x}}_{k|k} \ , \end{aligned}$$

where $\hat{\tilde{x}}_{k|k} \triangleq \mathbb{E}[\tilde{x}_k | \mathbb{I}_k^c]$. The optimal control is found to be

$$\tilde{u}_k = -L_k \hat{\tilde{x}}_{k|k} \ , \ L_k = (Q_2 + B^\top S_{k+1} B)^{-1} B^\top S_{k+1} A \ . \tag{3.16}$$

Substituting the expression for $\tilde{u}_k$ into V_k gives us a solution of the form in (3.15), with

$$\begin{aligned} S_k &= Q_1 + A^\top S_{k+1} A - A^\top S_{k+1} B (Q_2 + B^\top S_{k+1} B)^{-1} B^\top S_{k+1} A \ , \\ s_k &= \mathbb{E}[s_{k+1} | \mathbb{I}_k^c] + \mathsf{tr}\{S_{k+1} R_w\} + \mathsf{tr}\{A^\top S_{k+1} B (Q_2 + B^\top S_{k+1} B)^{-1} B^\top S_{k+1} A P_{k|k}\} \ , \end{aligned} \tag{3.17}$$

where the matrix S_k is positive semi-definite and not a function of the applied controls $\tilde{\boldsymbol{u}}_0^{k-1}$. The scalar s_k is not a function of the applied controls $\tilde{\boldsymbol{u}}_0^{k-1}$ if and only if $P_{k|k}$ has no dual effect (Bar-Shalom and Tse, 1974). From the expression for the error covariance $P_{k|k}$ (3.11), it is clear that a scheduling criterion that is not a function of the past control actions, such as (3.12), results in no dual effect. Under this condition, s_k is not a function of the applied controls $\tilde{\boldsymbol{u}}_0^{k-1}$ and the proof by induction is complete. Since the optimal control signal (3.16) is a function of only the estimate $\hat{\tilde{x}}_{k|k}$, the certainty equivalence principle holds.

Now, consider the system $\{\mathcal{P}, f, g^*\}$ with state x_k and control u_k^*. Solving the backward recursion as we did above, we find that V_N and V_{N-1} have a solution of the form (3.15), with $S_N = Q_0$ and $s_N = 0$, and S_{N-1} and s_{N-1} given by (3.17) with $k = N-1$. However, V_{N-2} results in a different minimization problem for this system because of the dual effect in $\{\mathcal{P}, f, g^*\}$, as indicated next. The optimal control signal u_{N-2}^* can be obtained by solving

$$\begin{aligned} \frac{\partial V_{N-2}}{\partial u_{N-2}^*} &= \frac{\partial}{\partial u_{N-2}^*} \big(\mathsf{tr}\{(Q_2 + B^\top S_N B) \mathbb{K}_{N-2} \cdot \mathbb{E}[P_{N-1|N-1} | \mathbb{I}_{N-2}^c]\} \big) \\ &\quad + 2(u_{N-2}^*)^\top (Q_2 + B^\top S_{N-1} B) + 2 \hat{x}_{N-2|N-2}^\top A^\top S_{N-1} B = 0 \ , \end{aligned}$$

where we set $\mathbb{K}_{N-2} = (Q_2 + B^\top S_{N-1} B)^{-1} A^\top S_N B (Q_2 + B^\top S_N B)^{-1} B^\top S_N A$. Multiplying the above expression with $(Q_2 + B^\top S_{N-1} B)^{-1}$ from the right and using (3.16) to denote $u_{N-2}^{CE} = -L_{N-2} \hat{x}_{N-2|N-2}$, we obtain the simpler equation

$$\frac{\partial}{\partial u_{N-2}^*} \Big(\mathsf{tr}\{\mathbb{K}_{N-2} \, \mathbb{E}[P_{N-1|N-1} | \mathbb{I}_{N-2}^c]\} \Big) + 2((u_{N-2}^*)^\top - (u_{N-2}^{CE})^\top) = 0 \ , \tag{3.18}$$

The first term in (3.18), related to the estimation error covariance $P_{N-1|N-1}$, is not equal to zero as implied by the dual effect property from Theorem 3.2. Due to this term, the above minimization problem is not linear, and thus, the solutions u^{CE}_{N-2} and u^{*}_{N-2} are not equal. Since u^{CE}_{N-2} has the same form as $\tilde{u}_{N-2}$, we also note that $\tilde{u}_{N-2}$ and u^{*}_{N-2} have very different forms. From this point on, the cost-to-go for the optimal control policy g^* does not have a solution of the form given by (3.15). Hence, the control signals $\{\tilde{\boldsymbol{u}}\}_0^{N-3}$ and $\{\boldsymbol{u}^*\}_0^{N-3}$ will not be equal. Now, the joint distribution of all system variables could be quite different for schedulers $\tilde{f}$ and f. Thus, the described transformation of the scheduling criterion does not result in an equivalent design. □

The above theorem provides us a motivation for using a state-based scheduler, despite the inherent difficulties associated with the closed-loop design. Due to the dual effect, the optimal control action takes on two roles. One, to control the plant, and the other, to probe the plant state which could result in an improved estimate (Åström and Wittenmark, 1995). The innovations-based scheduler results in a simpler closed-loop design, as shown in (3.16)–(3.17). However, a probing action cannot be implemented in any controller in this setup due to the lack of a dual effect. Thus, the resulting control actions for the closed-loop systems with the state-based and innovations-based schedulers are not the same.

3.3.3 Conditions for Certainty Equivalence

From the previous discussions, it is clear that a scheduling criterion independent of the past control actions, such as the innovations-based scheduler, results in no dual effect. This result is presented below.

Corollary 3.4. *For the closed-loop system defined by (3.1)–(3.5), with no exogenous network traffic, i.e., $n_k \equiv 0$, the optimal controller, with respect to the cost in (3.6), is certainty equivalent if and only if the scheduling decisions are not a function of the applied control actions, such as in (3.12).*

Proof. In the proof of Theorem 3.3, it is clear from (3.16) that the optimal control policy $\tilde{g}$ for the system $\{\mathcal{P}, \tilde{f}, \tilde{g}\}$ is certainty equivalent.

To show the necessity of this condition for certainty equivalence, we need to show that if the optimal control signal has the form in (3.16) at time k, then the scheduling policy is not a function of the controls for $n < k$, for all k. Accordingly, assume that the optimal control signal is given by (3.16) for $k = N - 1, \ldots, n+1$. Then, the optimal cost-to-go is of the form in (3.15), at time $n+1$ and $s_{n+1} = \sum_{k=n+1}^{N-1} \mathbb{E}[\mathsf{tr}\{A^\top S_{k+1} B(Q_2 + B^\top S_{k+1} B)^{-1} B^\top S_{k+1} A P_{k|k} + S_{k+1} R_w\} | \mathbb{I}_k^c]$, when written out explicitly. We know that the optimal control signal u_n is obtained by minimizing (3.14) at time n. This control signal will have the form in (3.16) for all $Q_2 > 0$ only if s_{n+1} is independent of u_n, or if the estimation error covariances $P_{k|k}$, for $k = \{n+1, \ldots, N-1\}$, are not a function of u_n. From the result in Theorem 3.2, this is only possible when the scheduling policy is not a function of u_n.

Since this is true for $n = 0, \ldots, N-1$, the scheduling policy must not be a function of $\boldsymbol{u}_0^{k-1}$. □

Corollary 3.4 provides us with a restriction on the scheduler to ensure certainty equivalence. Note that the resulting design is not equivalent to the optimal design, as shown in Theorem 3.3.

3.3.4 Effect of State-based Schedulers with Exogenous Network Traffic

In this subsection, we analyze the effects of a state-based scheduler on the control loop in the *presence* of exogenous network traffic. Thus, we have $n_k \not\equiv 0$ and a channel output given by (3.3). Recall from Lemma 3.1, that the network traffic indicator n_k is correlated to the state of the plant x_k. The certainty equivalence principle need not hold for plants where the measurement noise is correlated to the process noise (Bar-Shalom and Tse, 1974). To focus on the effect of state-based schedulers on the closed-loop system, the results presented in the previous subsection did not include exogenous network traffic. Now, we derive some of the above results for the system in the presence of exogenous network traffic.

Lemma 3.5. *For the closed-loop system defined by (3.1)–(3.5), the control signal has a dual effect of order $r = 2$.*

Proof. The MAC output δ_k (3.3) is clearly still a function of the applied controls, through the state-based scheduler outcome. Thus, the estimation error covariance $P_{k|k}$, in (3.11), remains a function of the applied controls $\boldsymbol{u}_0^{k-1}$. Since $P_{k|k}$ does not satisfy the condition (2.5) required to have no dual effect, we see that the system (3.1)–(3.5) exhibits a dual effect of order $r = 2$. □

With the above result, Theorem 3.3 can be easily extended to include the case with exogenous network traffic. However, it is not as straightforward to extend Corollary 3.4. When the measurement noise is correlated to the process noise, certainty equivalence need not hold. To see why, recall the proof of Theorem 3.3, where we derive a solution of the form $V_k = \mathbb{E}[x_k^\top S_k x_k | \mathbb{I}_k^c] + s_k$ for the Bellman equation (3.14). Now, if w_k is correlated to the variables in the information set $\mathbb{I}_k^c$, specifically $\boldsymbol{n}_0^k$, the minimization with respect to u_k in (3.16) must include the term $\mathsf{tr}\{S_{k+1} R_w\}$. Then, the optimal controller will not have the form shown in (3.16), and certainty equivalence will not hold.

We need to prove that w_k is independent of $\boldsymbol{n}_0^k$ for the certainty equivalence principle to hold, which we do below.

Corollary 3.6. *For the closed-loop system defined by (3.1)–(3.5), the optimal controller, with respect to the cost criterion (3.6), is certainty equivalent if the exogenous network traffic indicator n_k is independent of the process noise w_k, and, if the*

scheduling decisions are not a function of the applied controls, i.e., if

$$\gamma_k = \check{f}_k(\check{\omega}_k^s) \,, \tag{3.19}$$

where, $\check{\omega}_k^s \in \check{\Omega}_k^s$*, and* $\check{\Omega}_k^s$ *is the* σ*-algebra generated by the information set* $\check{\mathbb{I}}_k^S = \{x_0, \boldsymbol{w}_0^{k-1}, \boldsymbol{n}_0^{k-1}\}$.

Proof. Note that n_k is only correlated to $\boldsymbol{\delta}_0^k$ and thus, to the signals $\boldsymbol{w}_0^{k-1}$, from Lemma 3.1. As the process noise is i.i.d, n_k is independent with respect to w_k. A scheduler of the form (3.19) is not a function of the applied controls, and thus, certainty equivalence holds. □

3.4 Closed-Loop System Architecture

In this section, we find that symmetric scheduling policies simplify the observer design. We propose a dual predictor architecture for the closed-loop system, which results in a separation of the scheduler, observer and controller designs.

3.4.1 Observer Design

Due to the non-linearity of the problem, the estimate in general can be hard to compute. However, the estimation error is reset to zero with every transmission, as we send the full state. Consider one such reset instance, a time k such that $\delta_k = 1$. The state is sent across the network, $y_k = x_k$, so the estimate $\hat{x}_{k|k} = x_k$. A suitable control signal u_k is found and applied to the plant, which results in the next state x_{k+1}. Now, the scheduler can generate one of two outcomes. We consider each case, and find an expression for the estimate below:

a) $\delta_{k+1} = 0$: We need an estimate of w_k. We use the scheduler output as a coarse quantized measurement to generate this, as follows:

$$\begin{aligned} \hat{x}_{k+1|k+1} &= \mathbb{E}[x_{k+1} | \mathbb{I}_{k+1}^c, \delta_{k+1} = 0] \\ &= Ax_k + Bu_k + \mathbb{E}[w_k | \delta_{k+1}(\grave{f}(w_k)) = 0] \,, \end{aligned} \tag{3.20}$$

where, $\mathbb{E}[w_k|\delta_{k+1}(\grave{f}(w_k)) = 0] = \sum_{\gamma\in\{0,1\}} \mathbb{E}[w_k|\grave{f}(w_k) = \gamma, \delta_{k+1} = 0] \cdot \mathrm{P}(\gamma_{k+1} = \gamma|\delta_{k+1} = 0)$ and $\grave{f}(w_k) \equiv f(Ax_k + Bu_k + w_k|x_k, u_k)$.

b) $\delta_{k+1} = 1$: The estimation error is zero as $\hat{x}_{k+1|k+1} = x_{k+1}$.

The transformation to $\grave{f}$ in (3.20), is not intended to remove the dual effect, but merely serves to remove the known variables from the expression. The dual effect has influenced the packet's transmission, i.e., the value of δ_{k+1}. To understand this expression clearly, we look at the next time instant. Now a signal u_{k+1} is generated, and applied to the plant. We note that $x_{k+2} = A^2x_k + ABu_k + Bu_{k+1} + Aw_k + w_{k+1}$. The state x_{k+2} is either sent to the controller or not depending on the

scheduler outcome δ_{k+2}. Again, we look at both cases, and derive an expression for the estimate:

a') $\delta_{k+2} = 0$: We now need to estimate $Aw_k + w_{k+1}$, as the rest is completely known from x_{k+2}. We use both scheduler outputs δ_{k+1} and δ_{k+2} to generate an estimate of the unknown variables as

$$\hat{x}_{k+2|k+2} = A^2 x_k + ABu_k + Bu_{k+1} + \mathbb{E}[Aw_k + w_{k+1} | \delta_{k+1}(\grave{f}(w_k)) = 0, \delta_{k+2}(\grave{f}(Aw_k + w_{k+1})) = 0] \, .$$

b') $\delta_{k+2} = 1$: Again, $\hat{x}_{k+2|k+2} = x_{k+2}$.

This process can be continued recursively through a non-transmission burst, until finally a measurement is received and the estimation error is reset to zero. Thus, the observer computes the estimate at any time k as

$$\hat{x}_{k|k} = \begin{cases} x_k, & \delta_k = 1, \\ A^{k-\tau_k} x_{\tau_k} + \sum_{s=1}^{k-\tau_k} A^{s-1} B u_{k-s} & \\ \quad + \mathbb{E}[\sum_{s=1}^{k-\tau_k} A^{s-1} w_{k-s} | \delta(\grave{f}_k), .., \delta(\grave{f}_{\tau_k+1}) = 0], & \delta_k = 0, \end{cases} \tag{3.21}$$

where τ_k is the time index of the last received measurement at time k, as defined in (3.8), and the argument to the function $\grave{f}_t$ is given by the term $\sum_{s=1}^{t-\tau_t} A^{s-1} w_{t-s}$.

3.4.2 State-based Scheduler Design: Symmetric Schedulers

The computation of the term $\mathbb{E}[\sum_{s=1}^{k-\tau_k} A^{s-1} w_{k-s} | \delta(\grave{f}_k), .., \delta(\grave{f}_{\tau_k+1}) = 0]$ makes the estimate (3.21) hard to evaluate, because the quantized noise is not Gaussian. As a sub-optimal approach, consider the scheduling criterion given by any symmetric map $f^{sym}(r) = f^{sym}(-r)$ with

$$\gamma_k = f^{sym}(\sum_{s=1}^{k-\tau_{k-1}} A^{s-1} w_{k-s}) \, . \tag{3.22}$$

Since τ_k is not defined without the MAC output δ_k in (3.8), we replace it with τ_{k-1}, which is also a measure of the non-transmission burst. Choosing the scheduler in this manner results in a zero mean estimate from the quantized noise when there is no transmission. Now, the estimate (3.21) is easy to compute and the observer can be designed without knowledge of the scheduling policy. Also, a certainty equivalent control can be applied. This observation is summarized below, and is used to design the scheduler presented in Section 3.4.3.

Proposition 3.7. *For the closed-loop system defined by (3.1)–(3.6), the use of the symmetric scheduling policy (3.22) implies that certainty equivalence holds, and it also results in separation in design between the estimator and scheduler.*

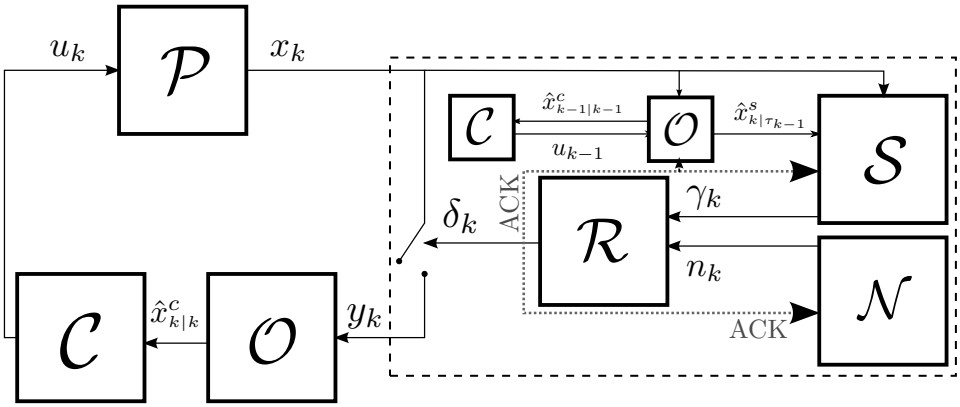


Figure 3.5: State-based Dual Predictor Architecture: the innovations to the observer serve as input to the scheduler. The resulting setup is certainty equivalent. The observer is simple, and computes the MMSE estimate.

3.4.3 The Dual Predictor Architecture

In this section, we examine closed-loop design of the complete system, including scheduler, observer and controller. From the results of Lemma 3.5 and Proposition 3.7, it is clear that the scheduler, observer and controller designs are coupled, in general. It is not possible to design the optimal scheduling policy independently and combine it with a certainty equivalent controller and optimal observer to get the overall optimal closed-loop system. At the same time, solving for the jointly optimal scheduler, observer and controller is a hard problem.

Thus, we propose an architecture, shown in Figure 3.5, for a design of the state-based scheduler, and the corresponding optimal controller and observer. There are two estimators in this architecture, and hence, we call it a dual predictor architecture. This architecture has been referred to previously in the context of mobile networks (Xu et al., 2004). The scheduler, observer and controller blocks are described below.

Scheduler ($\mathcal{S}$): The scheduler output γ_k is given by

$$\gamma_k = f(x_k, \hat{x}_{k|\tau_{k-1}}) = \begin{cases} 1, & |x_k - \hat{x}_{k|\tau_{k-1}}|^2 > \epsilon, \\ 0, & \text{otherwise}, \end{cases} \tag{3.23}$$

where $\epsilon > 0$ is a given threshold and $\hat{x}_{k|\tau_{k-1}}$ is the estimate at the controller at time k if the current packet is not scheduled for transmission. To realize such a scheduling policy, the observer must be replicated within the scheduler, and for the observer to be able to subtract the applied control, the controller must also be replicated within the scheduler. An explicit ACK is required to realize this information pattern, as indicated in Figure 3.5.

Observer ($\mathcal{O}$): The input to the observer is $y_k = \delta_k x_k$. The observer generates the estimate $\hat{x}_{k|k}$ as given by

$$\hat{x}_{k|k} = \bar{\delta}_k \hat{x}_{k|\tau_k} + \delta_k x_k \ . \tag{3.24}$$

Recall that $\bar{\delta}_k = 1 - \delta_k$ takes a value 1 when the packet is not transmitted. In such a case, the estimate is given by $\hat{x}_{k|\tau_k}$, a model-based prediction from the last received data packet at time τ_k. This estimate is given by

$$\hat{x}_{k|\tau_k} = A\hat{x}_{k-1|k-1} + Bu_{k-1} \ . \tag{3.25}$$

Controller ($\mathcal{C}$): The controller generates the signal u_k based on the estimate alone, as given by

$$u_k = -L_k \hat{x}_{k|k} \ , \tag{3.26}$$

where L_k is defined in (3.16).

Note that the scheduling criterion described in (3.23) can be rewritten as

$$|x_k - \hat{x}_{k|\tau_{k-1}}|^2 = |A\tilde{x}_{k-1|k-1} + w_{k-1}|^2 = |\tilde{x}_{k|\tau_{k-1}}|^2 \ .$$

Here, we use $\hat{x}_{k|\tau_{k-1}}$ as τ_k is not defined without δ_k. The criterion $|\tilde{x}_{k|\tau_{k-1}}|^2 \leq \epsilon$ captures the per-sample variance of the estimation error, when no transmission is scheduled. Taking expectations on both sides, we get $\mathsf{tr}\{P_{k|\tau_{k-1}}\} \leq \epsilon$. The scheduler attempts to threshold the variance of the estimation error, but this cannot be guaranteed in a network with multiple traffic sources. Also, note that the scheduling policy is a symmetric function of its arguments, as in Proposition 3.7. We now state the main result of this section.

Theorem 3.8. *For the closed-loop system given by the plant (3.1), the state-based dual predictor (3.23)–(3.26), and the cost criterion (3.6), it holds that*

i. The estimate (3.24) minimizes the mean-squared estimation error.

ii. The control signal does not have a dual effect.

iii. The certainty equivalence principle holds and the optimal control law is given by (3.26).

iv. The LQG cost is given by

$$\begin{aligned} J_{DP} = \hat{x}_0^\top S_0 \hat{x}_0 + \mathsf{tr}\{S_0 P_0\} + \sum_{n=0}^{N-1} \mathsf{tr}\{S_{n+1} R_w\} \\ + \sum_{n=0}^{N-1} \mathsf{tr}\{(L_n^\top (Q_2 + B^\top S_{n+1} B) L_n) P_{n|n}\} \ , \end{aligned} \tag{3.27}$$

where $P_{k|k}$ is the error covariance of the estimate at the observer, with $S_N = Q_0$ and S_k obtained by backward iteration of (3.17).

Proof. We know that

$$A^{k-\tau_k} x_{\tau_k} + \sum_{n=1}^{k-\tau_k} A^{n-1} B u_{k-n} = A\hat{x}_{k-1|k-1} + B u_{k-1} \ ,$$

$$\mathbb{E}[\sum_{\ell=1}^{k-\tau_k} A^{\ell-1} w_{k-\ell} | \boldsymbol{\delta}_{\tau_k+1}^{k} = 0] = 0 \ ,$$

where the last expression results from the use of a symmetric scheduling policy. Substituting for these terms in the expression for the estimate in 3.21, we get

$$\mathbb{E}[x_k | \mathbb{I}_k^c] = \begin{cases} x_k, & \delta_k = 1, \\ A\hat{x}_{k-1|k-1} + B u_{k-1}, & \delta_k = 0. \end{cases}$$

Thus, the estimate in (3.24) is the MMSE estimate (Kailath et al., 2000).

The error covariance at the estimator is given by (3.11), where, from (3.23) and (3.3), it is clear that the scheduler outcome γ_k and the MAC output δ_k do not depend on the applied controls $\boldsymbol{u}_0^{k-1}$. Thus, the error covariance satisfies the definition in (2.5), and the control signal in this architecture does not have a dual effect.

From the above conclusion, note that the scheduling policy in (3.23) is of the form (3.19). Thus, from Corollary 3.6, we know that the optimal controller for this setup is certainty equivalent. Then, the optimal control signal is given by (3.16), which has the same form as the controller in this architecture (3.26). The expression for the control cost remains the same as in the case with partial state information, and is given by (3.27). □

Thus, the dual predictor architecture results in a sub-optimal but simplified closed-loop system.

3.5 Extensions and Discussions

In this section, we extend the above results to an output-based system. We also identify the existence of a dual effect when the cost function penalizes network usage and when the transmission, with a state-based scheduler, occurs over limited data-rate channels. Finally, we discuss the dual effect property that we have encountered in this problem with respect to other NCS architectures.

3.5.1 Measurement-based Scheduler

We now consider a system without full state information, but with co-located measurements. We show that by placing an optimal observer, a Kalman Filter (KF) at the sensor, to estimate the state of the linear plant, and basing the scheduler

decisions on this estimate, instead of on the measurement, we are able to establish the same conclusions as before.

Consider a linear plant with a state z_k, and a measurement m_k given by

$$z_{k+1} = Az_k + Bu_k + w_{z,k} \ , \quad m_k = Cz_k + v_{z,k} \ , \tag{3.28}$$

where $w_{z,k}$ is i.i.d. zero-mean Gaussian with covariance matrix $R_{w,z}$. The initial state z_0 is zero-mean Gaussian with covariance matrix $R_{z,0}$. Also, the measurement $m \in \mathbb{R}^m$ and the matrix $C \in \mathbb{R}^{n \times m}$. The measurement noise $v_{z,k}$ is a zero mean i.i.d. Gaussian process with covariance matrix $R_{v,z} \in \mathbb{R}^{m \times m}$, and it is independent of $w_{z,k}$.

We can place a KF at the sensor node, which receives every measurement m_k from the sensor and updates its estimate ($\hat{z}^s_{k|k}$) as

$$\hat{z}^s_{k|k} = A\hat{z}^s_{k-1|k-1} + Bu_{k-1} + K_{f,k}e_k \ , \tag{3.29}$$

where $K_{f,k}$ denotes the KF gain and e_k denotes the innovation in the measurement. The innovation is Gaussian with zero-mean and covariance $R_{e,k}$. The error covariances for the predicted estimate and the filtered estimate are denoted P^s_k and $P^s_{k|k}$, respectively. These terms are given by

$$\begin{aligned} e_k &= m_k - C(A\hat{z}^s_{k-1|k-1} + Bu_{k-1}) \ , \\ K_{f,k} &= P^s_k C^\top R^{-1}_{e,k} \ , R_{e,k} = CP^s_k C^\top + R_{v,z} \ , \\ P^s_k &= AP^s_{k-1|k-1}A^\top + R_{w,z} \ , P^s_{k|k} = P^s_k - K_{f,k}R_{e,k}K^\top_{f,k} \ . \end{aligned}$$

If we use the estimate to define a new state, such that $x_k \triangleq \hat{z}^s_{k|k}$, we have a linear plant disturbed by i.i.d. Gaussian process noise $w_k = K_{f,k}e_k$. Thus, we have re-established the problem setup from section 3.2.1, and the results from before can be applied to this plant. Note that the scheduler is now defined with respect to the estimate $\hat{z}^s_{k|k}$ and not the measurements m_k. However, the scheduler output remains a function of the state and the measurement, through the estimate.

3.5.2 Penalizing Network Usage

We have shown, in the proofs of Theorem 3.2 and Theorem 3.3, that the applied controls play a significant role in a state-based scheduler and cannot be removed from the scheduler inputs to create an equivalent setup without a dual effect. However, the minimizing solution to a cost criterion can render the effect of the applied controls redundant. To see an example of this, consider the problem of finding the jointly optimal scheduler-controller pair for the classical LQG cost criterion in (3.6). Since there is no penalty on using the network, the optimal scheduler policy is to transmit all the time. Now, the structure of the closed-loop system does not resemble the one presented in Theorem 3.2, and consequently, that result does

not hold. In this scenario, there is no incentive for the controller to influence the transmissions and the jointly optimal scheduler-controller pair $(f^{\mathbb{1}}, g^{\mathbb{1}})$ is given by

$$f^{\mathbb{1}} : \delta_k^{\mathbb{1}} = 1 \; \forall \; k \; , \quad g^{\mathbb{1}} : u_k^{\mathbb{1}} = -L_k x_k \; \forall \; k \; , \tag{3.30}$$

where L_k is given in (3.16). Note that in the rest of this chapter, we do not consider finding the jointly optimal scheduler-controller pair, as the use of a contention-based MAC does not permit us to choose the schedule sequence.

Now, consider a cost criterion which penalizes the use of the network, such as

$$J_\Lambda = \min_{\boldsymbol{u}_0^{N-1}, \boldsymbol{\delta}_0^{N-1}} \mathbb{E}\left[x_N^\top Q_0 x_N + \sum_{s=0}^{N-1} \left(x_s^\top Q_1 x_s + u_s^\top Q_2 u_s \right) + \sum_{s=0}^{N-1} \Lambda \delta_s \right] , \tag{3.31}$$

where Q_0,Q_1 and Q_2 are positive definite weighting matrices and $\Lambda > 0$ is the cost of using the network. The optimal state-based scheduling policy chooses a schedule in relation to the penalty Λ, such that the average network use, i.e., $\mathbb{E}[\delta_k]$, decreases as Λ increases. Thus, we state the following result.

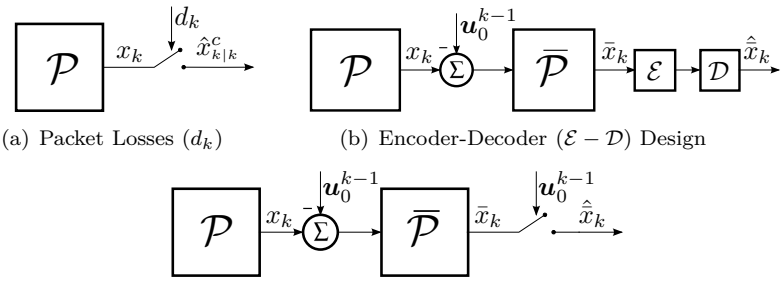
Proposition 3.9. *For the closed-loop system defined by (3.1)–(3.5), with no exogenous network traffic, the control signals derived from the jointly optimal scheduler-controller pair, which minimize the cost criterion in (3.31), exhibit a dual effect of order $r = 2$.*

Proof. It is easy to show that the scheduler-controller pair $(f^{\mathbb{1}}, g^{\mathbb{1}})$ does not minimize the cost in (3.31). Now, the scheduler uses the policy in (3.2) to select packets to send across the network. Thus, the closed-loop system has the same structure as in Theorem 3.2, and there is a dual effect of order $r = 2$ for any control signal in this setup. □

Proposition 3.9 provides the controller an incentive to modify the transmission outcome. We examine the above joint optimization problem in detail in Chapter 7. Using the results of Lemma 3.5, the above results can be extended to include the effect of exogenous network traffic.

3.5.3 Using a Rate-Constrained Channel

Our proof of the dual effect in Theorem 3.2 relies on the asymmetry in the resolution of the received information; the full state is sent with a transmission and only a single-bit quantized encoding of the state is sent when there is no transmission. However, data channels are generally rate-constrained, and a full state can never be sent. If the encoder-decoder pair on the sensor link uses R bits of information the estimation error at the controller can be written as $\tilde{x}_{k|k} = \delta_k \cdot (x_k - \mathbb{E}[x_k|\mathbb{I}_k^c, \delta_k = 1]) + \bar{\delta}_k \cdot (x_k - \mathbb{E}[x_k|\mathbb{I}_k^c, \delta_k = 0])$, in place of (3.10). Note that δ_k, and consequently the applied controls, cannot be removed from the above expression, unless the estimation error with and without a transmission result in the same expression,



(a) Packet Losses (d_k)

(b) Encoder-Decoder ($\mathcal{E} - \mathcal{D}$) Design

(c) State-based Scheduler

Figure 3.6: The estimate is not influenced by the applied controls in (a) and (b), with knowledge of the applied controls. In contrast, the applied controls cannot be removed from the decision process in (c).

i.e., $x_k - \mathbb{E}[x_k|\mathbb{I}^c_k, \delta_k = 1] = x_k - \mathbb{E}[x_k|\mathbb{I}^c_k, \delta_k = 0]$ for $R > 1$. Hence, there is a dual effect with a state-based scheduler, even when using a rate-constrained channel for transmission.

3.5.4 Relation to Other NCS Architectures

The dual effect and certainty equivalence properties have been noted previously in other NCS problems. We discuss these occurrences and the connections to our problem setup below.

Packet Drops over a Lossy Network: Packet drops in a lossy network are not influenced by the applied controls (Figure 3.6(a)). Hence, certainty equivalence holds, when there are packet drops on the sensor link (Gupta et al., 2007). However, when there are packet drops on the actuator link, separation holds only if there is an ACK of packets received or lost (Schenato et al., 2007).

Importance of Side Information: In any NCS problem, the classical information pattern must be reconstructed for the certainty equivalence principle to hold (Witsenhausen, 1968). This may require one or more side information channels to convey ACKs of received packets back to the transmitters (Bao et al., 2011; Schenato et al., 2007).

Limited Data Rate Channels: This problem differs from our setup in the sense that the encoder output is the only measurement available across the channel, and this potentially contains the same number of information bits, see Figure 3.6(b). In Tatikonda et al. (2004), the applied controls are shown to not influence the estimation error.

Event-based Systems: The results we have encountered in this chapter show that the applied controls can push the state across the scheduler threshold, and influence the transmission outcome, as illustrated in Figure 3.4. This is a consequence

Table 3.7: A comparison of control costs with (J_{SS}) and without (J_{CN}) a state-based scheduler in the closed-loop

Plant Type	$\mathcal{P}^{[T1]}$	$\mathcal{P}^{[T2]}$	$\mathcal{P}^{[T3]}$
J_{CN}	45.3074	10.0028	6.1213
J_{SS}	23.5785	8.3489	5.3803

of the unequal information in the measurement y_k, with and without a transmission, see Figure 3.6(c). A similar problem with a cost function such as (3.31), has been dealt with in Molin and Hirche (2009, 2010). They use a transformation similar to the one presented for the encoder design problem in Nair et al. (2007). There are, however, subtleties in defining an equivalence class for a state-based scheduler: using an equivalent scheduler need not result in an equivalent design, as shown in Theorem 3.3.

3.6 Examples

We present three examples in this section. The first example describes the problem setup, and illustrates the motivation for the problem. The second example illustrates the results of Theorem 3.2 and Theorem 3.3, which identify the dual role of the applied controls towards the information available to the controller. A counterexample is also presented, in which we identify some controllers which exploit this dual role. The third example illustrates the dual predictor architecture and provides an example of network-aware event triggering.

3.6.1 An Example of a Multiple Access NCS

This example illustrates the role of a state-based scheduler in our problem formulation in Section 3.2.1, where a number of closed-loop systems share a contention-based multiple access network on the sensor link. We use a p-persistent carrier sense multiple access (CSMA) protocol in the MAC. The observer and controller are chosen for simplicity of design, not as optimizers of any cost. We look at the performance of this network of control loops, with and without the state-based scheduler.

We consider a heterogenous network of 20 scalar plants, indexed by $j \in \{1, \ldots, 20\}$. There are three different types of plants, $\mathcal{P}^{[T1]}$,$\mathcal{P}^{[T2]}$ and $\mathcal{P}^{[T3]}$, given by

$$x_{k+1}^{(j)} = a^{[i]} x_k^{(j)} + u_k^{(j)} + w_k^{(j)} \ , \tag{3.32}$$

where $a^{[i]} \in \{1, 0.75, 0.5\}$, and $R_w^{[i]} \in \{1, 1.5, 2\}$, for the plant $\mathcal{P}^{[Ti]}$. The systems numbered $j \in \{1, \ldots, 6\}$ are of type $\mathcal{P}^{[T1]}$, $j \in \{7, \ldots, 13\}$ are of type $\mathcal{P}^{[T2]}$ and $j \in$

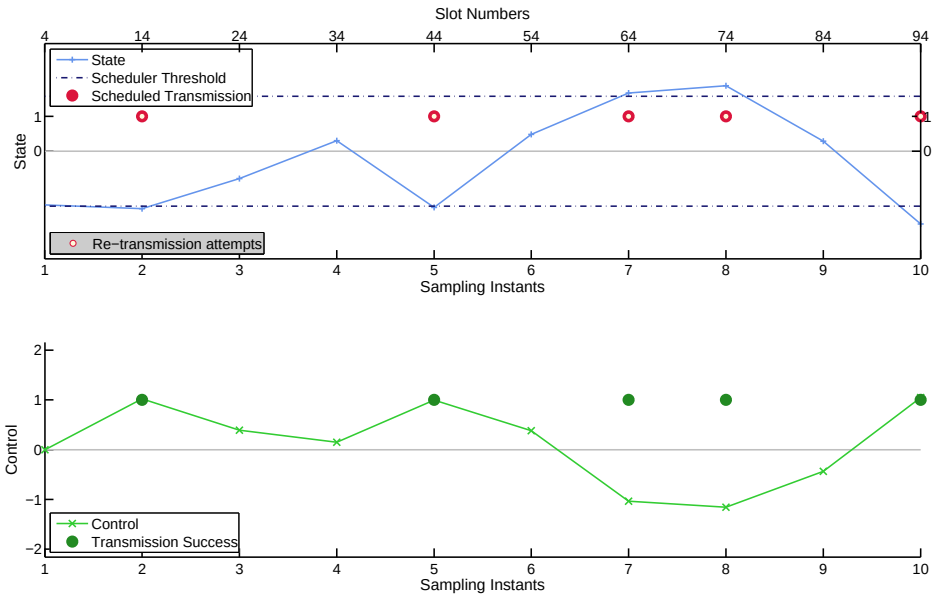


Figure 3.8: The state and the control signal with the channel use pattern: the red dots denote transmission requests, the white inner dots denote MAC re-transmission attempts, and the green dots denote transmission success. Note that the requested bound on the state, which is marked with a dotted line, is sometimes exceeded due to network traffic. Also, the control signal corresponds closely to the state only when there is a successful transmission.

$\{14, \ldots, 20\}$ are of type $\mathcal{P}^{[T3]}$. The plants are sampled with different periods given by $T^{[i]} \in \{10, 20, 25\}$, for the different types of plants, respectively. The state-based scheduler uses the criterion $x_k^{(j)^2} > \epsilon^{(j)}$. A p-persistent MAC, with synchronized slots, which permits three retransmissions is used. The persistence probability is given by $p_\alpha^{(r)}$, where r denotes the retransmission index, and $p_\alpha^{(r)} \in \{1, 0.75, 0.5\}$ for $r \in \{1, \ldots, 3\}$. The LQG criterion in (3.6), with $N = 10$ and $Q_0 = Q_1 = Q_2 = 1$ is used to design a certainty equivalent controller (3.16) as an ad hoc policy, not an optimal one, as we know from Corollary 3.6. The observer calculates a simple estimate as given by (3.24)-(3.25).

We look at the performance of a closed-loop system in this network without a state-based scheduler, i.e., when $\epsilon^{(j)} = 0$ for all j. The cost of controlling the plants in the contention-based network is denoted $J_{\text{CN}}^{[i]}$, and the values are listed in Table 3.7. We compare these values to the costs obtained with a state-based

scheduler in the closed-loop system, denoted $J_{\text{SS}}^{[i]}$, when $\epsilon^{(j)} = 2.5$. There is a marked improvement with a state-based scheduler in the closed-loop. Figure 3.8 depicts the state and the control signal for the first plant in this network, when a state-based scheduler is used. The above improvement is obtained due to fewer collisions in the contention-based MAC. The non-zero scheduling threshold reduces the traffic in the network, and increases the probability of a successful transmission for all the plants in the network.

3.6.2 A 2-Step Horizon Example

We now look at a simple example to see the computational difficulties in identifying optimal estimates and controls for a system with a state-based scheduler in the closed-loop. We also show that for a scheduler such as $\tilde{f}$ in Section 3.3.2, which renders the control signal free of a dual effect, the entire plant is altered, so the equivalence construction does not work. Finally, as a proof of the dual effect associated with a scheduler such as f in (2), we present a counterexample, obtained through simulations.

Consider a scalar plant, given by $x_{k+1} = ax_k + bu_k + w_k$, with $R_0 = R_w = 1$. The scheduling law is given by

$$\delta_k = \begin{cases} 1 & x_k \geq 0.5 \,, \\ 0 & \text{otherwise} \,. \end{cases} \tag{3.33}$$

Our aim is to find both the optimal controller, with dual effect, and the certainty equivalent controller for the innovations-based scheduler and show that these result in different control actions for the same scheduling sequence. The controllers are designed to minimize the LQG cost (3.6), for a horizon of two steps, i.e., $N = 2$, and with $Q_0, Q_1, Q_2 > 0$. We first derive the optimal controller with dual effect. Then, for the same schedule, we define the certainty equivalent controller, assuming that an innovations-based scheduler of the form $\tilde{f}$ in (3.12) has been designed. We compare the resulting control actions, and comment on the differences.

Estimator: The estimates $\hat{x}_{0|0}$ and $\hat{x}_{1|1}$ are obtained using (3.21). The estimation error covariances $P_{0|0}$ and $P_{1|1}$ are presented in Appendix A. Since the estimation error is non-Gaussian, we need to derive the probability density functions of the estimation errors at each time instant. This makes the computation of the estimation errors and the error covariances hard.

Optimal Controller: To solve for the optimal control signals, we use V_1 and V_0 from (3.14). The complete derivations of V_1 and V_0 are presented in Ramesh (2011). We find the control signal u_1 that minimizes V_1, and get

$$u_1 = -\frac{abQ_0}{Q_2 + b^2 Q_0} \hat{x}_{1|1} \,. \tag{3.34}$$

Then, to find u_0, we take a partial derivative of the expression for V_0 with respect

to u_0 and get

$$\frac{\partial V_0}{\partial u_0} = 2u_0(Q_2 + b^2 S_1) + 2\hat{x}_{0|0} abS_1 + \frac{a^2 Q_0^2 b^2}{Q_2 + b^2 Q_0} \cdot \frac{\partial}{\partial u_0} \left(\mathbb{E}[P_{1|1} | \mathbb{I}_0^c] \right) = 0 \, . \quad (3.35)$$

The optimal u_0 can be obtained by substituting for $P_{1|1}$ and solving the resulting equation.

Certainty Equivalent Controller: For the same scheduler outcomes δ_0, δ_1 obtained through an innovations-based scheduler which has no dual effect, the certainty equivalent controller gives us the control signals

$$\begin{aligned} u_1 &= -\frac{AbQ_0}{Q_2 + b^2 Q_0} \hat{x}_{1|1} \, , \\ u_0 &= -\frac{AbS_1}{Q_2 + b^2 S_1} \hat{x}_{0|0} \, . \end{aligned} \quad (3.36)$$

Note that the u_1 is found by minimizing V_1, which results in the same expression as for the optimal controller (3.34). However, when there is no dual effect, the last term in (3.35) vanishes, and u_0 for the certainty equivalent controller is obtained by solving

$$2u_0(Q_2 + b^2 S_1) + 2\hat{x}_{0|0} abS_1 = 0 \, . \quad (3.37)$$

Discussion: A comparison of the control signals for the certainty equivalent controller (3.36) with u_1 and u_0 obtained in (3.34) and (3.35), shows that the signal u_1 remains the same. However, u_0 is different, and displays a dual effect in the optimal controller. From (3.37), it is clear that the additional term in (3.35) alters the solution for the optimal controller. This observation can be explained as follows. In a controller with a dual effect, the control signal can be chosen to probe the plant state in order to improve the quality of the estimate. However, there is no motive in improving the estimate in a one-step optimization process. Thus, u_1 is the same for both controllers. When the optimization is performed over two steps, a probing effect in the first step can improve the estimate and the corresponding control applied in the next step. Thus, u_0 is different for the optimal controller.

Counterexample: To illustrate further the existence of the dual effect in the state-based scheduler setup discussed above, we consider an explicit numerical example with parameters $a = 2$, $b = 1$, $R_0 = 1$ and $R_w = 100$ for the linear plant, and a cost function with $Q_0 = 100$, $Q_1 = 1$ and $Q_2 = 1$. Finding a u_0 that solves (3.35) is hard. Instead, we evaluate the cost of using a certainty equivalent controller with a state-based scheduler such as in (3.33) and compare it with alternative controllers $u_0 = -L_0 \hat{x}_{0|0}$, which use a different value for the control gain L_0. We choose a range of values for L_0 centered around the certainty equivalent control gain L_0^{CE}, and plot the control costs obtained against the control gain in Figure 3.9, for different values of the scheduler threshold. The certainty equivalent control cost is marked by a dotted line in all the plots, while the cost of using an alternate controller is plotted with a solid line. For each of the scheduling thresholds shown in Figure 3.9, there

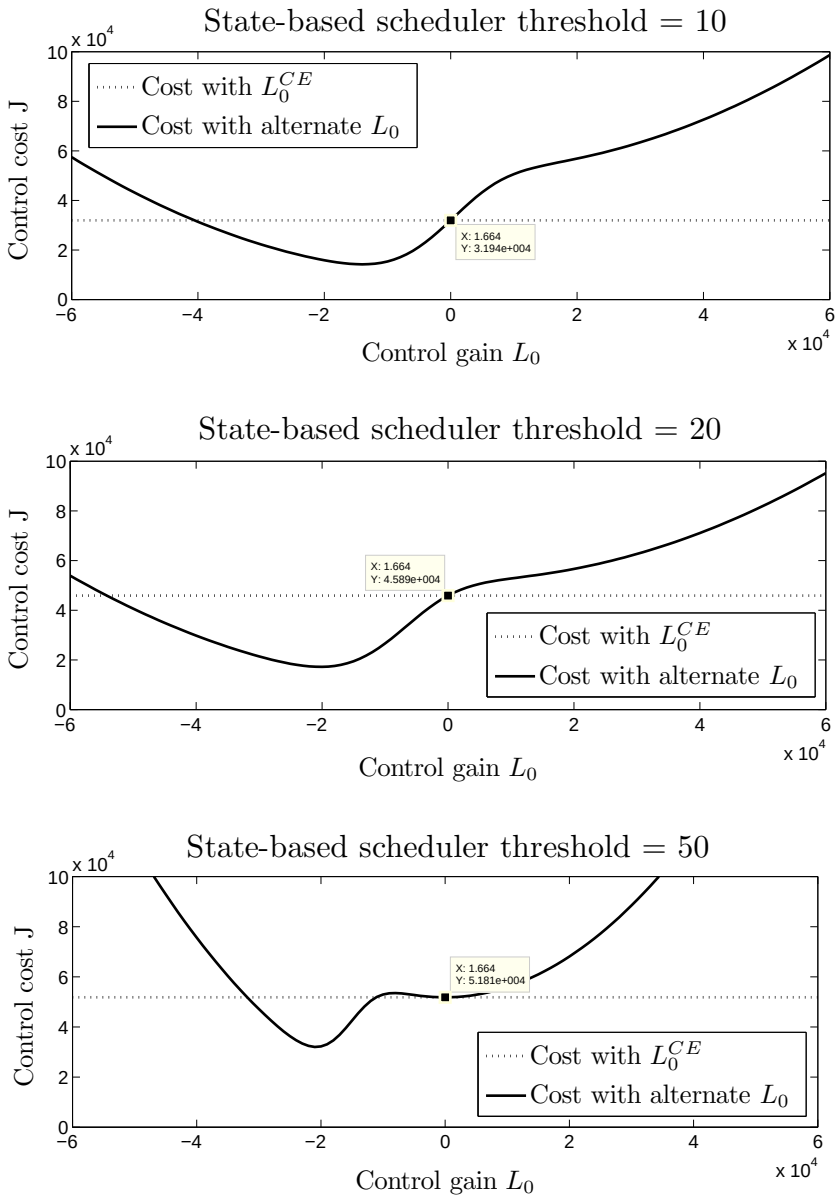


Figure 3.9: A comparison of the control costs obtained using the certainty equivalent controller (with gain $L_0^{\mathrm{CE}} = 1.664$, shown in dotted lines) and some alternative controllers (with gain L_0, shown in solid lines), for scheduler threshold $\epsilon = \{10, 20, 50\}$. Clearly, there are values of $L_0 \neq L_0^{\mathrm{CE}}$ that result in a lower cost. This can be explained by the dual effect in the control signal, as shown in Theorem 3.2.

exists a range of values of $L_0 \neq L_0^{\mathrm{CE}}$ for which the resulting control cost is lesser than the certainty equivalent control cost. This validates the discussion preceding the counterexample, and provides an example of controllers which utilize the probing incentive to improve upon the certainty equivalent controller. The improvement in cost reduces for larger thresholds, which might be explained by noting that the probing incentive, for a small threshold, is not accompanied by a high penalty in cost. As the threshold grows larger, the probing incentive might not be beneficial to exploit, in terms of the cost.

Another counterexample may be found in Curry (1970), who worked on the dual effect with nonlinear measurements. He examines a system with a non-linearity in the measurement, which may be interpreted as a simple state-based scheduling policy, and illustrates that the optimal controller for a two-step horizon cost differs from the certainty equivalent controller due to the dual effect.

This example shows that there is a dual effect, and even the same schedule can result in a different control sequence for a system without a dual effect. Thus, an equivalent construction for the scheduler does not result in an equivalent system.

3.6.3 An Example of the Dual Predictor Architecture

In this example, we present the dual predictor architecture applied to a shared network. We tune the threshold of the state-based scheduling law to probabilistically guarantee an achievable control performance, given the traffic over the network. We use a homogenous network in this example to simplify the comparison of control cost versus the scheduling threshold.

We consider a shared network of 20 scalar plants, indexed by $j \in \{1, \ldots, 20\}$ and given by (3.32), where $a^{(j)} = 1$ and $R_w^{(j)} = 1$ for all j. The plants are sampled with a period given by $T = 10$. The innovations-based scheduler uses a similar criterion to (3.23), where ϵ is the threshold of the scheduler. A p-persistent MAC, with synchronized slots, which permits three retransmissions, is used. The persistence probability is given by $p_\alpha^{(r)}$, where r denotes the retransmission index and $p_\alpha^{(r)} \in \{1, 0.75, 0.5\}$ for $r \in \{1, 2, 3\}$. The LQG criterion in (3.6), with $N = 10$ and $Q_0 = Q_1 = Q_2 = 1$ is used to design the optimal certainty equivalent controller (3.16). The observer calculates the MMSE estimate given by (3.24)-(3.25).

The effect of varying ϵ on the control cost is shown in Figure 3.10. For high values of ϵ, the network is under-utilized, and almost all the transmissions are successful. However, the control cost is high as the number of transmissions is low. As we decrease ϵ, the control cost initially decreases due to increased use of the network. However, for very low values of ϵ, the network is over-utilized and this results in collisions. Thus, the control cost increases again, due to dropped packets. It is interesting to note that the cost function is quite flat. Thus, it is not important, in practice, to use the optimal scheduling threshold ϵ.

Figure 3.11 depicts the state and control signal of the first plant obtained from our simulation, for the best value of ϵ picked from the above plot. Note that the estimation error is bounded, with a probability of 0.94, by the scheduling threshold,

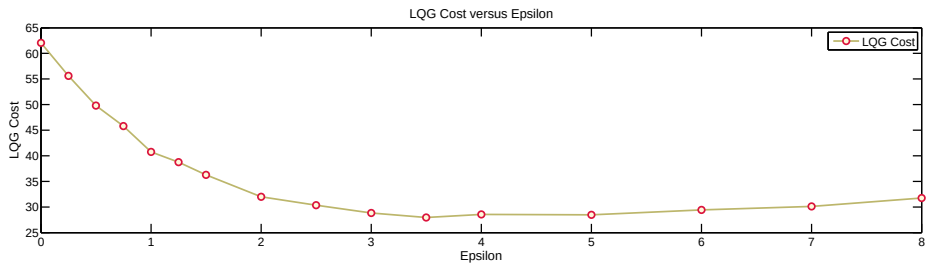


Figure 3.10: The control cost J_{DP} versus the scheduler threshold ϵ. For low thresholds, the high traffic in the network causes collisions, and a high J_{DP}. High values of ϵ result in an under-utilized network, and a high J_{DP} due to insufficient transmissions.

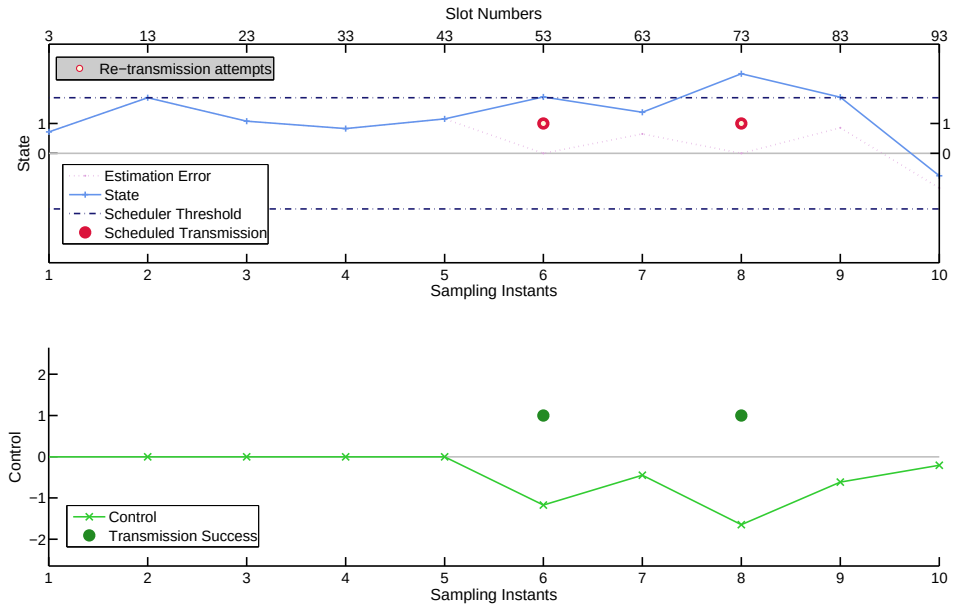


Figure 3.11: The estimation error, state and control signal with the channel use pattern. Note that the requested bound on the predicted estimation error, which is marked with a dotted line, is rarely exceeded. Also, the control signal corresponds closely to the state only when there is a successful transmission.

for the value $\epsilon^{(1)} = 3.5$, and the resulting control cost is $J_{\mathrm{DP}} = 27.9235$.

3.7 Summary

In this chapter, we examined the impact of a state-based scheduler on the design of a control system in a network. We found that a state- or measurement-based scheduler makes design of the optimal controller and observer hard, due to the presence of a dual effect. In general, certainty equivalence does not hold, unless the scheduler output is not influenced by the past applied control signals or policies. Furthermore, a scheduling policy which is symmetric in its arguments reduces the complexity of the estimator. We used these results to propose a dual predictor architecture for closed-loop systems with a state-based scheduler.

Now, we are ready to look at how a network of such control systems interact with each other. In the next chapter, we look at a network of systems with the dual predictor architecture, and analyze the performance of the channel access method.

Chapter 4

Modelling Network Interactions

We continue our analysis of a network of event-based systems. So far, we have identified the dual predictor architecture, which results in a separated design of the state-based scheduler, observer and controller, for each control system in the network. This architecture also gives us a precise form of the control law to be used. However, the event-triggering policy, or state-based scheduling policy, is only specified to be a symmetric function of the innovations. In this chapter, we choose a more specific template for this policy, and develop a method to analyze the performance of a network of systems using this policy.

The design of an event-triggering policy must be accompanied by the selection of a suitable multiple access protocol. In Example 1.1, we saw that the lack of a multiple access protocol can result in significant losses in a broadcast medium, even with very few interfering links. So, which multiple access protocol should we choose? In a time-triggered network, transmission requests can be anticipated a priori, and a schedule can be drawn up to accommodate all the transmissions. In contrast, transmission requests cannot be anticipated in an event-based network. Thus, the access decisions must be taken at each sensor node, in a distributed manner. Furthermore, coordinating access decisions between nodes is not easy to accomplish on wireless networks. Thus, we choose to use a random access protocol. These protocols use a contention resolution mechanism (CRM) to arbitrate access in a distributed, non-coordinated manner, between the nodes in the network, as discussed in Chapter 2. The carrier sense multiple access with collision avoidance (CSMA/CA) protocol is particularly well-suited to wireless networks and is used in Wifi (IEEE, 1999), Zigbee (ZigBee Alliance, 2005) and WirelessHart (HART Communication Foundation, 2007). In this chapter, we use the p-persistent CSMA protocol, which provides an analytical approximation for the CRM in CSMA/CA (Kleinrock and Tobagi, 1975).

So, we now consider a network of systems with the dual predictor architecture, which use an event-triggering policy along with a CRM to access the network. For such a network, we wish to understand how the network traffic affects the channel access probability of each event-based system. The channel access probability is

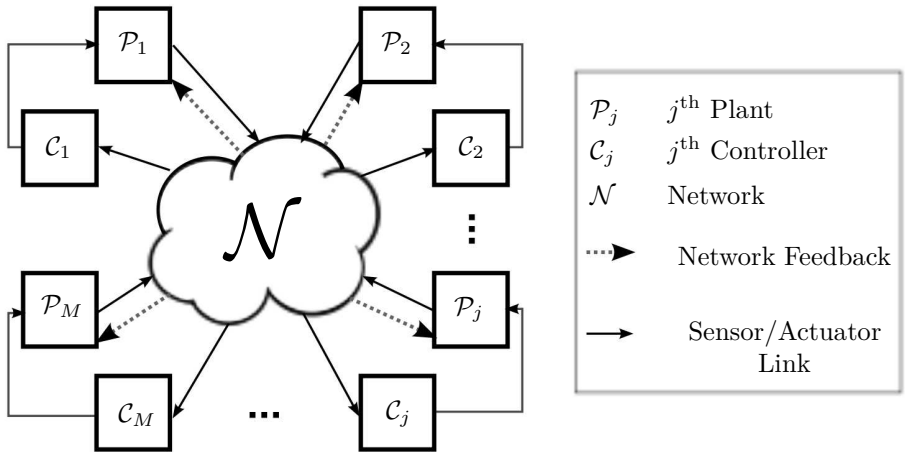


Figure 4.1: A network of M closed-loop systems, with each loop consisting of a plant $\mathcal{P}_j$ and a controller $\mathcal{C}_j$ for $j \in \{1, \ldots, M\}$. The systems share access to a common medium on the sensor link, and adapt their traffic rates to the feedback from the network. The controllers and actuators communicate over dedicated links.

determined by the event-triggering policy and the CRM. The network traffic is determined by the channel access policies of the other nodes in the network. Thus, we develop a model for the interactions in the multiple access channel comprising of event-based systems with CRMs. Using this model, we analyze the network performance of each event-triggering policy. This will, in turn, help us to design an event-triggering policy that adapts to the traffic in the network.

4.1 Contributions and Related Work

We begin by motivating our template for the event-triggering policy. We choose a simple policy that respects the constraints imposed by the dual predictor architecture; our policy applies a symmetric threshold on the estimation error at the scheduler. By varying the threshold with the delay since the last successful transmission, we can make the event-triggering policy adapt to the traffic in the network. This is a necessary property in an uncoordinated network, because packet collisions are an unavoidable consequence of distributed access decisions. Collisions result when two or more nodes transmit at the same time, and all the packets involved in a collision are lost. Naturally, collisions are detrimental to the performance of the control systems in the network. Adapting the event-triggering policy to the CRM response can reduce the number of collisions. However, it also makes the systems much harder to analyze. To see why, consider the systems illustrated in Figure 4.2. A typical network user generates traffic at a certain rate, and the network returns a probability of collision, which is a function of all the users' traffic rates. Thus, the

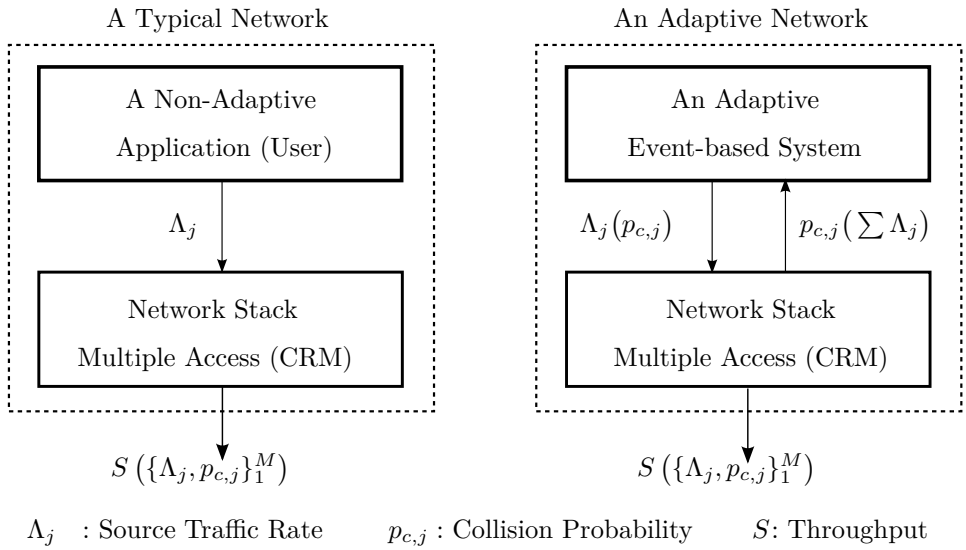


Figure 4.2: A typical network can be analyzed by evaluating the traffic rate of the user, and the collision probability of the multiple access protocol, in isolation. In an event-based network, the input traffic is adapted to the traffic in the network, i.e., Λ_j is a function of $p_{c,j}$. The event-triggering policy and multiple access protocol must be jointly analyzed in such a network.

user's rate and the performance of the multiple access protocol can be analyzed in isolation, in this case. In adaptive event-based networks, however, the traffic rate of each user is a function of the probability of collision of the network. Hence, a joint analysis of the event-triggering policy and the CRM is required.

Another consequence of random access is that network access for some nodes implies lack of access for other nodes in a resource-constrained network. Thus, the network access decisions are correlated, and for closed-loop systems, this correlation propagates to the system state. Closed-loop systems with exogenous noise processes become correlated due to their network interactions (Cervin and Henningsson, 2008; Rabi and Johansson, 2009a). Now, analyzing the resulting network is not a trivial task. To solve this problem, we derive inspiration from Bianchi's much-acclaimed analysis of the Distributed Coordination Function (Bianchi, 2000) in IEEE 802.11. To counter a similar problem of network-induced correlation between traffic sources, Bianchi assumes that a node that is ready to transmit, sees a busy channel as a time-averaged, independent process. The independence aspect of this assumption restores a renewal property in our setup, enabling the use of a Markov model to represent the interactions in an event-based network. The time-average assumption permits a performance analysis in steady state. We verify these assumptions through simulations.

There are two main contributions of this chapter. We present a joint analysis

of the event-triggering policy and the CRM. The analysis is made possible by the use of Bianchi's assumption. In doing so, we also present a new configuration for the applicability of Bianchi's assumption. Our final contribution is the resulting network model; a Markov chain which describes the event-triggering policy and the multiple access protocol. With this model, we can view the event-triggering policy as a set of steady state probabilities. This model facilitates the design of a set of probabilities that ensure a system-level guarantee. In the next chapter, we show that the model and analysis presented in this chapter can be used to design a network of event-based systems.

Various event-triggering policies have been proposed for different problem formulations (Rabi, 2006; Tabuada, 2007; Heemels et al., 2008; Henningsson et al., 2008). However, the multiple access problem for event-based systems has not received as much attention. Much of the work focussing on the design of event-based systems for a shared network (Wang and Lemmon, 2011; Molin and Hirche, 2012) does not explicitly deal with the problem of multiple access. Others use protocols such as the CAN bus for wired networks (Anta and Tabuada, 2009), or dynamic real-time scheduling for multiple tasks on a single processor (Tabuada, 2007). These protocols are not well-suited to wireless networks (Akyildiz et al., 1999; Gummalla and Limb, 2000).

There have been some attempts to analyze a network of event-based systems with random access, albeit with simplifying assumptions such as independent packet losses, or by ignoring collisions (Cervin and Henningsson, 2008; Rabi and Johansson, 2009a; Henningsson and Cervin, 2010). More recently, event-based systems which use Aloha and Slotted Aloha have been analyzed by Blind and Allgöwer (2011a), but with an event-triggering policy that is not adapted to the network. In this chapter, we use a Markov chain to model the interactions between the event-triggering policy and the CRM. A similar Markov chain has been used, but to model only the event-triggering policy, by Demirel et al. (2013) and Xia et al. (2013). The work presented in this chapter highlights the need for a joint analysis between the multiple access protocol and the event-triggering policy.

The rest of the chapter is organized as follows. We present the problem formulation in Section 4.2 and derive some important properties of the event-triggering policy, with no network traffic, in Section 4.3. We present the consequences of multiple access, and our solution using Bianchi's assumption in Section 4.4. The Markov model describing the joint interactions, and the corresponding performance analysis are presented in this section. Finally, we present some simulation results in Section 4.6 and validate the assumptions of our model.

4.2 Problem Formulation

We consider a network of M plants and controllers (indexed by $j \in \{1, \dots, M\}$), which communicate over a shared channel with an event-trigger in the loop, as shown in Figure 4.3. A model for the interactions between each event-based system

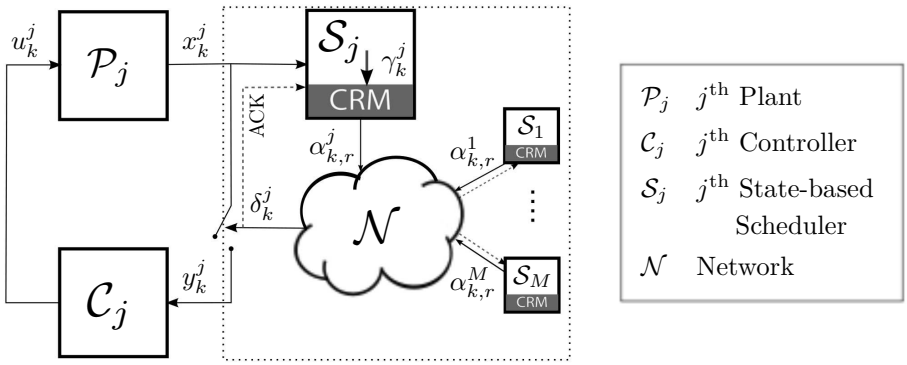


Figure 4.3: A network of M event-based systems using a CRM to access the network, where the j^{th} control loop is illustrated with its dependency on the other control loops $i \in \{1, \ldots, M\}$, $i \neq j$. The events are generated by the state-based schedulers. The other control loops are represented with their state-based schedulers alone.

and the network is depicted in Figure 4.4. The blocks in this figure are explained below.

Plant: The plant $\mathcal{P}_j$ has state dynamics given by

$$x_{k+1}^j = A_j x_k^j + B_j u_k^j + w_k^j \,, \tag{4.1}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$ and the initial state x_0^j and the process noise w_k^j are i.i.d. zero-mean Gaussians with covariance matrices R_0^j and R_w^j, respectively. They are independent and uncorrelated to each other and to the initial states and process noises of other plants in the network. This discrete time model is defined with respect to a sampling period T for each plant, and the sampling instants are generated by a synchronized network clock.

State-based Scheduler: There is a local scheduler $\mathcal{S}_j$, situated in the sensor node, between the plant and the controller, which decides if the state x_k^j is to be ignored or selected for transmission. The scheduler output γ_k^j is correspondingly chosen from the set $\{0, 1\}$, by the event-triggering policy f_k^j, implemented within this block. The policy used in our setup is presented below, but motivated in Section 4.3. The scheduler output is given by

$$\gamma_k^j = f_k^j(x_k^j - \hat{x}_{F,k}^j) = \begin{cases} 1 & |x_k^j - \hat{x}_{F,k}^j|^2 > \Delta_j(m_k^j) \,, \\ 0 & \text{otherwise} \,, \end{cases} \tag{4.2}$$

where, Δ_j is the threshold, which typically depends on the memory index of the event-triggering policy m_k^j. This index tracks the delay since the last received

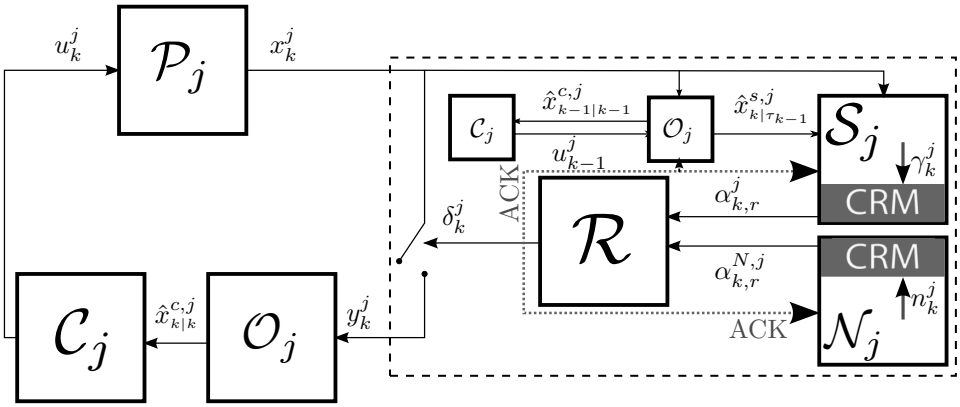


Figure 4.4: A model of the network in Figure 4.3, from the perspective of a single event-based system. The event-triggering policy uses the prediction error to determine when to transmit. An explicit acknowledgement (ACK) is needed to track the estimation error at the sensor node.

packet, d_{k-1}^j, for delays smaller than the maximum memory index F, i.e., $m_k^j = \min(d_{k-1}^j, F)$. In the above equation, $\hat{x}_{F,k}^j$ plays the role of a memory-limited predicted estimate at the sensor node (4.6).

Other Network Traffic: The block $\mathcal{N}$ models a fictionalized source, representing traffic from all other event-based systems in the network. This traffic is represented by the network traffic index $n_k^j \in \{0, 1\}$.

CRM: The multiple access protocol implements a CRM in each sensor node, which resolves contention between simultaneous channel access requests in a distributed manner. We consider the p-persistent CSMA mechanism, with R retransmissions. The retransmissions occur in the CRM time scale, which is much finer in resolution than the system time scale, as discussed in Chapter 2 and indicated in Figure 2.6. The time scales are assumed to be separated, i.e., all retransmissions corresponding to a single event are completed before the next sampling period. The access indicator for each retransmission $r = \{1, \ldots, R\}$ is denoted $\alpha_{k,r}^j \in \{0, 1\}$ at time k. The persistence probability in the r^{th} retransmission attempt is defined as $\mathrm{P}(\alpha_{k,r}^j = 1 | \gamma_k^j = 1)$, and denoted by $p_{\alpha,r}$ for brevity.

Channel Access Indicator: The channel access indicator $\delta_k^j \in \{0, 1\}$ denotes transmission failure or success, respectively, after R retransmission attempts. A transmission is successful if there is only one system that attempts to access the channel in that CRM slot, as given by

$$\delta_k^j = \bigvee_{r=1}^{R} \left[\alpha_{k,r}^j \cdot (1 - \alpha_{k,r}^{N,j}) \right] , \tag{4.3}$$

and then, $\sum_{j=1}^{M} \delta_k^j \leq R$. Thus, the number of retransmissions, R, also determines the maximum number of transmissions supported by the network protocol, for every system sampling instant.

Observer: The observer $\mathcal{O}_j$ receives y_k^j, given by

$$y_k^j = \begin{cases} x_k^j & \delta_k^j = 1 \ , \\ \varepsilon & \text{otherwise} \ , \end{cases} \tag{4.4}$$

where ε denotes a packet erasure when there is no event. The estimate is computed as

$$\hat{x}_{k|k}^{c,j} = \begin{cases} x_k^j & \delta_k^j = 1 \ , \\ A_j \hat{x}_{k-1|k-1}^{c,j} + B_j u_{k-1}^j & \text{otherwise} \ , \end{cases} \tag{4.5}$$

with $\hat{x}_{-1|-1}^{c,j} = 0$. We define the corresponding estimation error as $\tilde{x}_k^{c,j} = x_k^j - \hat{x}_{k|k}^{c,j}$. A copy of this observer is used at the sensor node to facilitate the event-triggering policy by generating the predicted estimate $\hat{x}_{k|\tau_{k-1}}^{s,j} = A_j^{(k-\tau_{k-1})} x_{\tau_{k-1}}^j + \sum_{l=\tau_{k-1}}^{k-1} A_j^{(k-l-1)} B_j u_l^j$. This is used to generate the memory-limited predicted estimate $\hat{x}_{F,k}^j$ in (4.2), which is given by

$$\hat{x}_{F,k}^j = \begin{cases} \hat{x}_{k|\tau_{k-1}}^{s,j} & d_{k-1}^j < F \ , \\ \hat{x}_{k|k-F}^{s,j} & \text{otherwise} \ . \end{cases} \tag{4.6}$$

Thus, $\hat{x}_{F,k}^j$ is given by the predicted estimate $\hat{x}_{k|\tau_{k-1}}^{s,j}$ when the delay is less than the memory F of the scheduler. When the delay exceeds this value, $\hat{x}_{F,k}^j$ takes the value $\hat{x}_{k|k-F}^{s,j}$, which is generated by assuming knowledge of x_{k-F}^j, in place of $x_{\tau_{k-1}}^j$.

Controller: The controller $\mathcal{C}_j$ generates an appropriate control signal, such as

$$u_k^j = -L_k^j \hat{x}_{k|k}^{c,j} \ , \tag{4.7}$$

where L_k^j is selected to minimize an appropriate cost function, such as the linear quadratic Gaussian (LQG) cost.

We are interested in analyzing the joint performance of the event-trigger and CRM in this network, in steady state. To do so, we define two metrics that characterize the network performance.

Definition 4.1 (Steady-state Delay Distribution)**.** *The delay since the last received packet is given by $d_k^j = k - \tau_k^j$, where τ_k^j is the time index of the last received packet, as illustrated in Figure 3.3. To avoid notational overhead, we skip the index j for τ_k and d_k, when the context is clear. The time index of the last received packet is defined as $\tau_k^j = \max\{t : \delta_t^j = 1\}$, for $-1 \leq t \leq k$ and $\delta_{-1}^j = 1$. Note that $-1 \leq \tau_k^j \leq k$. Then, the steady-state delay distribution is defined as $\mathrm{P}_d^j(\zeta) \triangleq \lim_{k\to\infty} \mathrm{P}(d_k^j = \zeta)$, for $\zeta \in \mathbb{Z}$.*

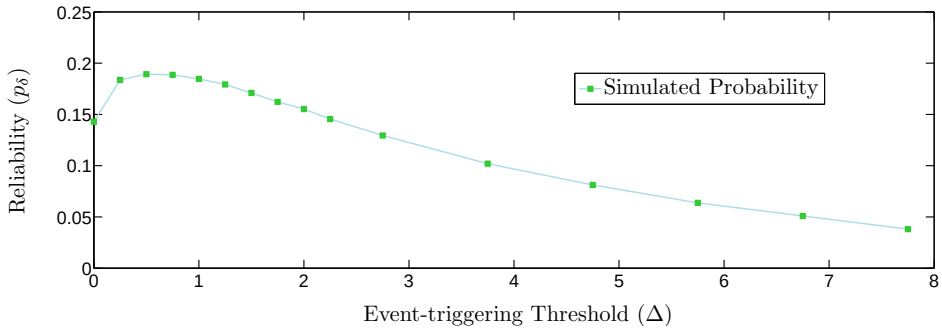


Figure 4.5: A plot of the simulated values of the reliability versus the scheduler threshold. This plot clearly shows that the above relationship cannot be approximated by an i.i.d. loss process or any other such simplistic modelling technique.

Definition 4.2 (Steady-state Reliability)**.** *Recall that δ_k^j is the channel access indicator. The steady state probability of a successful transmission as a consequence of the joint actions of the event-trigger and CRM is defined as $p_\delta^j \triangleq \lim_{k\to\infty} \mathrm{P}(\delta_k^j = 1)$. This indicates the network reliability on the sensing link for a given closed-loop system.*

The above information is a prerequisite for any design methodology that seeks to achieve a certain network or system guarantee.

4.2.1 Motivating Example

Before we delve into the main results, we present an example of a network of systems, and examine a performance analysis curve for this example obtained using Monte-Carlo simulations. With this example, we wish to motivate the methods used in the rest of this chapter.

Example 4.1

[Network and Experiment Setup] We consider a homogenous network of $M = 10$ nodes, with $R = 5$ retransmissions in the CRM. The dynamics of the plants are given by (4.1) for $x_k^j \in \mathbb{R}$ and $w_k^j \sim \mathcal{N}(0, 1)$. The plants are identical with state transition matrix $A_j = 1$ and control matrix $B_j = 1$, for $1 \leq j \leq 10$. We use a state-based scheduler (4.2) with the event-triggering policy $|x_k^j - \hat{x}_{F,k}^j|^2 > \Delta_j$, where Δ_j is a constant scheduler threshold that does not vary with the delay and $\hat{x}_{F,k}^j$ denotes the memory-limited predicted estimate (4.6) at time k. When the delay exceeds the memory of the policy F, the value x_{k-F}^j is assumed to be the last successfully received value while computing $\hat{x}_{F,k}^j$, thus limiting the memory of the event-triggering policy. To realize a scheduler such as this, we implement the dual predictor architecture presented in Figure 4.4. The CRM used to arbitrate access

is the p-persistent CSMA protocol, and $p_\alpha = 0.2$ for all 5 retransmission stages of the CRM.

A plot of the simulated values of reliability p_δ^j versus the scheduler threshold $\Delta_j \in (0, 8)$ is shown in Figure 4.5. This plot has been obtained using Monte-Carlo simulations. The nonlinear relationship depicted in the plot is not surprising, considering that a given scheduler threshold translates to a certain traffic rate depending on the distribution of the estimation error. However, it is important to note that the distribution of the estimation error with delay evolves based on the probability of a successful transmission, as a consequence of the adaptation illustrated in Figure 4.2. Thus, it is apparent from Figure 4.5 that there is no simple loss process that captures the interaction of a single system with the rest of the network.

We return to this example in Section 4.6, where we comment on the non-monotonic relationship obtained from simulations.

4.3 The Event-triggering Policy

We examine our event-triggering policy, to understand what it does for a single closed-loop system without other network traffic. We show that this policy adapts to the estimation error, or a part of it, when its memory is constrained. The renewal property of the estimation error is used to construct a Markov model that represents the functioning of the event-triggering policy. Since we only consider a single closed-loop system in this section, we drop the index j. The lack of other network traffic implies that $n_k \equiv 0$ for all $k \geq 0$. Now, there is no need for a multiple access protocol and every event results in a successful transmission, i.e., $\delta_k = \gamma_k$.

4.3.1 Properties of the Event-triggering Policy

We begin by motivating our selection of the event-triggering policy in (4.2). The events generated by our policy are not influenced by the past applied controls, resulting in a structural separation between the state-based scheduler, observer and controller, as shown in Chapter 3. Thus, the role of the controller is limited to regulating the estimate (4.5), and the role of the state-based scheduler is limited to reducing the estimation error. Accordingly, the policy defined in (4.2) adapts to the estimation error across the network; the input to this policy is the estimation error, for delays not exceeding F, or a related quantity, when the delay is F or larger. This can be seen from

$$|x_k - \hat{x}_{F,k}|^2 = \begin{cases} \left| x_k - \hat{x}^s_{k|\tau_{k-1}} \right|^2 & d_{k-1} < F \,, \\ \left| x_k - \hat{x}^s_{k|k-F} \right|^2 & \text{otherwise} \,. \end{cases}$$

By substituting for the prediction error in the above equation, we get

$$|x_k - \hat{x}_{F,k}|^2 = \begin{cases} \left|\sum_{l=\tau_{k-1}}^{k-1} A^{(k-l-1)} w_l\right|^2 & d_{k-1} < F \ , \\ \left|\sum_{l=k-F}^{k-1} A^{(k-l-1)} w_l\right|^2 & \text{otherwise} \ . \end{cases} \tag{4.8}$$

Note that the estimation error for $d_{k-1} \geq F$ is given by $\sum_{l=\tau_{k-1}}^{k-1} A^{(k-l-1)} w_l$. However, the value used in its place in the event-triggering policy is obtained by assuming that x_{k-F} was successfully transmitted, i.e., $\tau_{k-1} = k - F$. Thus, the statistical properties of the inputs to the above event-triggering policy vary with delay for $d_{k-1} < F$, but remain constant for $d_{k-1} \geq F$. Hence, we limit the memory of our adaptive policy to F.

Following a successful transmission, the estimation error is reset to zero at the observer, and this leads to some desirable properties for our policy, discussed below. For a sequence a_k, the notation $\boldsymbol{a}_{t_0}^{t_f}$ is used to denote the set $\{a_{t_0}, \ldots, a_{t_f}\}$.

Lemma 4.1. *For a single system given by (4.1)–(4.2), (4.5)–(4.7), with $\delta_k = \gamma_k$, $e_k = \tilde{\boldsymbol{x}}_{\tau_k}^k$ is a Markovian representation for the estimation error at the observer, $\tilde{x}_k$. In other words,*

$$\mathrm{P}(e_k | \boldsymbol{e}_0^{k-1}) = \mathrm{P}(e_k | e_{k-1}) \ . \tag{4.9}$$

Proof. At any time k, τ_k represents the time index corresponding to the last received packet. Then, $\tilde{x}_{\tau_k} = 0$, as the state x_{τ_k} is received by the observer. At time $\tau_k + 1$, the estimation error corresponds to the process noise w_{τ_k}. The process noise is i.i.d., and hence, independent of the state at τ_k or prior to it. This is also true for any future estimation error. Thus, we have

$$\mathrm{P}(\tilde{x}_k | \boldsymbol{y}_0^k, \boldsymbol{\delta}_0^k) = \mathrm{P}(\tilde{x}_k | \boldsymbol{y}_{\tau_k}^k, d_k) \ .$$

The delay since the last transmission, along with the measurement values since the last transmission form a sufficient statistic for the estimation error. Using this, and the relationship $\tau_k = \tau_{k-1}$ when $\delta_k = 0$, we obtain (4.9). □

Corollary 4.2. *For a single system given by (4.1)–(4.2), (4.5)–(4.7), with $\delta_k = \gamma_k$, the inter-arrival times at the controller are independent.*

Proof. The inter-arrival times are given by $t_{i+1} - t_i$, where $\{t : d_t = 0\}$ denotes the packet reception instants. Following the successful reception of a packet at time t_i, the future estimation error is independent of $\tilde{x}_{t_i | t_{i-1}}$. For the event-triggering policy in (4.2), the estimation error $\tilde{x}_{t_i+s|t_i}$, for $s > 1$, determines t_{i+1}. Thus, $t_{i+1} - t_i$ is independent of $t_i - t_{i-1}$. □

From Corollary 4.2, we can thus conclude that the event-triggering policy in (4.2) results in a traffic source that is a renewal process.

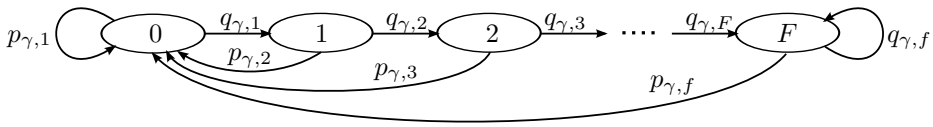


Figure 4.6: A Markov chain model representing the event-triggering policy in (4.2), when there is no exogenous network traffic. The estimation error grows with delay, resulting in different probabilities for events and non-events until F.

4.3.2 Markov Chain Representation

Using Lemma 4.1, we construct a Markov chain to represent the event-triggering policy, as shown in Figure 4.6. The state indices $m = \{0, \dots, F\}$ represent the memory of the event-triggering policy in (4.2). A return to the state $m = 0$ denotes a successful transmission, when the estimation error is reset to zero. From here on, the number of terms contributing to the input of the event-triggering policy continue to grow, as can be seen from the expression $\sum_{l=\tau_{k-1}}^{k-1} A^{(k-l-1)} w_l$. For $m < F$, we see two transitions out of every state; one to the next state $m+1$ indicating a non-transmission, and the other to 0 indicating a successful transmission. The corresponding probabilities are denoted $q_{\gamma,m+1}$ and $p_{\gamma,m+1}$, respectively, and defined as

$$p_{\gamma,m+1} = \mathrm{P}\left(\left| \sum_{l=k-(m+1)}^{k-1} A^{(k-l-1)} w_l \right|^2 > \Delta(m_k) \middle| d_{k-1} = m_k \right) , \quad m < F , \tag{4.10}$$
$$q_{\gamma,m+1} = 1 - p_{\gamma,m+1} .$$

For $m = F$, there are two transitions again, but a non-transmission returns to the same state, with probability $q_{\gamma,f} = 1 - p_{\gamma,f}$. The probability $p_{\gamma,f}$ is defined as in (4.10), with $m = F - 1$.

Event Probabilities: The probabilities of events and non-events in (4.10) can be computed given the event thresholds in (4.2), though this computation is not trivial as the estimation error does not have a Gaussian distribution. However, an event-triggered policy can be specified both in terms of event-thresholds or event-probabilities. For the rest of this chapter, we assume that the event-triggering policy in (4.2) is specified in terms of event probabilities, rather than event thresholds. The conversion from event probabilities to event thresholds becomes relevant when the event-triggering policy must be implemented, and we deal with that in Section 4.6.

Effect of independent Packet Losses: It is straightforward to extend the above model to include independent packet losses, which occur with probability p_L. A simple change of variables, with $p^L_{\gamma,m} = p_{\gamma,m} \cdot (1 - p_L)$ in place of $p_{\gamma,m}$ and $q^L_{\gamma,m} = 1 - p^L_{\gamma,m}$ in place of $q_{\gamma,m}$ gives us our modified Markov chain. This is

because the future estimation error and events remain independent of the past, after a transmission, and hence the statement of Lemma 4.1 continues to hold when there are packet losses.

4.4 The Multiple Access Event-triggered Problem

We now look at what happens when there are many event-based systems in the same network, i.e., $n_k \neq 0$. In this case, the CRM plays an important role as $\delta_k \neq \gamma_k$. We first examine the consequences of introducing the CRM in the network. Then, we use introduce Bianchi's assumption, and use this to construct a Markov model to represent the dynamics of the event-triggering policy and CRM.

4.4.1 Consequences of the CRM

The CRM impacts the network in two ways. The first consequence is that the event-triggered policy must be jointly analyzed with the CRM, as has already been illustrated in Figure 4.2. The other consequence is the correlation introduced between the various systems due to network interactions. We state and prove this below.

Lemma 4.3. *For the system described by (4.1)–(4.7), the estimation errors corresponding to different plants in the network are correlated, i.e.,*

$$\mathrm{P}(\tilde{x}_k^1, \ldots, \tilde{x}_k^M) \neq \prod_{j=1}^{M} \mathrm{P}(\tilde{x}_k^j) \,. \tag{4.11}$$

Proof. A network that supports R retransmissions must satisfy the constraint $\sum_{j=1}^{M} \delta_k^j \leq R$. Due to this, and the definition of δ_k in (4.3), the probability of a successful transmission depends on the probabilities of all the events in the network at time k. This can be expressed as

$$\mathrm{P}(\delta_k^j | \gamma_k^1, \ldots, \gamma_k^j = 1, \ldots, \gamma_k^M) \neq \mathrm{P}(\delta_k^j | \gamma_k^j = 1) \,. \tag{4.12}$$

The estimate (4.5) and the corresponding estimation error $\tilde{x}_k^j$ are determined by δ_k. Hence, the estimation error is correlated to all the events at time k. This is true for all the plants in the network, and thus, they become correlated to one another as indicated in (4.11). □

The above result reaffirms that the CRM introduces correlations between different event-based systems, as has been noted earlier by Cervin and Henningsson (2008) and also by Rabi and Johansson (2009a). The correlation between the estimation errors leads to correlation in the states, prediction errors and future scheduler outputs. An example of a scenario that might arise due to the above result is as follows; a large estimation error in a system that does not get to transmit,

perhaps due to random access or collisions, might result in increased congestion for the entire network due to persistent events from this system. This in turn might cause the estimation error to grow in other systems, and lead to further congestion. Hence, the properties in Lemma 4.1 and Corollary 4.2 do not hold for the systems in such a network, as formally proven below.

Lemma 4.4. *For the system described in (4.1)–(4.7), $e_k^j = \{\tilde{\boldsymbol{x}}^j\}_{\tau_k}^k$ is not a Markovian process. Consequently, the inter-arrival times at the observer are not independent.*

Proof. The Markovian properties of e_k^j in Lemma 4.1 followed from the independence of the estimation error, following a transmission, from its past. This is no longer true when there are interactions through the CRM. To see this, let us examine the prediction and estimation error following a transmission instant, τ_k^j, for the j^{th} plant and for some $k \geq 0$. The prediction error is given by $\tilde{x}^j_{\tau_k+1|\tau_k} = w^j_{\tau_k}$, and it is independent of the estimation error prior to τ_k^j due to the independence of the process noise $w^j_{\tau_k}$. Thus, we have

$$\mathrm{P}(\tilde{x}^j_{\tau_k+1|\tau_k} | \tilde{x}^j_{\tau_k|\tau_k}) = \mathrm{P}(\tilde{x}^j_{\tau_k+1|\tau_k}) \, .$$

Consequently, $\gamma^j_{\tau_k+1}$ is independent of all the other scheduler outputs. However, $\delta^j_{\tau_k+1}$ is still determined by all the scheduler outputs at time $\tau_k^j + 1$, as shown in (4.12). Thus, the estimation error $\tilde{x}^j_{\tau_k+1|\tau_k+1}$ is correlated with the estimation errors from all the other plants in the network, as shown in Lemma 4.3, some of which may be correlated with the estimation error of plant j prior to $\tau_k^j + 1$. Thus, the network interaction reintroduces a correlation with its past, and the estimation error following a reception instant is not independent of its past. In other words,

$$\mathrm{P}(\tilde{x}^j_{\tau_k+1|\tau_k+1} | \tilde{x}^j_{\tau_k|\tau_k}) \neq \mathrm{P}(\tilde{x}^j_{\tau_k+1|\tau_k+1}) \, .$$

Consequently, e_k is not Markovian. The lack of independence implies that arrival times are also correlated in this setup. □

A successful transmission for a node in a congested network need not reduce congestion for the other nodes in the network. The event-backlog may take a few sampling periods to dissipate. In the meanwhile, new events from the successful nodes will continue to see traffic conditions similar to those encountered by previous events from these nodes. Thus, the independence of the estimation error following a successful transmission is lost due to these network interactions. Analyzing the joint performance of the event-triggering policy and CRM is a challenging task due to the correlations in the network.

4.4.2 Bianchi's Assumption

We now use an assumption from the seminal paper by Bianchi (2000) that simplifies the network interactions. While presenting this assumption and utilizing it to construct a model, we consider the simplest setup in the multiple access network, which corresponds to the case when the CRM permits no retransmissions, i.e., $R = 1$. Accordingly, we denote the CRM access indicator $\alpha_{k,1}$ simply as α_k, with corresponding probability p_α. These results can be extended to include multiple retransmissions, which we discuss in Section 4.5.3.

Assumption 4.1. *For the systems described in (4.1)–(4.7), the conditional probability of a busy channel for a node that attempts to transmit in steady state, is given by an independent probability p for each node. Thus,*

$$\lim_{k\to\infty} \mathrm{P}(\delta_k^j = 0 | \gamma_k^j = 1, \alpha_k^j = 1) = p^j \ , \tag{4.13}$$

for all $j \in \{1, \dots, M\}$.

Simulations in Section 4.6 validate this assumption as a reasonable one to make for our problem setup. There are two aspects to this assumption; Firstly, (4.13) removes the correlation of the channel access indicator δ_k^j with the scheduler outputs of all the other plants in the network, which was shown in (4.12). Secondly, notice that p^j is not indexed by k; it is a time-average, and results in a steady state analysis, as we show in the rest of the chapter. Now, we use the independence aspect to extend the desirable properties of Lemma 4.1 and Corollary 4.2, to systems in the multiple access network.

Theorem 4.5. *For the systems described in (4.1)–(4.7), with Assumption 4.1,* $e_k^j = \{\tilde{\boldsymbol{x}}^j\}_{\tau_k}^k$ *is a Markovian representation for the steady state estimation error at the observer,* $\tilde{x}_k^j$*, for all* $j \in \{1, \dots, M\}$*. In other words,*

$$\lim_{k\to\infty} \mathrm{P}(e_k^j | \{\boldsymbol{e}^j\}_0^{k-1}) = \lim_{k\to\infty} \mathrm{P}(e_k^j | e_{k-1}^j) \ . \tag{4.14}$$

Consequently, the inter-arrival times at the observer for each plant is independent.

Proof. In Corollary 4.4, we showed that the estimation error following a packet reception instant $\tau_k^j + 1$, for some $j \in \{1, \dots, M\}$, is not independent of its past due to the correlation introduced by $\delta_{\tau_k+1}^j$. Re-examining (4.12), with Assumption 4.1, we now get

$$\mathrm{P}(\delta_k^j | \gamma_k^1, \dots, \gamma_k^j = 1, \dots, \gamma_k^M) = \sum_{\alpha_k^j \in \{0,1\}} \mathrm{P}(\delta_k^j | \gamma_k^j = 1, \alpha_k^j) \cdot \mathrm{P}(\alpha_k^j | \gamma_k^j = 1) \ ,$$

which implies that

$$\lim_{k\to\infty} \mathrm{P}(\delta_k^j = 0 | \gamma_k^1, \dots, \gamma_k^j = 1, \dots, \gamma_k^M) = p^j \cdot p_\alpha + 1 \cdot q_\alpha \ ,$$
$$\lim_{k\to\infty} \mathrm{P}(\delta_k^j = 1 | \gamma_k^1, \dots, \gamma_k^j = 1, \dots, \gamma_k^M) = q^j \cdot p_\alpha \ .$$

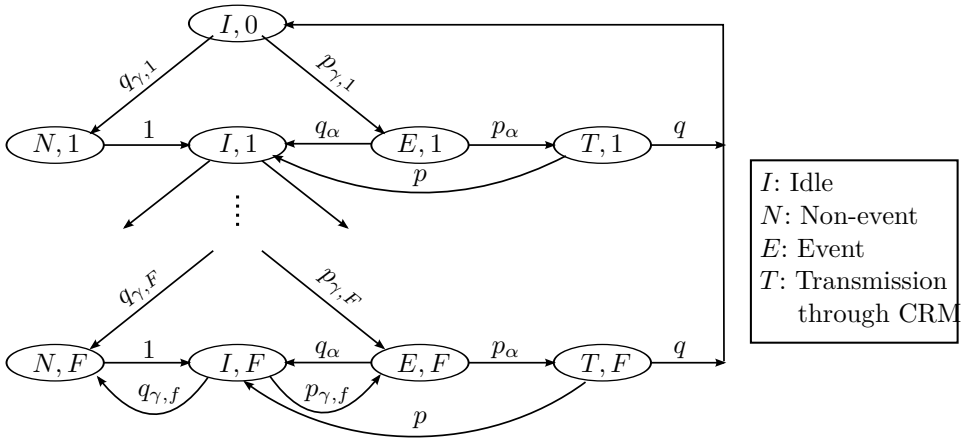


Figure 4.7: A Markov chain representation for the event-triggering policy in (4.2) and a simple CRM with no retransmissions. The variable F denotes the memory limit of the transmission history used by the scheduler.

Thus, the dependence on the other scheduler outputs vanishes due to Assumption 4.1. Now, the estimation error remains independent of its past, i.e.,

$$\lim_{k\to\infty} \mathrm{P}(\tilde{x}^j_{\tau_k+1|\tau_k+1}|\tilde{x}^j_{\tau_k|\tau_k}) = \lim_{k\to\infty} \mathrm{P}(\tilde{x}^j_{\tau_k+1|\tau_k+1}) \,.$$

Thus, using the same arguments as in the proof of Lemma 4.1, we can establish the Markovian property of e^j_k in (4.14). Consequently, the inter-arrival times are independent. □

Bianchi's assumption has converted the traffic source corresponding to the event-triggering policy and the CRM, into a renewal process. Now, analyzing the performance of this network is straightforward.

4.4.3 Markov Chain Representation

We use Bianchi's assumption to construct a Markov model of the event-triggering policy and CRM. The presentation in this section corresponds to a single system in the network, and thus, we skip the index j.

In the Markov chain in Figure 4.7, we assign two indices, (S, m), to each state and denote the probability of being in the state as $p_{(S,m)}$. The index m represents the steady state memory of the scheduler and is given by $\min(d_k, F)$. The states (S, m) and $(S, m+1)$ are one sampling period away from each other. The index S represents the four states a packet can be in during a sampling period. These are

1. $S = I$ [Idle State]: For $m > 0$, non-events and unsuccessful events return to this state before the next sampling instant. The initial state $(I, 0)$ indicates the idle state before the next sampling instant following a successful transmission.

2. $S = N$ [Non-event State]: This state is reached when the scheduler output $\gamma_k = 0$. A transition out of this state occurs instantaneously, and always to the idle state to wait for the next sampling instant.

3. $S = E$ [Event State]: This state is reached when $\gamma_k = 1$. A transition out of this state occurs to the transmission state or the idle state, depending on the CRM access indicator α_k. When $\alpha_k = 0$, the event is discarded and the system moves to the idle state to wait for the next sampling instant.

4. $S = T$ [Transmission in CRM State]: The CRM's inclusion can be seen directly in this state; it is reached only when the CRM permits channel access, or when $\alpha_k = 1$. Note that only a node in state T actually attempts a transmission. A transition out of this state occurs instantaneously, with two possibilities: transmission success or failure.

The transition probabilities in Figure 4.7 are explained below:

- $p_{\gamma,m}$ and $q_{\gamma,m}$ denote the probability of an event and non-event respectively, and are defined in (4.10).

- $p_{\gamma,f}$ and $q_{\gamma,f}$ denote the probability of an event and non-event, respectively, when $d_{k-1} \geq F$.

- p_α denotes the probability of accessing the channel through the CRM. Conversely, $q_\alpha = 1 - p_\alpha$ represents the probability of discarding an event.

- p denotes the conditional probability of a busy channel, as defined in (4.13). A successful transmission occurs with probability q.

Note that the Markov chain in Figure 4.7 represents the event-triggering and CRM of one system in the network. Thus, each system has its own such Markov chain, and these interact to produce the busy channel process in (4.13).

4.5 Steady State Performance Analysis

In this section, we use the time-averaged aspect of Bianchi's assumption along with the Markov chain in Figure 4.7 to derive a steady state analysis. We also present extensions to more advanced network settings.

4.5.1 Steady State Performance

Theorem 4.6. *For a system described by (4.1)–(4.7), with Assumption 4.1, the network reliability is given by*

$$p_{\delta}^{j} = (1 - p^{j}) \cdot p_{\text{tx}}^{j} \,, \tag{4.15}$$

where, p^{j} is the conditional probability of a busy channel for nodes attempting to transmit as defined in (4.13), and $p_{\text{tx}}^{j} = \sum_{m=1}^{F} p_{(T,m)}^{j}$ is the steady state probability that a node attempts to transmit, or is in any of the (T, m) states.

Proof. We begin by evaluating the probabilities $p_{(S,m)}$, in steady state, using the transition probabilities defined above. Then, we describe the interaction between the Markov models (Figure 4.7) corresponding to each of the systems in the network, to find an expression for the probability of a successful transmission.

The state (I, m), for $m > 0$, is always reached unless there is a successful transmission. The probability of a successful transmission in the m^{th} stage is given by $p_{\gamma,m} p_{\alpha} q$. Thus, we obtain the recursive expression

$$\begin{aligned} p_{(I,m)} &= (1 - p_{\gamma,m} p_{\alpha} q) p_{(I,m-1)} \,, \quad m = 1, \ldots, F-1 \,, \\ p_{(I,F)} &= \frac{1 - p_{\gamma,F} p_{\alpha} q}{p_{\gamma,f} p_{\alpha} q} p_{(I,F-1)} \,. \end{aligned} \tag{4.16}$$

In the final stage, (I, F) can be reached from state $(I, F-1)$ and from state (I, F) itself, which gives us the above equation. Also, at any sampling instant, a node must be in any of the (I, m) states. Thus, we have

$$\sum_{d=0}^{F} p_{(I,m)} = 1 \,. \tag{4.17}$$

The states (N, m) and (E, m) are reached by transitioning from state $(I, m-1)$ with probabilities $q_{\gamma,m}$ and $p_{\gamma,m}$, respectively. Thus, we have $p_{(N,m)} = q_{\gamma,m} p_{(I,m-1)}$ and $p_{(E,m)} = p_{\gamma,m} p_{(I,m-1)}$, respectively, for $m = 1, \ldots, F-1$. The final states (N, F) and (E, F) can be reached both from $(N, F-1)$, or $(E, F-1)$, and from (N, F), or (E, F), respectively. This gives us $p_{(N,F)} = q_{\gamma,F} p_{(I,F-1)} + q_{\gamma,f} p_{(I,F)}$ and $p_{(E,F)} = p_{\gamma,F} p_{(I,F-1)} + p_{\gamma,f} p_{(I,F)}$, respectively. The states (T, m), are reached only from the event states (E, m), and so we have $p_{(T,m)} = p_{\alpha} p_{(E,m)}$. Note that a node in any of the (T, m) states gets to transmit. The transmission probability of a node, denoted $p_{\text{tx}} = \sum_{m=1}^{F} p_{(T,m)}$. A busy channel results when more than one such node accesses the channel at the same time. For a network with M nodes, the j^{th} node's probability of a busy channel is

$$p^{j} = 1 - \prod_{i \neq j, i=1}^{M} (1 - p_{\text{tx}}^{i}) \,, \tag{4.18}$$

where p^i_{tx} is the transmission probability of any of the other $M-1$ nodes. Note that we use the independence aspect of Assumption 4.1 here, which simplifies the analysis.

For a network with M nodes, we have $2M$ equations (4.17) and (4.18) in $2M$ variables, $p^j_{(I,0)}$ and p^j for $j \in \{1,\dots,M\}$. These can be solved to find the corresponding steady state solution for each node in the network. Finally, a node that is successful in transmission, transitions to the state $(I,0)$. Thus, the probability of a successful transmission is given by $p_{(I,0)}$ in (4.15). □

The reliability is a joint measure of transmission obtained from the event-triggering policy and CRM. Other performance measures can also be found from the above Markov chain-based analysis. The steady state conditional probability of a successful transmission given that an event has occurred is given by $\lim_{k\to\infty} \mathrm{P}(\delta^j_k = 1|\gamma^j_k = 1) = p_\alpha q^j$, which does not depend on the memory of the scheduler in steady state. This quantity measures the contribution of the CRM and other network traffic towards congestion, or the lack of it. Similarly, we can evaluate the delay distribution for a node in this network, as we show below.

Corollary 4.7. *The delay distribution for a system described by (4.1)–(4.7), with Assumption 4.1 and $\zeta \in \mathbb{Z}$, is given by*

$$\mathrm{P}^j_d(\zeta) = \begin{cases} p^j_{(I,d_k)} p^j_{\gamma,d_k} p_\alpha q^j & d^j_k < F\,, \\ \hat{p}^j_{(I,d_k)} p^j_{\gamma,f} p_\alpha q^j & d^j_k \geq F\,, \end{cases} \tag{4.19}$$

where $\hat{p}^j_{(I,d_k)} = (1 - p^j_{\gamma,F} p_\alpha q^j) p_{(I,F-1)} + (1 - p^j_{\gamma,f} p_\alpha q^j)^{(d_k - F)} p_{(I,F)}$.

Proof. The probability of a delay $d < F$ is given by the probability of a successful transmission from the state (T,d) to the state $(I,0)$, in Figure 4.7. We use the same principle while computing the probability of a delay $d \geq F$. A delay of $d^j_k = F$ is incurred when a successful transmission from (T,F) is preceded by a transition from state $(I,F-1)$ to (I,F). A delay of $d^j_k > F$ is incurred when a successful transmission from (T,F) is preceded by $d^j_k - F$ transitions from state (I,F) to itself and the aforemention transition from state $(I,F-1)$ to (I,F). Using the expressions in (4.16), we obtain (4.19). □

Thus, the Markov model in Figure 4.7 helps us characterize the performance of the event-triggering policy and the CRM, for the entire network.

4.5.2 An Event-triggering Policy as a Set of Probabilities

In the Markov model presented in Figure 4.7, the probability of an event $p_{\gamma,m}$ varies with m. This is because the input arguments to the policy, defined in (4.8), and the event thresholds, Δ in (4.2), vary with m. Now, given that the event-triggering policy uses the estimation error as input, the set of thresholds $\{\Delta(0),\dots,\Delta(F)\}$,

for $0 \leq m \leq F$, represent the chosen policy. This set of thresholds can be translated into the corresponding set of probabilities $\{p_{\gamma,1}, \ldots, p_{\gamma,F}, p_{\gamma,f}\}$ using (4.10). Thus, the set of probabilities are an alternative representation of the chosen policy. In fact, the set of probabilities can represent any given event-triggering policy. Furthermore, this set of probabilities determines the performance of the event-based network, as we saw in Theorem 4.6 and Corollary 4.7. Thus, we consider this set of probabilities as the specification of our event-triggering policy.

To implement a given event-triggering policy, a set of thresholds corresponding to the specified set of probabilities must be found. This is not a trivial task, as the prediction errors are not Gaussian. Furthermore, its probability densities are determined by the conditional probability of a busy channel p. However, it is worth noting that finding the set of probabilities corresponding to a set of thresholds is equally hard, as the underlying density functions need to be evaluated either way. It is easier to accomplish a translation from one representation to the other numerically, as we show in Section 4.6.

4.5.3 Extensions to More General Network Settings

We now extend the Markov model presented in Figure 4.7 to include more general network settings, such as retransmissions in the CRM and asynchrony.

CSMA with retransmissions

A realistic CRM is likely to use retransmissions to spread congested network traffic over the sampling interval, as described in Figure 2.6. The model corresponding to such a CRM requires a Markov chain of its own, as shown in Figure 4.8. Here, the event and CRM states, (E, m) and (T, m) for each m, are replaced by multiple states, (E, m, r) and (T, m, r), for $r = 1, \ldots, R$ retransmission attempts. Each successive retransmission attempt sees the same or lesser traffic from all the nodes in the network, due to a strictly non-negative probability of successful transmission in the previous attempt. Thus, the resulting Markov chain must have a unique conditional probability of a busy channel, p_r, for each retransmission attempt $1 \leq r \leq R$, analogous to p in Assumption 4.1.

Assumption 4.2. *For the systems described in (4.1)–(4.7), the conditional probability of a busy channel for a node that attempts to transmit is given by an independent probability p_r for each retransmission stage and each system. Thus,*

$$\lim_{k\to\infty} \mathrm{P}(\delta_k^j = 0 | \gamma_k^j = 1, \alpha_{k,r}^j = 1) = p_r^j \,, \tag{4.20}$$

for all $j \in \{1, \ldots, M\}$ and all $r \in \{1, \ldots, R\}$.

Generating a complete Markov chain for $m = 0, \ldots, F$, using the states shown in Figure 4.8, we can re-evaluate all the probabilities in the proof of Theorem 4.6. Note that only some of the terms change. The probability of an unsuccessful transmission

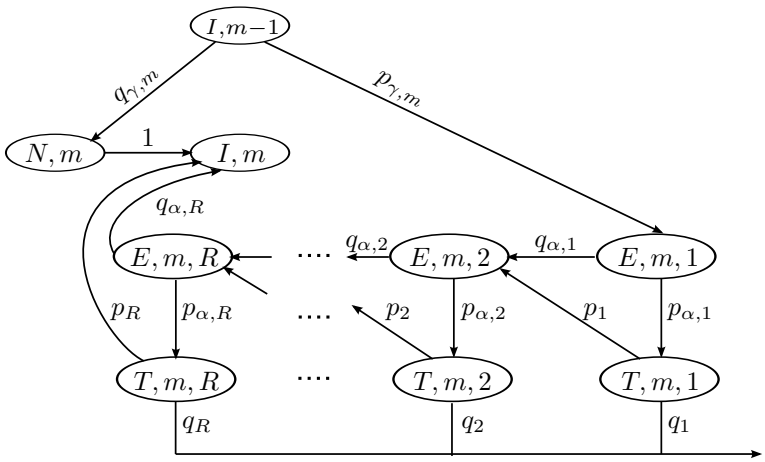


Figure 4.8: Embedding a CRM with R distinct re-transmission stages in the Markov chain model

in the m^{th} stage is now given by $\prod_{r=1}^{R}(p_{\alpha,r}p_r+q_{\alpha,r})p_{\gamma,m}$, as follows from Figure 4.8. This gives us

$$p_{(I,m)} = \prod_{r=1}^{R}(p_{\alpha,r}p_r + q_{\alpha,r})p_{\gamma,m}p_{(I,m-1)} \ , \quad m = 1,\ldots,F-1 \ ,$$

$$p_{(I,F)} = \frac{\prod_{r=1}^{R}(p_{\alpha,r}p_r + q_{\alpha,r})p_{\gamma,F}}{1-\prod_{r=1}^{R}(p_{\alpha,r}p_r + q_{\alpha,r})p_{\gamma,f}}p_{(I,F-1)} \ .$$

The probability of the states (T,m,r) is given by $p_{(T,m,r)} = (\prod_{q=1}^{r-1}(p_{\alpha,q}p_q + q_{\alpha,q}))$ $\cdot\, p_{\alpha,r}p_{\gamma,m}p_{(I,m-1)}$, and the corresponding probability of transmission from any of the (T,m,r) states, for different values of r, is given by $p_{\text{tx},r} = \sum_{m=1}^{F} p_{(T,m,r)}$. Now, the conditional probability of a busy channel in the r^{th} retransmission stage can be derived as

$$p_r^j = 1 - \prod_{i\neq j, i=1}^{M}(1-p_{\text{tx},r}^i) \ , \text{for } r \in \{1,\ldots,R\} \ . \tag{4.21}$$

Using the above equations in place of (4.16) and (4.18), we obtain similar expressions for the reliability and delay distribution as in Theorem 4.6 and Corollary 4.7, respectively.

We present simulations to validate Assumption 4.2 in Section 4.6.

Asynchronous networks

Consider an asynchronous network, with the CRM operating in a beacon-enabled mode. In this mode, the CRM slots remain synchronized across the network, but

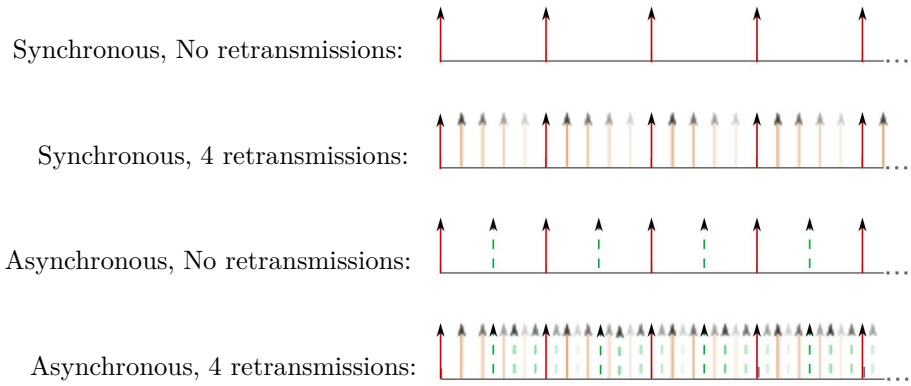


Figure 4.9: A comparison between synchronous and asynchronous traffic, with and without retransmissions in the CRM. The steady state analysis differs for each traffic pattern, as nodes in the (T, m) or (T, m, r) states see different traffic patterns under each configuration.

different systems can choose to initiate sampling at randomly selected CRM slots. Consecutive samples are spaced by the sampling period T CRM slots, for all the systems in the network. An illustration of the behaviour, with and without retransmissions in the CRM, for synchronous and asynchronous networks, is provided in Figure 4.9. For an asynchronous network with no retransmissions in the CRM, the number of interfering transmissions in the (T, m) states is given by $M_j < M$, where M_j is the number of nodes whose sampling instants lie in the same CRM slot of the j^{th} node. Thus, the performance of the network depends on the initial sampling slots chosen by the nodes. The more spread apart they are, the better the performance. For an asynchronous network with retransmissions in the CRM, the steady state seen by each retransmission state (T, m, r) can only be determined by knowing which of the retransmission states of other nodes interferes with transmissions from the r^{th} stage. Thus, to predict the performance of such a network, one must know the initial sampling slots chosen by all the nodes in the network.

If we assume that the initial sampling slots are chosen uniformly across a frame, we can predict the average performance of an asynchronous network. We can compute the conditional probability of a busy channel by averaging across all possible combinations of interactions in each retransmission state. This averaging makes a node in the state (T, m, r) see a busy channel due to other nodes in any of their (T, m, r) states, for all m and r. Thus, the probability of a busy channel is uniform across all retransmission states, i.e., $p_r^j = p^j$ for $r = 1, \ldots, R$. The modified version of Assumption 4.1 is stated below.

Assumption 4.3. *For the systems described in (4.1)–(4.7) in an asynchronous network, the conditional probability of a busy channel for a node that attempts to*

transmit, is given by an independent probability p for all retransmission stages, for each system. Thus,

$$\lim_{k\to\infty} \mathrm{P}(\delta_k^j = 0 | \gamma_k^j = 1, \alpha_{k,r}^j = 1) = p^j \ , \tag{4.22}$$

for all $j \in \{1, \ldots, M\}$ *and all* $r \in \{1, \ldots, R\}$.

To evaluate the probability in (4.22), we average across all the competing transmissions during the slots corresponding to state (T, m, r) of the j^{th} node, for some m and r. There are R^{M-1} different combinations of interactions between the R retransmission stages of the other $M-1$ nodes in the network, due to different initial sampling slots. If each of these interactions are equally likely, the resulting expression is quite simple. The conditional probability can be found to be

$$p^j = 1 - \prod_{i\neq j, i=1}^{M} (1 - q_{\text{tx}}^i) \ , \tag{4.23}$$

where $q_{\text{tx}}^i = (1/R) \cdot \sum_{r=1}^{R} q_{\text{tx},r}^i$ denotes the average transmission probability across all retransmission states. This equation can be used in place of (4.18) to find expressions for the reliability and delay distribution as before. We perform simulations in Section 4.6 to validate Assumption 4.3 and the resulting analysis. However, note that to obtain this result, we simulate across R^{M-1} different combinations of interactions, due to R^{M-1} different combinations of initial sampling slots. The result obtained for a single selection of sampling slots can be quite different from the averaged values.

4.6 Examples and Simulations

We now return to Example 4.1, and apply our analysis to this experimental setup. We present a number of variations of this example to validate each of the assumptions we have used for analyzing different network configurations. In each case, we present the reliability obtained through Monte-Carlo simulations, and compare it to the analytical value obtained using the analysis presented above. The differences are negligible in each case, thus validating our assumptions and verifying our analysis. We also evaluate the LQG control cost, defined as

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \mathbb{E}\left[x_n^\top Q_1 x_n + u_n^\top Q_2 u_n\right] \ ,$$

where Q_1 and Q_2 are the state and control weighting matrices, respectively. We use the LQG cost as a control-theoretic performance metric for a given event-triggering policy.

Table 4.10: A comparison of analytical and simulated values of p

Parameter	Simulation	Analysis
p_δ	0.1840	0.1872
p_1	0.5937	0.5944
p_2	0.5655	0.5620
p_3	0.5367	0.5277
p_4	0.5076	0.4917
p_5	0.4778	0.4542

Example 4.2
[Event-Triggering Policy as a Set of Probabilities] We use the same setup described in Example 4.1, comprising of a homogenous network of $M = 10$ scalar systems, with $R = 5$ retransmissions. The event-triggering policies are described by (4.2), with constant thresholds. For a chosen set of event probabilities, we discuss the implementation of the event-triggering policy. We also compare the results of Monte-Carlo simulations with results obtained from our analysis.

The event probabilities are given to be $p_{\gamma,m} = \begin{bmatrix} 0.3171 & 0.5138 \end{bmatrix}$ for $m = 1, \ldots, M$. Computing thresholds from a set of event probabilities is not easy, as the estimation error distributions are not Gaussian. In fact, a closed-form expression cannot be found for the distribution, though the evolution of the distribution can be described iteratively. Thus, we empirically select thresholds which result in the desired probabilities. In fact, $\Delta = 1$ achieves the given probabilities.

The values of reliability and the conditional probabilities of a busy channel obtained through simulations and analysis are presented in Table 4.10. The simulated values agree closely with the analytical values computed using Theorem 4.6. Thus, Assumption 4.2 is a reasonable approximation and motivates the Markov modelling.

Example 4.3
[Simple Network with No Retransmissions] We now consider a setup consisting of $M = 2$ nodes. There are no retransmissions in this network, i.e., $R = 1$, and the CRM probability is $p_\alpha = 0.5$. The plant model and event-triggering policy are identical to the ones in Example 4.1. The scheduler threshold is varied from $\Delta = 0$ to $\Delta = 8$ in this experiment. Each value of Δ results in a set of event probabilities and a corresponding network performance. The reliability p_δ obtained through analysis and simulations is plotted against the threshold in Figure 4.11. As the threshold increases, the reliability decreases. Thus, at larger thresholds, too few events are

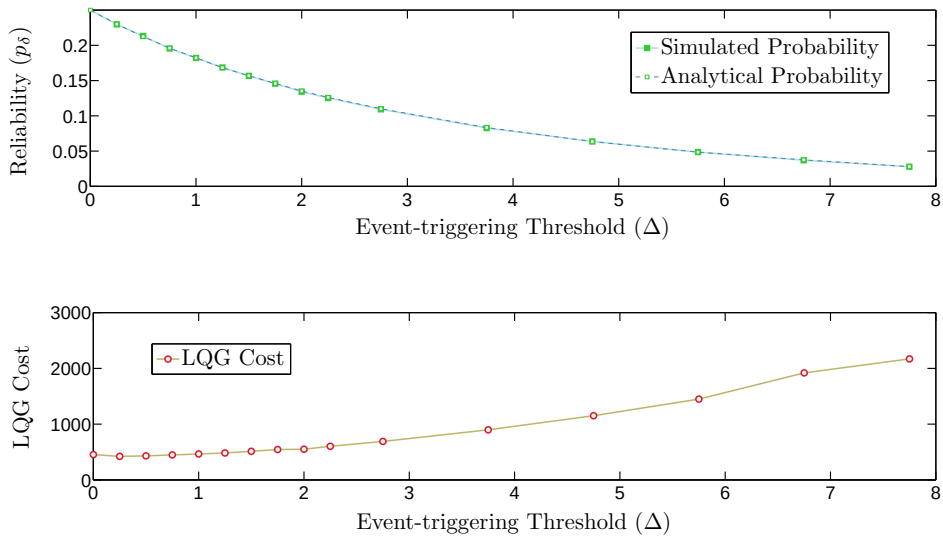


Figure 4.11: A comparison of the analytical and simulated values of the reliability versus the event-triggering threshold, in a simple network with no retransmissions in the CRM. The close correspondence of these values validates Assumption 4.1 and the results of Theorem 4.6.

being generated to completely utilize the network resources. Note the close correspondence between the simulated and analytical values, validating Assumption 4.1. The corresponding control costs obtained through simulations are also plotted in the graph below, indicating an expected increase in cost with decreasing reliability.

Example 4.4

[Retransmissions in the CRM] We return to the problem setup in Example 4.1, with $M = 10$ nodes and $R = 5$ retransmissions. A comparison of analytical and simulated values of the reliability versus the threshold for this synchronized network is shown in Figure 4.12. The performance obtained from the network is, in accordance with expectations, poor due to synchronization and congestion. Low thresholds cause many packets to flood the network, and result in a low probability of a successful transmission due to congestion. High thresholds reduce the utilization of the network, and the probability of a successful transmission decreases again. Note that there is a threshold that optimizes use of the network resources. A system-level performance analysis is required to characterize this threshold.

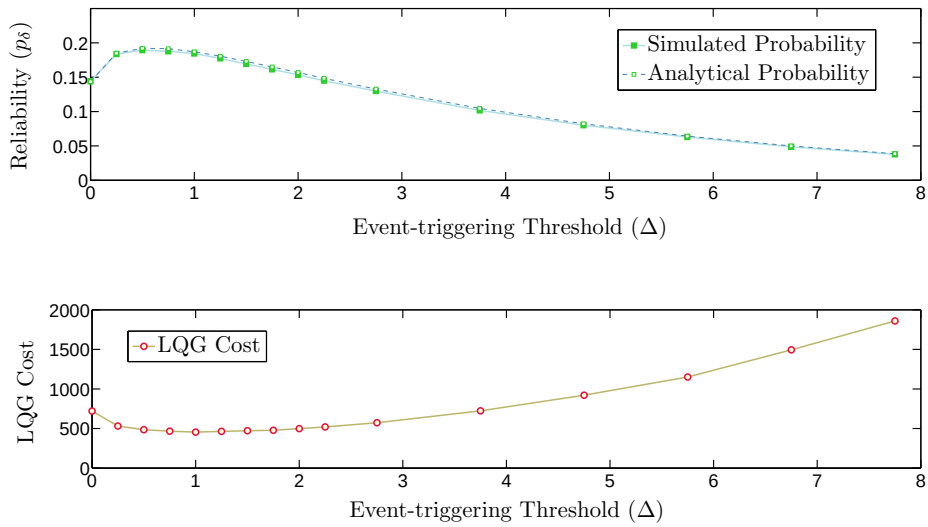


Figure 4.12: A comparison of the analytical and simulated values of the reliability versus the scheduler threshold, with retransmissions in the CRM. This example validates Assumption 4.2. Note that low thresholds result in a low $\mathrm{P}(\delta_k = 1)$ due to congestion. High thresholds also result in a low $\mathrm{P}(\delta_k = 1)$, but due to under-utilization of the network.

Example 4.5

[Unsaturated Traffic] In this example, we look at sparse traffic and show that Bianchi's assumption holds well even in this scenario. We have now validated Bianchi's assumption in two different scenarios, with and without retransmissions in the CRM. However, Bianchi's assumption is theoretically motivated by a mean field analysis. Thus, it is important to ascertain that this assumption holds just as well when there is not much traffic in the network. So, consider a network with $M = 2$ nodes and $R = 5$ retransmissions in the CRM. The sampling period corresponds to $T = 5$ CRM slots. Thus, each of the nodes has sufficient slots to transmit successfully. The plant model and event-triggering policy are the same as in Example 4.1.

A comparison of the reliability obtained for different thresholds is shown in Figure 4.13. Note that the maximum reliability obtained in this network is for the lowest value of the threshold, i.e., $\Delta = 0$. In other words, all samples are chosen as events, and even so, the network is largely successful in delivering them to the respective controllers. Also note that the reliability falls sharply as the threshold increases, indicating that too few events are generated to fully utilize the available

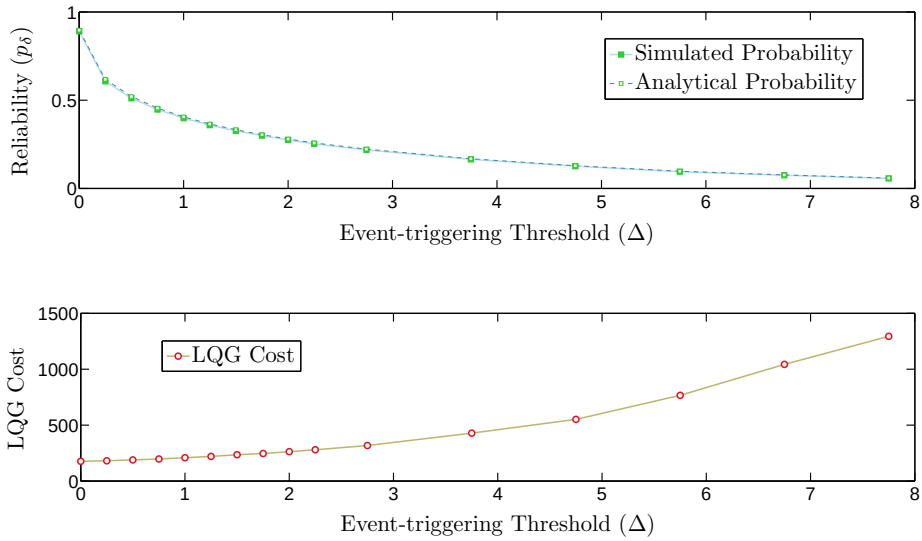


Figure 4.13: A comparison of the analytical and simulated values of the reliability versus the scheduler threshold, with unsaturated traffic. There are just 2 nodes in the network and the CRM permits 5 retransmissions. Even so, Bianchi's assumption seems to hold, indicating that this is a good approximation of unsaturated and saturated network conditions.

network resources.

Example 4.6

[Asynchronous Traffic] We now look at an asynchronous network, with $M = 5$ nodes, a sampling period of $T = 3$ slots and $R = 2$ retransmissions in the CRM. The plant model and event-triggering policy are identical to the setup in Example 4.1. The persistence probabilities of the CRM are chosen to be $p_{\alpha,1} = p_{\alpha,2} = 0.4$. A comparison of the reliabilities obtained for various thresholds is shown in Figure 4.14. The analytical and simulated values bear close correspondence, thus validating Assumption 4.3. Note that the values obtained in this experiment are averaged across all possible selections of initial sampling slots by all five nodes in the network. The results may be quite different for a given selection of initial sampling slots. In other words, the highest reliability obtainable from this system may far exceed the average reliability shown in Figure 4.14.

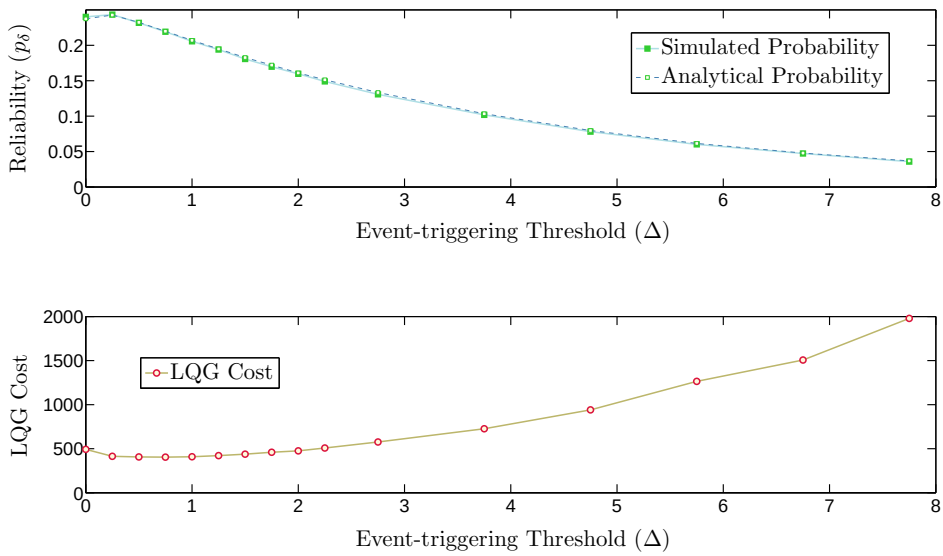


Figure 4.14: A comparison of the analytical and simulated values of the reliability versus the event-triggering threshold, for an asynchronous network. The network consists of 5 nodes, with sampling periods of 3 slots and 2 retransmissions in the CRM. The reliability obtained through analysis and simulations is averaged across all possible selections of initial sampling slots by the 5 nodes in the network. This plot validates Assumption 4.3.

4.6.1 Discussion

Let us now compare the results we have obtained in the above examples, to comment on the underlying network configurations. Examples 4.3 and 4.5 deal with networks consisting of two closed-loop systems each, but permitting one and five re-transmissions, respectively. The higher number of retransmissions results in a significantly higher reliability, and correspondingly lower LQG cost. This can be seen by comparing Figures 4.11 and 4.13. Example 4.4 deals with a network consisting of ten synchronized closed-loop systems and five retransmissions. The ratio of transmissions slots to number of systems is equal to 0.5, which is the same as for Example 4.3. A comparison of Figures 4.12 and 4.11 indicates a slightly reduced reliability in the multiple retransmission case, especially for small values of the event-triggering threshold Δ. This can be attributed to the increase in congestion at every sampling instant in a synchronized network with more systems. The reliability curve for the asynchronous network in Figure 4.14 improves the performance for small values of Δ.

We now comment on Bianchi's assumption, which has been shown to hold under different network configurations. The above results validate the use of Bianchi's

assumption for modelling the interactions of event-triggering policies and CRMs. Previously, Bianchi's assumption has been shown to hold in setups where the probability of accessing the network in different stages results from independent random processes, such as random backoffs in CSMA/CA. The probability of accessing the network in our model is not independent in each stage, as the estimation error for event-triggering policies is correlated to its past, as shown in Lemma 4.4. Thus, what we have here is a new configuration for the applicability of Bianchi's assumption. A theoretical motivation of this assumption is beyond the scope of this chapter. An explanation of Bianchi's assumption in the context of CSMA/CA is presented by Bordenave et al. (2010).

4.7 Summary

In this chapter, we have presented a method to analyze the performance of a network of event-based systems that use a CRM to access the shared network. We have shown that a Markov model can be constructed to represent the event-triggering policy and CRM, once we use Bianchi's assumption. Based on this model, we have analyzed the steady state performance of the resulting network. This analysis assumed conditional independence from other traffic when a node attempts to transmit. We validated this assumption through simulations, and provided extensions to more complex network configurations.

We now have a model for the interference on a multiple access channel with state-based channel access policies. Bianchi's conditional independence assumption provides us a conceptually simple model for this interference, with one caveat. The busy channel probability is a nonlinear function of the parameters of the channel access policy itself. Despite this, the model we have presented in this chapter can be used to analyze the behaviour and control performance of an event-based system in this network. It can also be used to design an event-based system, as we show in the next chapter.

Chapter 5

Stability Analysis and Design

We are now ready to design a network of event-based systems. Our design methodology develops on the results we have achieved so far: We know how to choose event-triggering and control policies to obtain separation in the feedback loop, using the dual predictor architecture. Furthermore, we know how to model the network interactions between multiple event-based systems, each of which use the dual predictor architecture along with a contention resolution mechanism (CRM). In other words, we have a model for the interference in a multiple access channel from a network of event-based systems. We now use this interference model to analyze the stability of each event-based system. Then, we use the results of this analysis to design a network of event-based systems that guarantee stability.

In the previous chapter, we considered event-triggering policies that detect a level crossing in the plant output. The thresholds in our event-triggering policy were allowed to vary over time, to allow the event-triggering policy to adapt to the traffic in the network. However, we did not specify how these thresholds or triggering levels were to vary with the delay since the last successful transmission. If the physical medium causes packet losses, then altering the triggering level to permit more frequent transmissions improves the packet reception rate (Rabi and Johansson, 2009a). However, this strategy may not work when packets are lost due to collisions. This is because increasing the number of events may increase the number of collisions. Thus, the triggering levels may have to be altered to reduce the number of transmissions, so as to alleviate congestion in the network. This is the same principle used in congestion control in TCP/IP or in the backoff mechanism in carrier sense multiple access (CSMA) protocols. However, will such a policy lead to stability of the networked control system (NCS)? In other words, how should the levels be selected, to ensure stability of the network *and* stability of the control system? Answering this question is one of the main objectives of this chapter.

5.1 Contributions and Related Work

There are two main contributions in this chapter. Our first contribution is to identify stability conditions for a network of event-based systems. To analyze stability of this network, we use the network-interaction model proposed in Chapter 4. Here, Bianchi's assumption (Bianchi, 2000) is used to decouple interaction between the various loops, resulting in a steady state Markov model. A statistical description of the system evolution through the states of the Markov chain is not analytically tractable, and hence, we identify an upper bound to describe the system using majorization theory. We obtain sufficient conditions for Lyapunov mean square stability by analyzing the resulting upper bound, and find that this notion of stability is achievable, if the probability of increasing delay is suitably restricted.

Our second contribution is to use the above stability analysis to design event-triggering policies that guarantee stability. We introduce a constant-law policy, where the event probabilities are mandated to remain constant, with increasing delay. We derive conditions for Lyapunov mean square stability for this policy, and present a design algorithm that guarantees it for a network of control systems using this policy. Hence, this chapter delivers an explicit policy that guarantees network and closed-loop stability under suitable assumptions.

The problem of level selection after a packet loss was introduced by Rabi and Johansson (2009a), where the authors evaluated the control cost of level triggering subject to i.i.d. packet losses. Stochastic stability of event-based systems with i.i.d. intervals between arrivals has been studied by Antunes et al. (2011) and Lemmon and Hu (2011). However, event arrivals in a contention-based network are not i.i.d, and the event arrivals considered in this chapter exhibit a dependence on the delay since the last transmission. The notion of stability that we use in this analysis has been used by Gupta and Martins (2010), to analyze i.i.d. erasures, with a provision to extend to Markov models, in NCSs.

The rest of this chapter is organized as follows. The problem formulation, along with a Markov chain representation, is presented in Section 5.2. The main results on sufficient conditions for Lyapunov mean square stability are presented in Section 5.3, and three design laws are presented in Section 5.4. Some examples follow in Section 5.5.

5.2 Problem Formulation

We consider a network of M event-based systems, shown in Figure 4.3. We first describe a model for each event-based system in the network, indexed by $j \in \{1, \ldots, M\}$, and then present a model for the interaction of the M systems.

5.2.1 Closed-loop System Model

The network on the sensor link can be modelled from the perspective of a single control system, as illustrated in Figure 5.1. We describe each block in this model

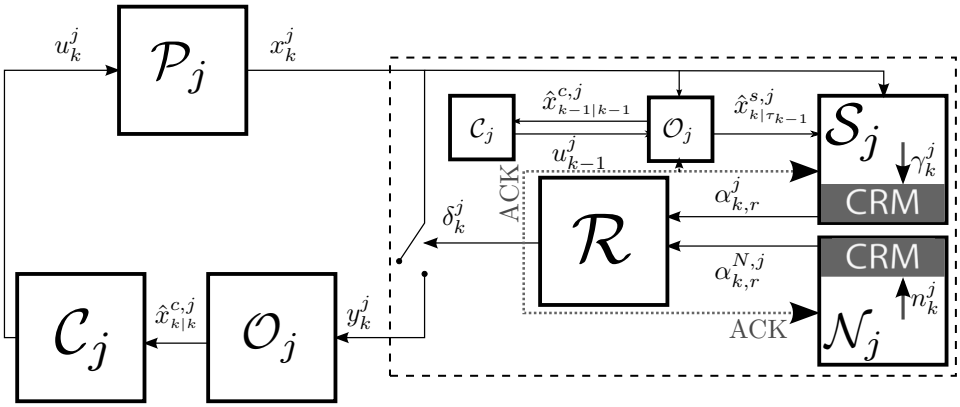


Figure 5.1: A model of the control system from the perspective of a single NCS in the network. The other control loops in the network are abstracted by the network traffic block ($\mathcal{N}$). The resolution block ($\mathcal{R}$) maps the CRM output α to the channel access indicator δ. A copy of the observer ($\mathcal{O}$) and controller ($\mathcal{C}$) are required at the scheduler.

below. When the context is clear, we skip the system index j.

Plant: The plant $\mathcal{P}_j$ has state dynamics given by

$$x_{k+1}^j = A_j x_k^j + B_j u_k^j + w_k^j \,, \tag{5.1}$$

where $x_k^j \in \mathbb{R}^n$, $u_k^j \in \mathbb{R}^m$ and the initial state x_0^j and the process noise w_k^j are i.i.d. zero-mean Gaussians with covariance matrices R_0^j and R_w^j, respectively. They are independent and uncorrelated to each other and to the initial states and process noises of other plants in the network. This discrete time model is defined with respect to a sampling period T for each plant, and the sampling instants are generated by a synchronized network clock.

Scheduler: A local scheduler $\mathcal{S}_j$, situated in the sensor node, executes the event-triggering policy. The event indicator is denoted $\gamma_k^j \in \{0, 1\}$, with $\gamma_k^j = 1$ in the case of an event. The event-triggering policy uses the innovations process to determine γ_k^j, as given by

$$\gamma_k^j = \begin{cases} 1, & ||x_k^j - \hat{x}_{k|\tau_{k-1}}^{s,j}|| > \Delta_d^j, \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

Here, $\hat{x}_{k|\tau_{k-1}}^{s,j} = A_j \hat{x}_{k-1|k-1}^{c,j} + B_j u_{k-1}^j$ and $\hat{x}_{k-1|k-1}^{c,j}$ denotes the estimate at the controller, defined in (5.5) below. Furthermore, τ_k^j is the time index of the last received packet, given by $\tau_k^j = \{\max\{n, -1\} : \delta_n^j = 1, n \leq k\}$. Also, $\Delta_d^j > 0$ is the event threshold, and it may vary with the delay $d = d_k^j \triangleq k - \tau_k^j$. The parameters τ_k^j and d_k^j are illustrated in Figure 3.3. To realize the above event-triggering

policy, the observer and controller must be replicated within the scheduler, and an explicit acknowledgement (ACK) of a successful transmission is required.
Network: The network $\mathcal{N}$ generates exogenous traffic, as is indicated by $n_k^j \in \{0,1\}$. It takes a value 1 when a network source generates an event, and 0 otherwise. The network traffic is stochastic, and hence, $n_k^j \in \{0,1\}$ is not required to be i.i.d.
CRM: The CRM resolves contention between simultaneous channel access requests. For simplicity, we assume that the network uses p-persistent CSMA with either no retransmissions or multiple retransmissions. We describe the CRM without retransmissions here, and explain how our model extends to the multiple retransmissions case in Section 5.2.2. The CRM output is denoted $\alpha_k^j \in \{0,1\}$, and we have

$$\mathrm{P}(\alpha_k^j = 1 | \gamma_k^j = 1) = p_\alpha \tag{5.3}$$

where p_α denotes the persistence probability of the CRM. Thus, with probability $q_\alpha = 1 - p_\alpha$, some events are suppressed by the CRM and not permitted to access the medium. Similarly, $\alpha_k^{N,j}$ is the CRM output for the rest of the network, and the persistence probability remains the same, i.e., $\mathrm{P}(\alpha_k^{N,j} = 1 | n_k^j = 1) = p_\alpha$.

The resolution block ($\mathcal{R}$) maps the CRM outputs α_k^j and $\alpha_k^{N,j}$ to the channel access indicator δ_k^j, as given by

$$\delta_k^j = \alpha_k^j (1 - \alpha_k^{N,j}) \tag{5.4}$$

where $(\delta_k^j = 1)$ indicates that a successful transmission of the event has occurred. This is possible only when the CRM permits a transmission *and* none of the other nodes attempt to transmit.
Observer ($\mathcal{O}_j$): The input to the observer is the received measurement signal $y_k^j = \delta_k^j x_k^j$. The observer generates the estimate $\hat{x}_{k|k}^{c,j}$ as given by

$$\hat{x}_{k|k}^{c,j} = (1 - \delta_k^j)(A_j \hat{x}_{k-1|k-1}^{c,j} + B_j u_{k-1}^j) + \delta_k^j x_k^j \,, \tag{5.5}$$

where the estimate for $\delta_k^j = 0$ is the model-based prediction from the last received data packet at time τ_k^j. The estimation error is defined as $\tilde{x}_{k|k}^{c,j} \triangleq x_k^j - \hat{x}_{k|k}^{c,j}$, and $P_{k|k}^j = \mathbb{E}[\tilde{x}_{k|k}^{c,j} (\tilde{x}_{k|k}^{c,j})^\top]$ is the covariance of the estimation error. We denote the variance as $\mathsf{tr}\{P_{k|k}^j\}$, where tr is the trace operator.
Controller ($\mathcal{C}_j$): The controller generates the signal u_k^j as given by

$$u_k^j = -L_j \hat{x}_{k|k}^{c,j} \,, \tag{5.6}$$

where L_j is the controller gain chosen to minimize a control cost, such as an infinite horizon linear quadratic Gaussian (LQG) cost function.

We are interested in investigating mean square boundedness of the plant state in steady state, or equivalently Lyapunov mean square stability. It is defined below for a control system in the above network. We skip the index j as the definition is applicable for each control system.

Definition 5.1 (Lyapunov Mean Square Stability (Kozin, 1969))**.** *A state is said to possess Lyapunov mean square stability if given $\zeta > 0$, there exists $\xi(\zeta) > 0$ such that $|x_0| < \xi$ implies*

$$\limsup_{k\to\infty} \mathbb{E}[x_k^\top x_k] \leq \zeta \ . \tag{5.7}$$

The certainty equivalence principle has been shown to hold in the architecture described in (5.1)–(5.6) in Chapter 3. Thus, we can translate the stability property in Definition 5.1 from the state to the estimation error, as shown below.

Lemma 5.1. *For a control system in the network given by (5.1)–(5.6), there exists a constant ς, with $0 < \varsigma < \zeta$, such that (5.7) is equivalent to*

$$\limsup_{k\to\infty} tr\{\mathbb{E}[P_{k|k}]\} \leq \varsigma \ . \tag{5.8}$$

Proof. The estimate at the controller in (5.5) can be rewritten as

$$\hat{x}^c_{k|k} = (A - BL)\hat{x}^c_{k-1|k-1} + \delta_k(A\tilde{x}^c_{k-1|k-1} + w_{k-1}) \ . \tag{5.9}$$

Since $\hat{x}^c_{k-1|k-1}$ is the minimum mean square error estimate (see Theorem 3.8), we have $\mathbb{E}[x_k^\top x_k] = \mathsf{tr}\{(A-BL)\,\mathbb{E}[\hat{x}^c_{k-1|k-1}(\hat{x}^c_{k-1|k-1})^\top](A-BL)^\top\} + \mathsf{tr}\{\mathbb{E}[P_{k|k}]\}$, which must be bounded in steady state for stability, as per Definition 5.1. Certainty equivalence implies that the control law ensures mean square boundedness of the estimate $\hat{x}^c_{k-1|k-1}$ in (5.9). Hence, the stability condition depends only on the estimation error, so x_k possesses Lyapunov mean square stability iff $\limsup_{k\to\infty} \mathsf{tr}\{\mathbb{E}[P_{k|k}]\} \leq \varsigma$. □

In the rest of the chapter, we identify sufficient conditions that guarantee Lyapunov mean square stability, in the sense of (5.8), for the states of each of the M control systems described above. Furthermore, we seek a design procedure for selecting the event thresholds, Δ_d, so as to guarantee Lyapunov mean square stability for the overall network of systems.

5.2.2 Network Interaction Model

We have defined a model and a notion of stability for each control system. Next, we model the interactions in the network of M control systems, and define a notion of stability for the entire network. We first present an example to motivate the need for such a model.

Example 5.1

We consider scenarios corresponding to $M \in \{2, 4, 6, 8, 10\}$ event-based systems. The rest of the parameters, chosen identically for all the systems in the network, are $A = 1$, $B = 1$, $\Sigma_w = 1$ and $\Delta_d = 0.25$, $\forall d > 0$. The network uses p-persistent CSMA with 10 retransmissions in the CRM. Retransmissions improve the performance of a CRM, and are further explained in Remark 5.2.2. From the simulations shown in Figure 5.2, we see that the upper bound of the state magnitude varies with M.

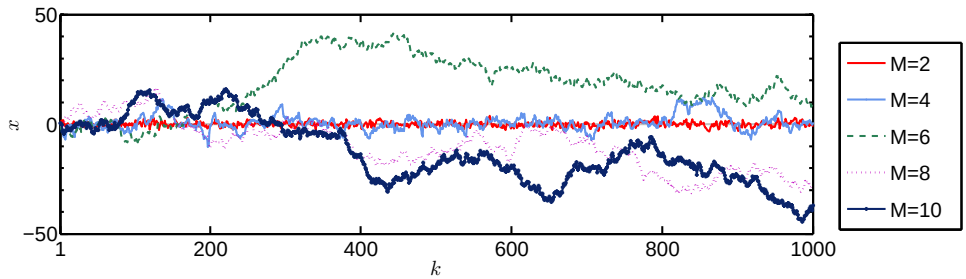


Figure 5.2: A comparison of the trace of the plant state x for identical plants in different sized networks; the event-triggering design appears to result in stability for small sized networks only.

The result of the example is not surprising. However, it is not clear how we can identify the network size that can be supported by a given event-triggering policy. Equivalently, for a given network size, how can we identify a stabilizing event-triggering policy? From the example, it is clear that the network interaction between the event-based systems provides the answer. Thus, we now present a model for the network interactions, and derive stability conditions and stabilizing designs using this model.

We use a Markov chain to jointly model the event-triggering policy and CRM, through which each control system interacts with the rest of the network. Since this model applies to each control system, we skip the index j unless we need it to explain the interaction between multiple systems. However, it is useful to keep in mind that every parameter in the following discussion, including probabilities, are unique to each control system, and must be understood to be indexed by j. The delay d and an index S are used to denote each state in the Markov chain in Figure 5.3. The index $S \in \{I, N, E, T\}$ denotes an idle state (I), a non-event state (N), an event-state (E) and a transmission state (T), respectively. We denote the steady state probability of the state (S, d) as $\pi_{(S,d)}$, and compute these values in Lemma 5.2. A successful transmission brings the system to state $(I, 0)$, where it awaits the next sampling instant. If the packet is not transmitted, either due to a collision or the lack of an event, the delay increases.

Let us trace through the chain for some delay $d_{k-1} = d - 1$, beginning with the plant in the idle state $(I, d-1)$. At the next sampling instant k, the state x_k is declared to be an event or a non-event. The control system transitions to (E, d) with event probability $p_{\gamma,d} \triangleq \mathrm{P}(\gamma_k = 1 | d_{k-1} = d - 1)$, or to (N, d) with complimentary event probability $q_{\gamma,d} = 1 - p_{\gamma,d}$, respectively. From the non-event state (N, d), the system transitions directly to the next idle state (I, d), to wait for the next sampling instant.

An event is sent to the CRM, where it is either transmitted or suppressed. The control system transitions to (T, d) with persistence probability p_α, or returns to

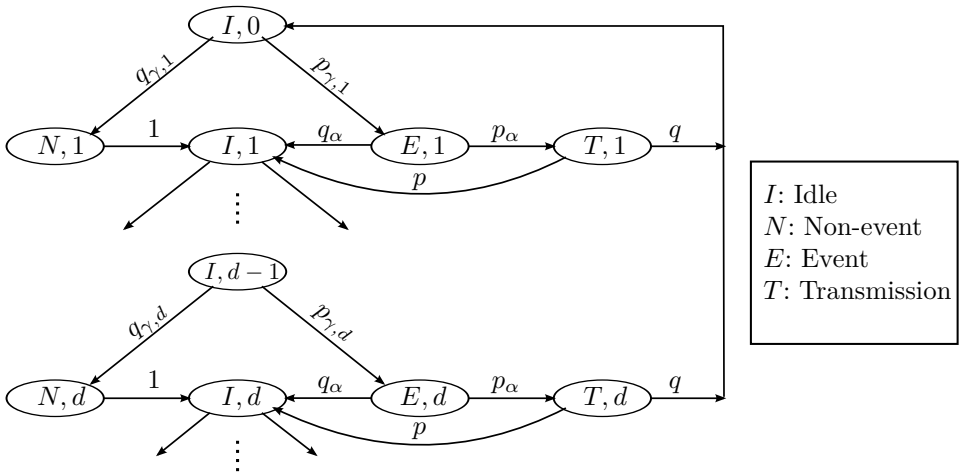


Figure 5.3: A Markov chain representation for the event-triggering policy in (5.2) and p-persistent CSMA with no retransmissions.

the next idle state (I, d) with complimentary persistence probability $q_\alpha = 1 = p_\alpha$, respectively. A system in the transmission state, (T, d), sees a busy channel if another control system in the network is in one of its transmission states, (T, d), for any $d > 0$. This happens with probability p, and the packet is lost due to a collision. The system then returns to the idle state (I, d). With complimentary probability $q = 1 - p$, the transmission is successful, and the system transitions to the state $(I, 0)$, with the delay reset to zero.

We now present our first assumption, used in the construction of the Markov model.

Assumption 5.1 (Bianchi's Conditional Independence Assumption)**.** *The conditional probability of a busy channel for a node that is ready to transmit (in state (T, d), for $d > 0$), is given by an independent probability p. This probability, called the busy channel probability, can be evaluated as*

$$p^j = 1 - \prod_{i \neq j, i=1}^{M} \left(1 - \sum_{d=1}^{\infty} \pi^i_{(T,d)}\right), \tag{5.10}$$

where, $\pi^i_{(T,d)}$ is the steady state probability of the i^{th} control system in the state (T, d) in the Markov model in Figure 5.3.

This assumption was introduced by Bianchi in his much-acclaimed analysis of CSMA/CA in 802.11, and has been verified by many studies (Bianchi, 2000; Ramachandran et al., 2007; Pollin et al., 2008). For the problem setup considered in this chapter, it is verified through simulations in Chapter 4.

Extension to CSMA with Multiple Retransmissions: The CRM typically makes multiple attempts to transmit the same packet, within a single plant sampling period. This is because the operational time-scale of the CRM is much finer that that of the control system itself, as discussed in Chapter 2 and indicated in Figure 2.6. The Markov model presented above can also be used to model such a CRM, by redefining the conditional probability of a busy channel p. We do this by first defining a unique conditional probability of a busy channel, p_r, for each retransmission attempt $1 \leq r \leq r_{\max}$. Applying Assumption 5.1 to each retransmission attempt, the conditional probability of a busy channel in the r^{th} retransmission instant is given by

$$p_r^j = 1 - \prod_{i \neq j, i=1}^{M} \left(1 - \sum_{d=1}^{\infty} \pi^i_{(T,d,r)}\right) , \tag{5.11}$$

where $\pi^i_{(T,d,r)}$ is the steady state probability of the i^{th} control system in the state (T,d) during the r^{th} retransmission attempt. We now redefine the busy channel probability p to represent an aggregate conditional probability of a busy channel across all the retransmission instants, as given by

$$p = 1 - \frac{1}{p_\alpha} \left(1 - \prod_{r=1}^{r_{\max}} (1 - p_\alpha q_r)\right) , \tag{5.12}$$

where q_r denotes the complimentary probability $1 - p_r$.

We now analyze the reliability of a link, defined as the probability of a successful transmission, in this network.

Lemma 5.2 (Reliability Analysis)**.** *For a network of event-based systems described by (5.1)–(5.6) under Assumption 5.1, the network reliability in steady state is given by* $\lim_{k\to\infty} \mathrm{P}(\delta_k = 1) = \pi_{(I,0)} = q \cdot \sum_{d=0}^{\infty} \pi_{(T,d)}$.

Proof. The steady state distribution of the Markov chain, $\pi_{(S,d)}$ corresponding to the state (S,d), can be calculated when Assumption 5.1 holds. The steady state probabilities of a node in the states (I,d) and (T,d), respectively, are given by

$$\pi_{(I,d)} = (1 - p_{\gamma,d} p_\alpha q)\pi_{(I,d-1)} , \tag{5.13}$$

$$\pi_{(T,d)} = p_{\gamma,d} p_\alpha \pi_{(I,d-1)} . \tag{5.14}$$

Then, the probability of a successful transmission is given by $\mathrm{P}(\delta_k = 1) = \pi_{(I,0)}$ and can be obtained by simultaneously solving $\sum_{d=0}^{\infty} \pi_{(I,d)} = 1$ with (5.10) or (5.12). □

We now define network steady state as a notion of stability for the network of M control systems.

Definition 5.2. *The network of M control systems is said to be in steady state when $0 \leq p < 1$, for the busy channel probability p in (5.12).*

When $p = 1$, no transmissions occur in the Markov chain in Figure 5.3. Thus, the network steady state property simply implies that at least some transmissions occur successfully in the network. We show that network steady state is a necessary condition for Lyapunov mean square stability, for unstable plants.

Proposition 5.3. *For unstable plants with spectral radius $\rho(A) > 1$ in the network given by (5.1)–(5.6), network steady state is a necessary condition for Lyapunov mean square stability, under Assumption 5.1.*

Proof. The states (S, d), $\forall S \in \{I, N, E, T\}, d \geq 0$, are transient when the busy channel probability $p = 1$, except for the infinite-delay states. For an unstable system, the condition for Lyapunov mean square stability given by (5.8) cannot be satisfied when $p = 1$, as the variance of the estimation error at infinite delay is not bounded. □

The above lemma clarifies that a control system cannot be Lyapunov mean square stable without network stability, in the sense defined above. Thus, we begin with the necessary condition that network steady state exists, and then proceed to find conditions for Lyapunov mean square stability. Network steady state is not sufficient to guarantee Lyapunov mean square stability. However, the Lyapunov mean square stability conditions we derive guarantee that network steady state holds.

Necessary Conditions for Bianchi's Assumption to Hold: The existence of the independent process in Bianchi's assumption has been studied by Bordenave et al. (2010), among others. The conditions for the decoupling to occur would provide necessary conditions for Proposition 5.3. An analysis of such conditions for stability of the control system is out of the scope of this chapter.

From Event Thresholds to Event Probabilities: Note that the event thresholds do not directly appear in the Markov model in Figure 5.3. The model uses a set of event probabilities $\{p_{\gamma,d}\}$, in place of the event thresholds $\{\Delta_d\}$ to represent the event-triggering policy. The event probabilities are obtained using the event thresholds and the underlying distributions. This alternative representation affords no loss of generality.

We now summarize the design approach used in this chapter. We use a two step strategy to design a stabilizing event-triggering policy. First, we select a stabilizing set of event probabilities, and then, we find a set of event thresholds that result in the designed event probabilities. The motivation for this strategy is because our analysis of a network of event-based systems is parameterized by the event probabilities, rather than the triggering levels, as we saw in the above model. To select suitable event probabilities, we first find conditions that guarantee stability of the control systems in the network, and then outline a design process based on these conditions, as depicted in Figure 5.4.

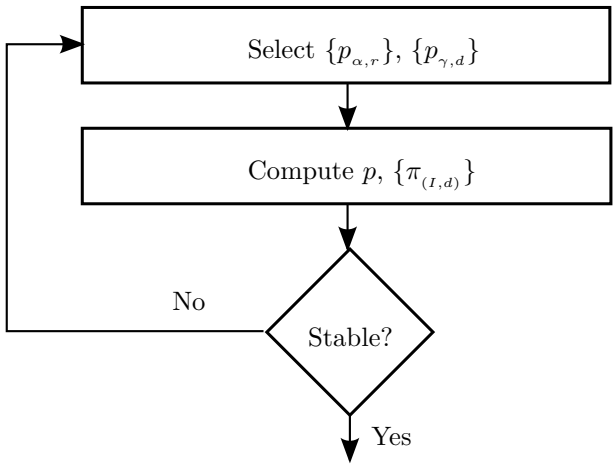


Figure 5.4: The event probabilities $\{p_{\gamma,d}\}$ and the persistence probability $\{p_{\alpha,r}\}$ are the inputs to our design process. The stability conditions that we derive in this chapter give us a stability guarantee, as an output. The Markov model parameters, p in (5.12) and $\{\pi_{(I,d)}\}$ in (5.13), are intermediary parameters that must be computed to check the stability conditions.

5.3 Stability Analysis

We use the Markov chain in Figure 5.3 to analyze the stability of each control system in this section. This analysis includes the effect of all the other control systems in the network through the parameters of the Markov chain. We dispense with the system index j in this section, as our analysis applies to each control system in the network. We begin with the main stability result in this section, and then develop its proof. To arrive at this proof, we examine the underlying density of the estimation error, and construct an auxiliary system that furnishes an upper bound for the variance of each control system.

5.3.1 Main Result: Stability Conditions for the Markov Chain

We begin by presenting one of the main results of the chapter. It is a sufficient condition for stabilizing the Markov chain in Figure 5.3, in a Lyapunov mean square sense.

Theorem 5.4. *Consider the network of control systems (5.1)–(5.6) and suppose Assumption 5.1 holds. Let $\rho(A_j)$ denote the spectral radius of the j^{th} control system.*

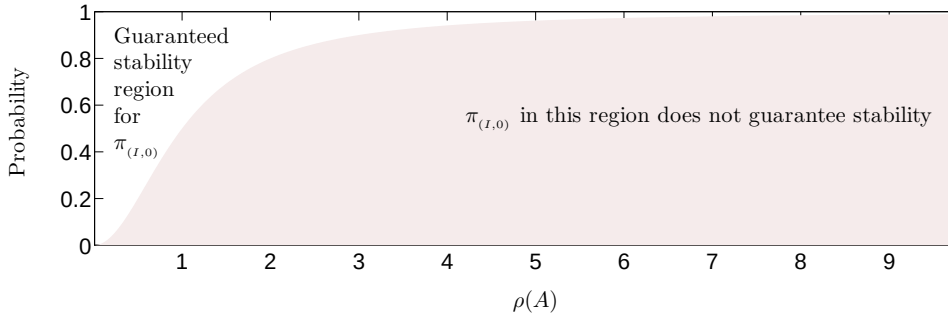


Figure 5.5: A sufficient condition for Lyapunov mean square stability that requires the network reliability, $\pi_{(I,0)}$, to be greater than the line demarcating the regions, with respect to the spectral radius $\rho(A)$. Thus, unstable processes require a higher network reliability to guarantee stability.

For $1 \leq j \leq M$*, if*

$$\limsup_{d\to\infty} \frac{\pi^j_{(I,d+1)}}{\pi^j_{(I,d)}} < \frac{1}{1+\rho(A_j)^2} \tag{5.15}$$

holds, then each of the control systems in the network is Lyapunov mean square stable.

The proof is presented in Section 5.3.5. The above result requires the probability of the idle states in the tail of the Markov model in Figure 5.3 to decrease in a stipulated manner, as determined by the spectral radius of each control system $\rho(A)$. The larger the value of $\rho(A)$, the sharper is the mandated fall off in the probabilities of the idle states.

The role of the spectral radius suggests a similarity to other mean square stability results in NCSs, particularly for packet losses in the sensing or actuation channel (Gupta and Martins, 2010; Kar et al., 2012) and encoder design for data rate limited channels (Nair and Evans, 2004; Tatikonda and Mitter, 2004a,b). The results for packet loss channels specifies a critical probability of loss, beyond which a control system cannot be stabilized in the mean square sense. This result is obtainable only under a Bernoulli packet loss model, which cannot be applied to our problem setup. The stability result in the case of encoder design specifies a stabilizing rate, derived from a source coding perspective. The Markov model we consider in Figure 5.3 is quite general, and a more specific stability result is difficult to find. In practice, one must find a finite parameterization of the Markov model parameters to obtain a stability condition that can be checked, as illustrated for an event-triggering design that we present in Section 5.4.

Plant versus network stability: Network steady state is not sufficient to guar-

antee Lyapunov mean square stability. This can be seen by noting that the condition for the busy channel probability, $p < 1$, as required by network steady state, implies that $\pi_{(I,d+1)} < \pi_{(I,d)}$. Thus, network steady state ensures that the loop is sometimes closed, as against the case $p = 1$, when the loop is never closed. But, this feedback may not be sufficient to stabilize the control system, in the sense of Definition 5.1. However, Lyapunov mean square stability for all M control systems in the network ensures a network steady state, in the sense of Definition 5.2. To see this, note that $\pi_{(I,d+1)} < \frac{\pi_{(I,d)}}{1+\rho(A)^2} < \pi_{(I,d)}$, for all $\rho(A) > 0$. Hence, network steady state is indeed achieved by the control systems in stability.

Unstable Processes: Let us assume that we choose the event probabilities such that (5.15) is true for all $d \geq 0$, as opposed to the tail of the sequence alone. Then, using (5.15) in $\sum_{d=0}^{\infty} \pi_{(I,d)} = 1$, we get a lower bound for the network reliability as $\pi_{(I,0)} > \rho(A)^2/(1+\rho(A)^2)$. This implies that the network reliability must lie above the line shown in Figure 5.5, and thus, unstable processes require a higher network reliability to guarantee stability.

Role of the Persistence Probability: Using the recursive relationship for the idle state probabilities $p_{(I,d+1)}$ in (5.13), along with the sufficient condition in (5.15), we obtain

$$\limsup_{d\to\infty} \frac{(1 - p_{\gamma,d+1} p_\alpha q)\pi_{(I,d)}}{\pi_{(I,d)}} < \frac{1}{1+\rho(A)^2} \ ,$$

which can be rearranged to obtain $\limsup_{d\to\infty} p_{\gamma,d} q > \kappa_\alpha$, where $\kappa_\alpha = \frac{1}{p_\alpha} \frac{\rho(A)^2}{1+\rho(A)^2}$. The value of κ_α can be tuned by varying the persistence probability p_α. A small value for p_α can increase the lower bound for $p_{\gamma,d} q$, which in turn can improve the network reliability.

We first attempt to prove Theorem 5.4 directly by examining the underlying density of the estimation error. This is a difficult approach, as we show in Section 5.3.2. Then, we construct auxiliary systems in Sections 5.3.3 and 5.3.4, and use these systems to prove Theorem 5.4 in Section 5.3.5.

5.3.2 A Difficult Direct Approach

We seek an expression for the variance of the estimation error. Let us associate with each state (S, d) for $S \in \{I, N, E, T\}$ a probability density function (PDF) for the estimation error (filtered or predicted) at the controller, denoted by $\phi_{(S,d)}$, for the appropriate estimation error corresponding to the state (S, d) of the Markov model.

Then, the variance of the estimation error conditioned on a delay d is given by $\mathsf{tr}\{P_d\}$, where $P_d = \int_{-\infty}^{\infty} \tilde{x}\tilde{x}^\top \phi_{(I,d)}(\tilde{x}) d\tilde{x}$. Marginalizing over the idle state distribution, we get

$$\mathsf{tr}\{\mathbb{E}[P_{k|k}]\} = \sum_{d=0}^{\infty} \mathsf{tr}\{P_d\} \pi_{(I,d)} \ . \tag{5.16}$$

The above expression is simple, but the PDFs can be hard to evaluate. To see why, let us look at the evolution of these PDFs as the delay d increases. For $d = 0$, $\phi_{(I,0)} := \lim_{k\to\infty} \phi(\tilde{x}^c_{k|k-1}|\delta_{k-1} = 1)$ is the PDF associated with the predicted estimate, one step after a transmission. Clearly, $\phi_{(I,0)} = \phi_N(\Sigma_w)$, where ϕ_N is the PDF of a normal distribution with covariance Σ_w. For any delay d, the PDFs associated with the event ($\gamma_k = 1$) and non-event ($\gamma_k = 0$) states are truncated versions of the PDF associated with the previous idle state. They can be defined as $\phi_{(N,d)} := \lim_{k\to\infty} \phi(\tilde{x}^c_{k|\tau_{k-1}}|\gamma_k = 0, d_{k-1} = d-1)$ and $\phi_{(E,d)} := \lim_{k\to\infty} \phi(\tilde{x}^c_{k|\tau_{k-1}}|\gamma_k = 0, d_{k-1} = d-1)$, respectively. Thus, we get

$$\phi_{(N,d)} = \begin{cases} \frac{\phi_{(I,d-1)}(\tilde{x})}{q_{\gamma,d}} & |\tilde{x}| \le \Delta_d \ , \\ 0 & \text{otherwise} \ , \end{cases} , \quad \phi_{(E,d)} = \begin{cases} \frac{\phi_{(I,d-1)}(\tilde{x})}{p_{\gamma,d}} & |\tilde{x}| > \Delta_d \ , \\ 0 & \text{otherwise} \ , \end{cases} \tag{5.17}$$

where, $q_{\gamma,d} = \int_{-\Delta_d}^{\Delta_d} \phi_{(I,d-1)}(\tilde{x})d\tilde{x}$ is the probability of a non-event and $p_{\gamma,d} = 1 - q_{\gamma,d}$ is the probability of an event.

Then, let us denote e_d as the innovations process that does not get transmitted after a delay d, and denote its PDF as $\phi^e_{(I,d)} := \lim_{k\to\infty} \phi(\tilde{x}^c_{k|k}|\delta_k = 0, d_k = d)$. This PDF can be rewritten as

$$\phi^e_{(I,d)} = \phi_{(N,d)}(\tilde{x}) \cdot \frac{q_{\gamma,d}}{q_{\gamma,d} + p_{\gamma,d}(q_\alpha + pp_\alpha)} + \phi_{(E,d)}(\tilde{x}) \cdot \frac{(q_\alpha + pp_\alpha)p_{\gamma,d}}{q_{\gamma,d} + p_{\gamma,d}(q_\alpha + pp_\alpha)}$$

Substituting for $\phi_{(N,d)}$ and $\phi_{(E,d)}$ from (5.17), we obtain

$$\phi^e_{(I,d)} = \begin{cases} \phi_{(I,d-1)}(\tilde{x}) \cdot \frac{1}{q_{\gamma,d} + p_{\gamma,d}(q_\alpha + pp_\alpha)} & |\tilde{x}| \le \Delta_d \\ \phi_{(I,d-1)}(\tilde{x}) \cdot \frac{(q_\alpha + pp_\alpha)}{q_{\gamma,d} + p_{\gamma,d}(q_\alpha + pp_\alpha)} & |\tilde{x}| > \Delta_d \end{cases} \tag{5.18}$$

The PDF of the idle state with delay d is defined as $\phi_{(I,d)} := \lim_{k\to\infty} \phi(\tilde{x}^c_{k+1|\tau_k}|d_k = d)$. For a plant with an invertible A matrix[1], we can use the state update equation in (5.1) to find an expression for $\phi_{(I,d)}$ as

$$\phi_{(I,d)} = \frac{1}{|\det(A)|} \phi^e_{(I,d)}(A^{-1}\tilde{x}) * \phi_N(\Sigma_w) \ , \tag{5.19}$$

where $*$ denotes the convolution operator.

The above operations must be performed recursively, to obtain the PDF associated with the state (I, d). This computation is in general hard. Hence, we find auxiliary systems that result in upper bounds for the variance of the estimation error in the following subsections.

[1]If the matrix A is non-invertible, the PDF corresponding to the idle state is no longer defined in $\mathbb{R}^n$. A measure on the subspace orthogonal to the null-space of A is absolutely continuous w.r.t. the Lebesgue measure, and the PDF, along with the expected value or covariance, is defined using this measure. Thus, the approach presented in this chapter is applicable for plants with non-invertible A matrices. However, for ease of exposition, we present the results assuming that A is invertible.

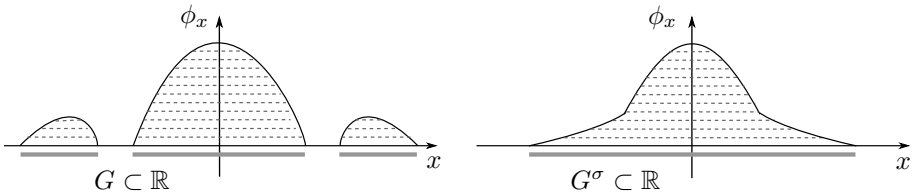


Figure 5.6: An illustration of symmetric rearrangement in Definition 5.4. The level sets of the PDF on the left are symmetrically placed around the origin, to obtain the symmetric rearrangement on the right. An example of a level set is G, given by the union of the three shaded segments on the left. The symmetric rearrangement of this level set results in G^σ on the right, with length equal to the sum of the lengths of the three shaded segments on the left. From this figure, it is easy to see how the variance of the symmetrically rearranged PDF is always less than the variance of the original PDF (Lemma 5.5).

5.3.3 Auxiliary PDFs for First-Order Systems

We wish to find an upper bound for the variance of the estimation error in the idle states of the Markov chain. To do this, we must first find a sequence of PDFs, $\hat{\phi}_{(I,d)}$, that are more 'spread out' than the PDFs $\phi_{(I,d)}$. We use stochastic majorization to do this. Our approach, in this section, is restricted to first-order systems, due to a symmetry requirement on PDFs. In the following section, we extend these results to higher-order systems using other methods that give us more conservative results. We need the following notation and definitions, adapted from Hajek et al. (2008), to define majorization.

Definition 5.3 (Symmetric Non-increasing Function)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be symmetric non-increasing if $f(x) = \phi(|x|)$, for some non-increasing function ϕ on $\mathbb{R}^+$, where $|x|$ denotes the Euclidean norm of $x \in \mathbb{R}^n$.*

Given any integrable, non-negative function, we wish to 'rearrange' the function to obtain a symmetric non-increasing function. The exact sense in which we rearrange the function is defined below. We begin with a definition for the symmetric rearrangement of a Borel set. Then, we apply this definition to the level sets of a non-negative function, and obtain its symmetric rearrangement. We illustrate this notion in Figure 5.6.

Definition 5.4 (Symmetric Rearrangement)**.** *Let $G \in \mathcal{B}$ be a Borel set in $\mathbb{R}^n$, with finite Lebesgue measure $\mathcal{L}(G)$. The* symmetric rearrangement *of G, denoted by G^σ, is the open ball in $\mathbb{R}^n$ centered at the origin, with measure $\mathcal{L}(G^\sigma) = \mathcal{L}(G)$.*

For an integrable, non-negative function h on $\mathbb{R}^n$, its symmetric non-increasing

rearrangement, denoted h^σ is given by

$$h^\sigma(x) \triangleq \int_0^\infty I_{\{x':h(x')>l\}^\sigma}(x)dl \ , \tag{5.20}$$

where $I_{\{x':x'\in G\}^\sigma}(x) = \{x : x \in G^\sigma\}$, denotes the set of elements belonging to the symmetric rearrangement of its argument set G.

We now define majorization with the help of the distribution functions corresponding to the symmetrically rearranged densities.

Definition 5.5 (Majorization)**.** *Given two PDFs ϕ_a and ϕ_b on $\mathbb{R}^n$, we say that ϕ_a majorizes ϕ_b, denoted as $\phi_a \succ \phi_b$, if*

$$\int_{|x|\le\rho} \phi_a^\sigma(x)dx \ge \int_{|x|\le\rho} \phi_b^\sigma(x)dx \ , \ \forall \rho \ge 0 \ .$$

Thus, ϕ_a, as per the above definition is more contained, or less spread out, than ϕ_b. Some results involving the majorization operator are listed in Appendix B. The most important consequence for us is that we obtain an upper bound for the estimation error variance. This is stated below.

Lemma 5.5 (Ordering of Estimation Error Variance)**.** *If ϕ_a and ϕ_b are symmetric non-increasing PDFs on $\mathbb{R}^n$ such that $\phi_a \succ \phi_b$, then $\int_{-\infty}^{\infty} |x|^2\phi_a(x)dx \le \int_{-\infty}^{\infty} |x|^2\phi_b(x)dx$.*

Proof. Use $h(x) = |x|^2$ in Lemma B.5 to obtain the results. □

We now describe the PDFs that we are interested in, as adapted from Lipsa and Martins (2011).

Definition 5.6 (Neat PDF)**.** *We say that a PDF ϕ is* neat *if it is quasi-concave and if there exists a real number r such that ϕ is non-decreasing on $(-\infty, r]$ and non-increasing on $[r, \infty)$.*

Note that PDFs on $\mathbb{R}$ are symmetric non-increasing if and only if they are neat and even. Thus, for neat PDFs, the definition of majorization can be directly applied to the PDF itself. Using Definition 5.5, we find a more spread out $\hat{\phi}_{(I,d)}$, as stated below.

Lemma 5.6. *Let the auxiliary PDF, $\hat{\phi}_{(I,d)}$, be defined by the recursive relation*

$$\hat{\phi}_{(I,d)} = \frac{1}{A}\hat{\phi}_{(I,d-1)} * \phi_N \ ,$$

with $\hat{\phi}_{(I,0)} = \phi_N$. Then, $\phi_{(I,d)} \succ \hat{\phi}_{(I,d)}$ for all $d \ge 0$.

Proof. We show this using induction. Trivially, at $d = 0$, $\phi_{(I,0)} = \hat{\phi}_{(I,0)} = \phi_N$. Let us assume that, for some d, $\phi_{(I,d)} \succ \hat{\phi}_{(I,d)}$. Then, from (5.18), we can show that $\phi^e_{(I,d+1)} \succ \phi_{(I,d)}$. To see this, recall that $\phi^e_{(I,d+1)}$ is obtained by appropriately combining the truncated PDFs for the event and non-event states, as shown in (5.18). Then, we have

- For $|e| \leq \Delta_{d+1}$, we have

$$\int_{-e}^{e} \frac{\phi_{(I,d)}(\tilde{x})}{q_{\gamma,d+1} + p_{\gamma,d+1}(q_\alpha + pp_\alpha)} d\tilde{x} \geq \int_{-e}^{e} \phi_{(I,d)}(\tilde{x}) d\tilde{x} \ ,$$

  because $q_{\gamma,d+1} + p_{\gamma,d+1}(q_\alpha + pp_\alpha) \leq 1$.

- For $|e| > \Delta_{d+1}$, we have

$$\begin{aligned} &\int_{-\infty}^{-e} \phi_{(I,d)}(\tilde{x}) \frac{(q_\alpha + pp_\alpha)}{q_{\gamma,d+1} + p_{\gamma,d+1}(q_\alpha + pp_\alpha)} d\tilde{x} \\ &\quad + \int_{e}^{\infty} \phi_{(I,d)}(\tilde{x}) \frac{(q_\alpha + pp_\alpha)}{q_{\gamma,d+1} + p_{\gamma,d+1}(q_\alpha + pp_\alpha)} d\tilde{x} \\ &\quad \leq \int_{-\infty}^{-e} \phi_{(I,d)}(\tilde{x}) d\tilde{x} + \int_{e}^{\infty} \phi_{(I,d)}(\tilde{x}) d\tilde{x} \ , \end{aligned}$$

  because $\frac{(q_\alpha + pp_\alpha)}{q_{\gamma,d+1} + p_{\gamma,d+1}(q_\alpha + pp_\alpha)} \leq 1$.

Since $\phi^e_{(I,d+1)} \succ \phi_{(I,d)}$ and $\phi_{(I,d)} \succ \hat{\phi}_{(I,d)}$, we have

$$\begin{aligned} \phi^e_{(I,d+1)} &\succ \hat{\phi}_{(I,d)} \\ \frac{1}{A}\phi^e_{(I,d+1)}(\frac{\tilde{x}}{A}) &\succ \frac{1}{A}\hat{\phi}_{(I,d)}(\frac{\tilde{x}}{A}) \\ \phi_N * \frac{1}{A}\phi^e_{(I,d+1)}(\frac{\tilde{x}}{A}) &\succ \phi_N * \frac{1}{A}\hat{\phi}_{(I,d)}(\frac{\tilde{x}}{A}) \ , \end{aligned}$$

where, the last two expressions are obtained from the results of Lemma B.4 and Lemma B.1, respectively. Hence, $\phi_{(I,d+1)} \succ \hat{\phi}_{(I,d+1)}$. □

Worst-Case Evolution of the System: The PDFs given by $\hat{\phi}_{(I,d)}$ correspond to the evolution of the control system when the busy channel probability $p = 1$, in the Markov model. For such systems, no event is successfully transmitted, and hence, the density of the tail ($|\tilde{x}| > \Delta$) is never reduced, due to a perpetually busy channel ($p = 1$). Thus, the gaussian property of the estimation error is retained and its PDF is given by $\hat{\phi}_{(I,d)}$.

5.3.4 Auxiliary PDFs for Higher-Order Systems

In this section, we find an upper bound for the variance of the estimation error, for higher-order systems. The PDF of the state for such control systems need not be symmetric, and hence, the results developed in the previous section cannot be directly applied to such systems. We denote the multivariate PDFs in this section with Φ in place of ϕ.

We now find an upper bound for the variance of the estimation error associated with the Markov chain states (I, d) for all $d \geq 0$, by finding suitable PDFs $\hat{\Phi}_{(I,d)}$. We first define the matrix $\bar{A} = \rho(A)I_n$, where $\rho(A)$ is the spectral radius of A and $I_n \in \mathbb{R}^{n\times n}$ is an identity matrix. Let $\mathrm{var}(\Phi)$ denote the variance of the PDF Φ. We now have the following bound on the variance, cf. Lemma 5.6.

Lemma 5.7. *Let the auxiliary PDF, $\hat{\Phi}_{(I,d)}$, be defined by the recursive relation*

$$\hat{\Phi}_{(I,d)}(\tilde{x}) = \frac{1}{|\det(\bar{A})|}\hat{\Phi}_{(I,d-1)}(\bar{A}^{-1}\tilde{x}) * \Phi_N \ ,$$

with $\hat{\Phi}_{(I,0)} = \phi_N(\Sigma_w)$. Then, $\mathrm{var}(\phi_{(I,d)}) \leq \mathrm{var}(\hat{\Phi}_{(I,d)})$ for all $d \geq 0$.

Proof. We show this using induction. Trivially, at $d = 0$, $\Phi_{(I,0)} = \hat{\Phi}_{(I,0)} = \Phi_N(\Sigma_w)$. Thus, the variances are equal for $d = 0$. Let us assume that, for some $d > 0$, $\mathrm{var}(\phi_{(I,d)}) \leq \mathrm{var}(\hat{\Phi}_{(I,d)})$. We denote the variance of e_d, the innovations process that does not get transmitted after a delay d, as $\mathrm{var}(\phi^e_{(I,d+1)})$, following the notation in (5.18). Then, we show that $\mathrm{var}(\phi^e_{(I,d+1)}) \leq \mathrm{var}(\Phi_{(I,d)})$ in Lemma B.6. Combining this with our induction assumption, we obtain $\mathrm{var}(\hat{\Phi}_{(I,d)}) \geq \mathrm{var}(\phi_{(I,d)}) \geq \mathrm{var}(\phi^e_{(I,d+1)})$.

At the next sampling instant, the state is updated according to the state-space model, with a linear transformation and an addition of process noise. The linear transformation of a random vector results in the PDFs denoted $\Phi^{e,+}_{(I,d+1)}$ for the original system, and $\hat{\Phi}^{+}_{(I,d)}$ for the auxiliary system. The transformed PDFs are given by

$$\begin{aligned}\Phi^{e,+}_{(I,d+1)} &= \frac{1}{|\det(A)|}\Phi^{e}_{(I,d+1)}(A^{-1}\tilde{x}) \\ \hat{\Phi}^{+}_{(I,d)} &= \frac{1}{|\det(\bar{A})|}\hat{\Phi}_{(I,d)}(\bar{A}^{-1}\tilde{x}) \ .\end{aligned}$$

The variances can be written as

$$\begin{aligned}\mathrm{var}(\Phi^{e,+}_{(I,d+1)}) &= \mathsf{tr}\{A\Sigma^e_{(I,d+1)}A^\top\} = \mathsf{tr}\{\Sigma^e_{(I,d+1)}A^\top A\} \\ \mathrm{var}(\hat{\Phi}^{+}_{(I,d)}) &= \mathsf{tr}\{\bar{A}\hat{\Sigma}_{(I,d)}\bar{A}^\top\} = \mathsf{tr}\{\hat{\Sigma}_{(I,d)}\bar{A}^\top\bar{A}\} \ ,\end{aligned}$$

where $\Sigma^e_{(I,d+1)}$ and $\hat{\Sigma}_{(I,d)}$ are the covariance matrices associated with PDFs $\Phi^e_{(I,d+1)}$ and $\hat{\Phi}_{(I,d)}$, respectively. Now, note that $\Sigma^e_{(I,d+1)}$ and $A^\top A$ are symmetric matrices,

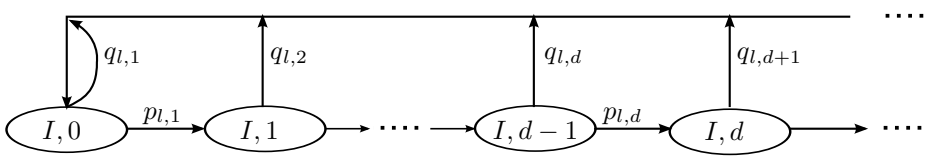


Figure 5.7: The majorized PDFs, from Lemma 5.6, along with the probabilities from the original Markov chain are combined to describe a lossy sensor link. The estimation error variance of this system is an upper bound for the control system.

and that their product $\Sigma^{e,+}_{(I,d+1)}$ is also a symmetric matrix. Thus, the matrices commute, and we can apply the spectral value inequality to obtain $\rho(\Sigma^{e,+}_{(I,d+1)}) \leq \rho(\Sigma^{e}_{(I,d+1)}) \cdot \rho(A^\top A)$. Furthermore, $\mathsf{tr}\{\hat{\Sigma}_{(I,d)} \bar{A}^\top \bar{A}\} = \rho^2(A)\mathrm{var}(\hat{\Phi}_{(I,d)})$. Combining these facts, we obtain $\mathrm{var}(\phi^{e,+}_{(I,d+1)}) \leq \mathrm{var}(\hat{\Phi}^{+}_{(I,d)})$.

We have not yet accounted for the addition of process noise in the state update. This operation results in an addition of a constant term $\mathsf{tr}\{R_w\}$, corresponding to the variance of the process noise w_k, to both the original and auxiliary system. Thus, the variance ordering is preserved, and we have the desired result $\mathrm{var}(\phi_{(I,d+1)}) \leq \mathrm{var}(\hat{\Phi}_{(I,d+1)})$. □

Lossy Network as Upper Bound: The PDFs of the auxiliary systems are used along with the probabilities in the Markov chain in Figure 5.3 to upper bound the variance of the estimation error of the control system. The resulting approximation describes the evolution of a system with a lossy sensor link, albeit with a loss probability that varies with delay, as shown in Figure 5.7. The loss probability is given by $p_{l,d} = 1 - p_{\gamma,d} p_\alpha q$. The estimation error covariance of this system for zero delay is clearly $\hat{P}_0 = \Sigma_w$, and for all other delays $d > 0$, is given by

$$\hat{P}_d = \rho(A)^2 \hat{P}_{d-1} + \Sigma_w \ . \tag{5.21}$$

5.3.5 Proof of Theorem 5.4

Let us now prove the stability conditions in Theorem 5.4 using the auxiliary systems we have identified.

Proof. The estimation error covariance can be bounded from above, using the approximations from Lemma 5.6, as

$$\mathsf{tr}\{\mathbb{E}[P_{k|k}]\} = \sum_{d=1}^{\infty} \pi_{(I,d)} \, \mathsf{tr}\{P_d\} \leq \sum_{d=1}^{\infty} \pi_{(I,d)} \, \mathsf{tr}\{\hat{P}_d\} \ .$$

For this expression to be bounded (Rudin, 1976), we require

$$\limsup_{d\to\infty} \frac{\pi_{(I,d+1)}\,\mathsf{tr}\{\hat{P}_{d+1}\}}{\pi_{(I,d)}\,\mathsf{tr}\{\hat{P}_d\}} < 1\ .$$

Since $\mathsf{tr}\{\hat{P}_d\} = \Sigma_w(1+\rho(A)^2+\cdots+\rho(A)^{2(d-1)})$, the left hand side of the above inequality can be written as

$$\begin{aligned}&\limsup_{d\to\infty} \frac{\pi_{(I,d+1)}}{\pi_{(I,d)}}\left[1+\rho(A)^2\frac{\rho(A)^{2(d-1)}}{1+\rho(A)^2+\cdots+\rho(A)^{2(d-1)}}\right]\\ \leq &\limsup_{d\to\infty} \frac{\pi_{(I,d+1)}}{\pi_{(I,d)}}[1+\rho(A)^2]\ .\end{aligned}$$

By requiring the last expression to be strictly less than 1, we satisfy the condition in (5.15) required to obtain Lyapunov mean square stability. □

5.4 Event-Triggering Policy Synthesis

We now look at the problem of designing stabilizing event-triggering policies. In particular, how should the event probabilities $\{p_{\gamma,d}\}$ be chosen as a function of d to achieve Lyapunov mean square stability? We can immediately think of three possible ways to let the event probabilities vary with the delay: holding it a constant, additively increasing or decreasing it, or multiplicatively increasing or decreasing it. We discuss the constant-probability policy in detail and identify stability conditions for such policies. We then discuss the feasibility of the other policies briefly.

5.4.1 Constant-Probability Policy

The constant-probability policy provides a constant event probability for all delays, i.e. $p_{\gamma,d} = p_\gamma$, for all $d > 0$. Using the lossy network model from Section 5.3, we identify stability conditions for this particular policy.

Theorem 5.8. *For the control system given by (5.1)–(5.6), a sufficient condition for Lyapunov mean square stability for the constant-probability event-triggering policy is given by*

$$p_\gamma\left(\sum_{r=1}^{r_{\max}}\left(q_r\cdot\prod_{r=1}^{r_{\max}-1}(1-p_\alpha q_r)\right)\right) > \frac{1}{p_\alpha}\left(1-\frac{1}{\rho(A)^2}\right)\ . \tag{5.22}$$

Proof. Using a constant scheduling law, note that the lossy network model has a constant loss probability, $p_l = 1-p_\gamma p_\alpha q$. Thus, the estimation error variance in this model, given in (5.21), converges if

$$p_l\rho(A)^2 < 1\ .$$

Substituting for p_l above from (5.12), and rearranging, we obtain the condition in (5.22). □

The constant-probability policy results in simple, closed-form expressions for the probability of successful transmission $p_{I,0}$ and the loss probability p_l. To see this, note that the sum of the probabilities of the idle states is given by the sum of a geometric series, $\sum_{d=0}^{\infty} \pi_{(I,d)} = \pi_{(I,0)} \sum_{d=0}^{\infty} p_l^d$. Thus, we have

$$\pi_{(I,0)} = p_\gamma p_\alpha q \,, \tag{5.23}$$

using the expression for the loss probability, where q is the complimentary busy channel probability $1 - p$. The conditional probability of a busy channel in each retransmission attempt can be computed using (5.11) and (5.14) as

$$\sum_{d=1}^{\infty} p_{T,d,r} = p_\gamma p_\alpha \prod_{s=1}^{r} (1 - p_\alpha q_s) \sum_{d=1}^{\infty} \pi_{(I,d-1)}$$

which leads to

$$p_r = 1 - \left(1 - p_\gamma p_\alpha \prod_{s=1}^{r} (1 - p_\alpha q_s)\right)^{M-1} \,, \; r \in \{1, \ldots, r_{\max}\} \tag{5.24}$$

Equations (5.23)–(5.24) along with (5.22), gives us the event probability $p_{\gamma,d}$ for a given persistence probability p_α, that guarantees Lyapunov mean square stability. The flowchart in Figure 5.4 gives us a set of p_α and p_γ that generate a stable system. We present an example of this design procedure in Section 5.3.

5.4.2 Additive-Probability Policy

The additive-probability policy is designed to provide an additive increase/decrease in the event probability with delay, i.e., $p_{\gamma,d} = p_{\gamma,d-1} + \nu$, for $\nu \gtrless 0$. Note that $\lim_{d\to\infty} p_{\gamma,d} \to \infty$ for $\nu > 0$ and $\lim_{d\to\infty} p_{\gamma,d} \to -\infty$ for $\nu < 0$. Thus, we let the additive terms decrease in magnitude, such that $\sum_{d=1}^{\infty} \nu_d$ is bounded and $\lim_{d\to\infty} p_{\gamma,d} < 1$. Many such examples can be found. A simple example is

$$p_{\gamma,d} = p_{\gamma,1} + \eta + \eta^2 + \cdots + \eta^{d-1} \,, \tag{5.25}$$

which gives rise to an increasing law when $\eta > 0$, and a decreasing law when $\eta < 0$. Thus, $p_{\gamma,\infty} = p_{\gamma,1} + \frac{\eta}{1-\eta}$, and $p_{\gamma,1}$ and η must be chosen such that $p_{\gamma,1} < 1$ and $p_{\gamma,\infty} < 1$. Then, we can apply Theorem 5.4 to identify designs that are guaranteed to result in Lyapunov mean square stability.

5.4.3 Exponential-Probability Policy

The exponential-probability policy is designed to provide an exponential increase or decrease in the event probability with delay, i.e., $p_{\gamma,d} = \mu p_{\gamma,d-1}$, for $\mu < 1$.

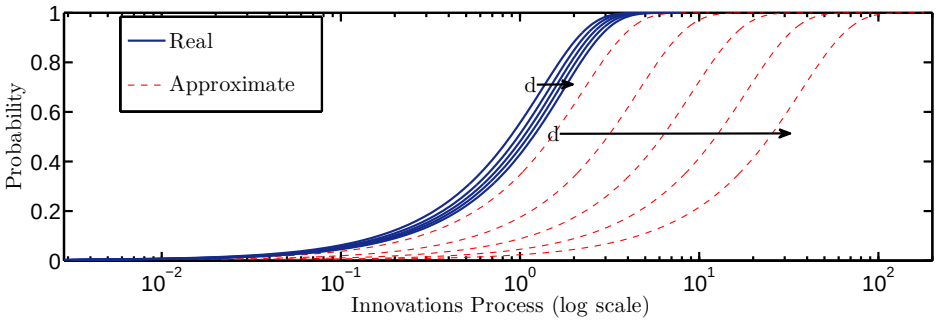


Figure 5.8: The approximate PDF in Lemma 5.6 is majorized by the actual PDF, as seen in this comparison of the CDFs. The approximate distribution has a larger variance, and is an upper bound for the actual variance.

Note that if $\mu > 1$, $p_{\gamma,d}$ increases exponentially with delay and the sequence of event probabilities $\{p_\gamma\}_1^\infty$ diverges. For $\mu < 1$, the decreasing probability law can be checked for Lyapunov mean square stability using Theorem 5.4.

5.5 Example

We now illustrate some of the results presented in this work. We begin with an illustration of the upper bound derived in Section 5.3.3. Our next example illustrates how the sufficient conditions for Lyapunov mean square stability, presented in Theorem 5.4, can be used to infer stability properties of the control system. Our third example illustrates the design of a constant-law scheduler that guarantees Lyapunov mean square stability. The final example illustrates the selection of event thresholds corresponding to a given design.

Example 5.2

[Illustration of Majorization] In Lemma 5.6, we find an approximating PDF $\hat{\phi}_{(I,d)}$, which is majorized by the real PDF $\phi_{(I,d)}$, for all delays $d > 0$. We illustrate this for a control system with parameters $A = 2$, $B = 1$, $\Sigma_w = 1$ and a constant event threshold $\Delta_d = 1$, for all $d > 0$. The CRM persistence probability is set to $p_\alpha = 1$, and the conditional probability of a busy channel is $p = 0.6$. For this setup, we compare the cumulative distribution function (CDF) corresponding to $\phi_{(I,d)}$, with the CDF corresponding to $\hat{\phi}_{(I,d)}$, for $d = \{1, \ldots, 5\}$, in Figure 5.8. The arrows indicate the increasing delays. Clearly, $\phi_{(I,d)} \succ \hat{\phi}_{(I,d)}$, for each of the five delays, according to Definition 5.5. This figure also illustrates why the estimation error covariance of the approximated PDF is greater than that of the real PDF.

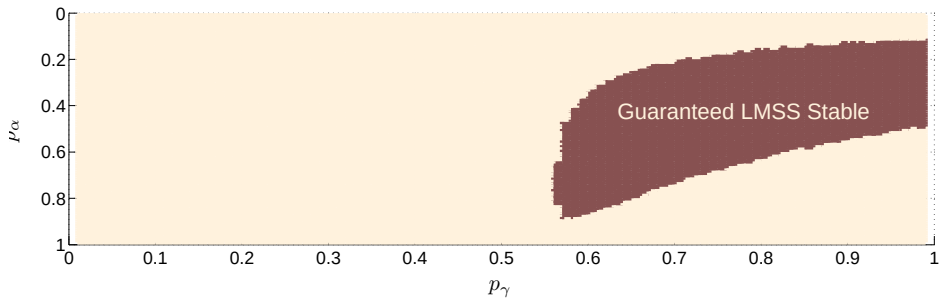


Figure 5.9: The shaded region denotes the set of event and persistence probabilities, p_γ and p_α, respectively, that guarantee Lyapunov mean square stability. We use the sufficient conditions in Theorem 5.8, for a constant-probability scheduler, to determine Lyapunov mean square stability.

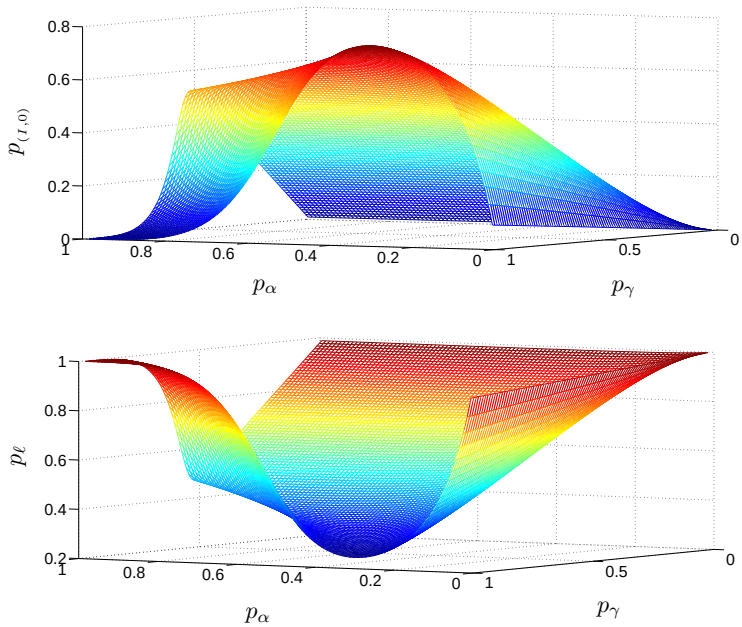


Figure 5.10: A surface plot of the network reliability $\pi_{(I,0)}$, and the probability of loss p_l, respectively, versus the event probability p_γ and the persistence probability p_α, for a constant-probability scheduler.

The auxiliary system was chosen to correspond to the worst-case evolution of the real system, with saturated network traffic. Thus, the upper bound is tighter for a large busy channel probability p and small state transition matrix A.

Next, we return to Example 5.1, where we illustrated that different network sizes result in different stability properties. We use our sufficient conditions for Lyapunov mean square stability to confirm the observed stability properties for two network sizes.

Example 5.3

[Checking for Lyapunov Mean Square Stability] We consider two network scenarios: case 1 corresponds to a network with $M = 2$ nodes and case 2 to a network with $M = 10$ nodes. The control systems in both network scenarios are identical to the systems described in Example 5.1, and so is the CRM. We use Theorem 5.4 to show that in case 1, Lyapunov mean square stability is achievable, and that in case 2, Lyapunov mean square stability cannot be guaranteed. This can be seen by using (5.15), where we see that the idle state probabilities must achieve a ratio of less than 0.5 for large d. Case 1 achieves a ratio of less than 0.1 for $d > 10$, whereas case 2 has a ratio of 0.98 even after $d = 50$. The Lyapunov mean square stability properties can be inferred from a trace of the state x as illustrated in Figure 5.2.

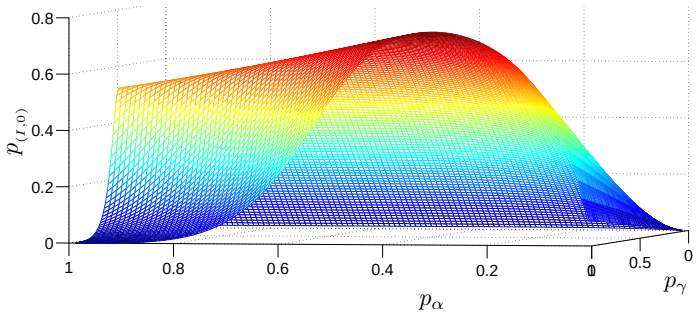
In the next example, we illustrate the design procedure outlined in Figure 5.4.

Example 5.4

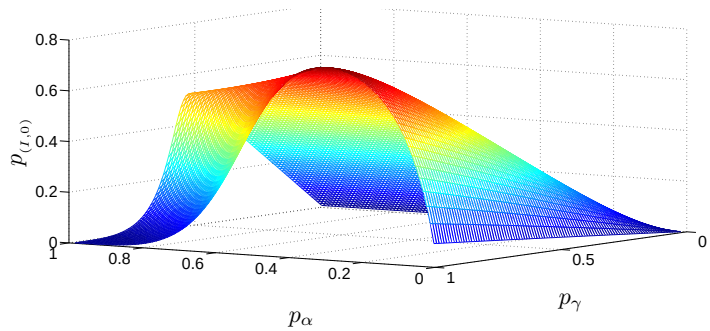
[Constant-Probability Policy] We consider a network of $M = 5$ control systems with identical parameters $A = 1.5$, $B = 1$ and $\Sigma_w = 1$, and $r_{\max} = 10$ retransmissions in the CRM. For simplicity, we assume that the persistence probability p_α is constant for all retransmission attempts. Using an algorithm similar to the one outlined in the flowchart in Figure 5.4, we obtain the set of event probabilities p_γ, and the set of persistence probabilities p_α, that result in Lyapunov mean square stability. The results are depicted in Figure 5.9. The shaded region in the figure corresponds to this set. In Figure 5.10, we present surface plots of the network reliability $\pi_{(I,0)}$ and the probability of loss p_l, respectively, versus p_γ and p_α. It is interesting to note the importance of jointly selecting p_γ and p_α.

Now, we compare the Lyapunov mean square stability regions obtained for the same network, but with control system parameters $A = 1.25$ and $A = 2$, i.e., less unstable and more unstable systems, respectively. The surface plots of the probability of successful transmission are shown in Figure 5.11(a) and Figure 5.11(b). The shaded regions in Figure 5.12(a) and Figure 5.12(b) denote the sets of event and persistence probabilities that guarantee Lyapunov mean square stability. Notice how the Lyapunov mean square stability region given by our sufficient condition in Theorem 5.4 shrinks as A increases.

We now illustrate how to select event thresholds for an event probability p_γ and persistence probability p_α chosen from the outcome of Example 5.4.



(a) Probability of Transmission Success, $A = 1.25$



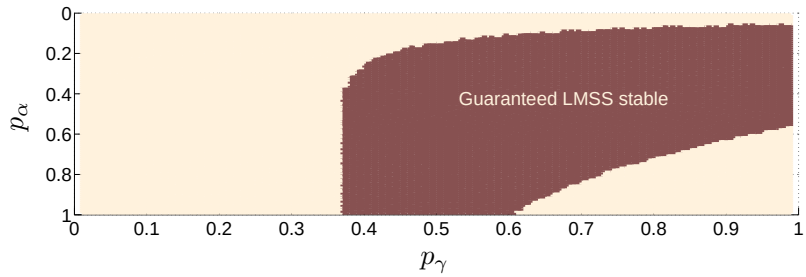
(b) Probability of Transmission Success, $A = 2$

Figure 5.11: Surface Plots of the probability of transmission success, for $A = 1.25$ (left) and $A = 2$ (right). The region guaranteed to be Lyapunov mean square stability by our sufficient conditions shrinks considerably for highly unstable control systems.

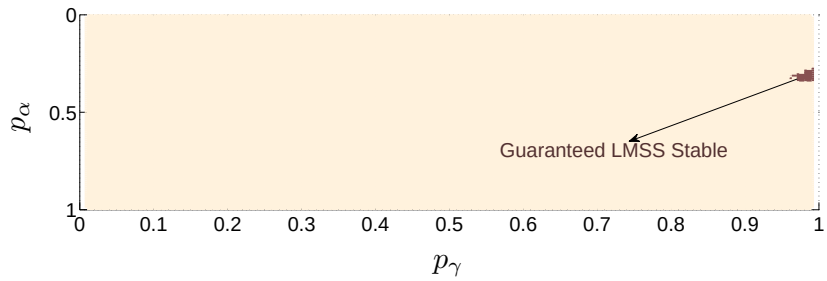
Example 5.5

[Selecting Event Thresholds] For the control system with parameters $A = 1.5$, , $B = 1$ and $\Sigma_w = 1$, choose $p_\gamma = 0.8$ and $p_\alpha = 0.4$. This choice of probabilities yields a network reliability of $\pi_{(I,0)} = 0.7056$ and a loss probability of $p_l = 0.2944$, from Figure 5.9 and Figure 5.10, respectively. The delay distribution for the constant-probability scheduler is easily seen to be given by $\mathrm{P}(d_k = d) = \pi_{(I,0)} \cdot p_l^d$, for any delay $d \geq 0$. The exponential delay distribution considerably simplifies our task. We now need to identify only a set of D event thresholds, where we choose D to be sufficiently larger than the smallest probability we wish to consider. In this example, we choose $D = 12$.

We now numerically compute event thresholds that give us the above event probabilities. To do this, we simulate the evolution of distributions described in (5.17)–(5.19), and assign the event thresholds as $\Delta_d := t : \int_{|x| \leq t} \phi_d dx = p_\gamma$, for all



(a) Probability of Loss, $A = 1.25$



(b) Probability of Loss, $A = 2$

Figure 5.12: Plots indicating the region of stability, for $A = 1.25$ (left) and $A = 2$ (right). The region guaranteed to be Lyapunov mean square stability by our sufficient conditions shrinks considerably for highly unstable control systems.

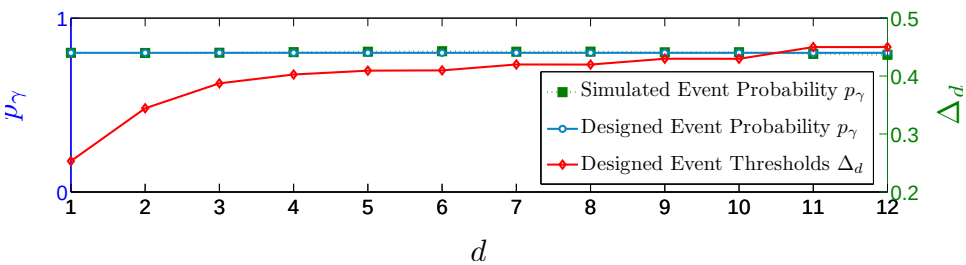


Figure 5.13: A comparison of analytical and simulated values of the event probability p_γ, plotted against the axis on the left. The numerically computed values of the event thresholds Δ_d are plotted against the axis on the right.

Table 5.14: A comparison of analytical and simulated values of p_γ

Event Threshold	Simulated Event Probability	Designed Event Probability
$\Delta_1 = 0.2533$	0.8000	0.8000
$\Delta_2 = 0.3444$	0.7992	0.8000
$\Delta_3 = 0.3874$	0.8006	0.8000
$\Delta_4 = 0.4027$	0.8042	0.8000
$\Delta_5 = 0.4093$	0.8068	0.8000
$\Delta_6 = 0.4098$	0.8095	0.8000
$\Delta_7 = 0.4199$	0.8067	0.8000
$\Delta_8 = 0.4198$	0.8071	0.8000
$\Delta_9 = 0.4299$	0.8030	0.8000
$\Delta_{10} = 0.4297$	0.8035	0.8000
$\Delta_{11} = 0.4500$	0.7943	0.8000
$\Delta_{12} = 0.4500$	0.7890	0.8000

$d \geq 0$. We present the event thresholds, thus identified, in Figure 5.5. To validate our design procedure, we run Monte Carlo simulations using the thresholds identified above, and confirm that the event probabilities we obtain are as desired, as shown in Figure 5.5. For delays larger than $D = 12$, the probabilities we obtain are not accurate, as there are too few instances of these events to result in a precise value.

5.6 Summary

We have presented a design for a network of event-based systems that ensures stability of each control system and of the entire network. Each control system uses a CRM to access the shared network. The event-triggering policy and the CRM sometimes result in congestion, and consequently packet losses and delays. To counter this, our design for the event-triggering policy guarantees Lyapunov mean square stability for each control system in the network. Our event probability designs were based on the stability analysis presented in this chapter.

With this, we conclude our presentation of event-based systems. In the next chapter, we look at another realization of a state-based channel access method.

Chapter 6

State-based Prioritized Access

In this chapter, we consider a state-based prioritized channel access mechanism. The prioritized approach presented here can be thought of as an alternative to the event-based approach presented so far. It is also an extension, in a sense, of the event-based approach. An event-triggering policy can be seen as a prioritization policy that generates two priority levels. It also chooses different channel access probabilities for each priority level, even if one of them is simply zero. We now extend this approach to a multilevel prioritization policy for a network of control systems. We present a design for a prioritization policy, along with an analysis of the resulting network and control performance.

Consider the scenario depicted in Figure 6.1, where a number of physical systems are monitored or controlled over a shared wireless network. This is typical of wireless sensing and actuating networks, where many sensors monitor physical systems and transmit the collected data to a Data Processing Unit (DPU) across a shared network. In sensor networks, the DPU may track the state of the physical system. In actuating networks, the DPU may issue a control signal to regulate the state of the physical system. In either case, the sensing link from each system to the DPU belongs to a shared network, as depicted in the figure. We wish to design a prioritized multiple access protocol for this network. Wireless actuating networks differ from sensor networks due to the consequences of delayed control action on critical dynamical processes (Walsh and Ye, 2001; Schenato et al., 2007). The communication infrastructure for such a network must meet harsher time constraints, in proportion to how far the state or measurement is from the nominal value. At the same time, the sensing paradigm continues to apply to actuating networks; implying that there are many nodes in the shared network. Thus, any multiple access protocol must be scalable with network size.

A prioritized access mechanism addresses most reliability concerns. In the absence of a prioritized access mechanism, a sensor containing critical information could be blocked by a sensor containing regular or non-critical measurements. Static priorities are easy to implement, but often inefficient. A fire detector in a chemical plant is likely to be assigned a higher priority than a temperature sensor monitor-

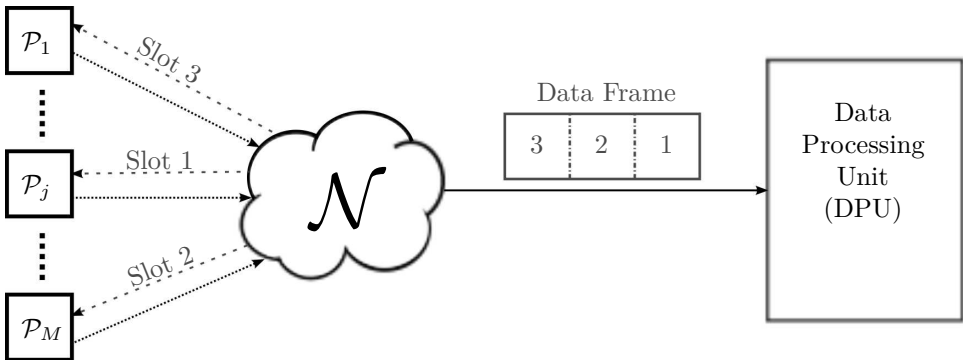


Figure 6.1: An overview of a multiple access network ($\mathcal{N}$) of plants ($\mathcal{P}_j$), $j \in \{1, \ldots, M\}$, with prioritized access in the MAC. The plant sensors communicate a priority to the network, which evaluates it distributedly, and suitably orders data packets in the frame.

ing operational levels, but the temperature measurements at most time instants are likely to contain more important information regarding operation of the plant as compared to routine (safe) measurements from the fire detectors. Dynamic priorities based on information in the current measurements solves the problem, but these are hard to assign and evaluate in a wireless network. A protocol that allows for prioritization of data based on the current measurement, in a distributed setting, would meet all the above requirements.

6.1 Contributions and Related Work

The main contribution of this chapter consists of a method of assigning measurement-based priorities. A node evaluates the criticality of the current measurement to be transmitted to the controller or monitoring unit and assigns an appropriate priority. The priority is a measure of the attention that a packet requires from the controller, and hence called the *attention factor*. These priorities are evaluated in a distributed manner, through a tournament. The throughput and delay of such an access mechanism are analyzed. This analysis is verified through simulations. We also find an upper-bound for the resulting estimation and control costs in the network.

In addition, we provide an implementation of the tournaments on low-rate wireless devices using IEEE 802.15.4 (IEEE, 2006) compliant radios. This is meant as proof-of-concept, to illustrate that the proposed mechanism is compatible with current radio transceivers, and can be implemented on current off-the-shelf hardware. We explore the throughput-tradeoffs in the implementation, and include these effects in our analysis. The protocol is implemented as part of the IEEE 802.15.4 standard (IEEE, 2006), which serves as the basis for Zigbee (ZigBee Alliance, 2005). We

also present experimental results from the KTH wireless testbed, with the proposed modifications to the protocol.

Previous attempts at introducing priorities within the carrier sense multiple access protocol with collision avoidance (CSMA/CA) include arbitration of interframe space or contention window differentiation, such as in IEEE 802.11e (Bianchi et al., 2005). This protocol still results in random access, but with prioritized access probabilities. It also does not allow for sufficient priority levels, as will be required when the priorities are based on the current measurement. In this chapter, we introduce a prioritized access scheme with transmission slots reserved for nodes that win a tournament. The idea of a tournament to resolve contention-based on static priorities is already prevalent in the literature, e.g., the CAN Bus Protocol (Robert Bosch GmbH, 1991) and its recent adaptation to wireless networks in WiDOM (Pereira et al., 2007). However, the priority mechanism in our proposal is dynamic, and priorities are assigned to data packets, not to nodes.

The idea of using the state or measurement of a physical system to determine channel access has been prevalent for some time now (Walsh et al., 1999; Otanez et al., 2002; Yook et al., 2002). The deviation in the state from the nominal value was used to determine a priority in Try-Once-Discard (TOD) (Walsh et al., 1999). Deadbands around the nominal value were used to limit the use of the channel by Otanez et al. (2002) and Yook et al. (2002). Both these ideas have given rise to many related works, which we discuss below.

Maximum error first is the prioritization principle used in TOD, to guarantee input-to-state stability for deterministic systems with disturbances. The implementation of the original idea was centralized, and required a network coordinator to collect and compare errors from the various physical processes in the network. This contention-free implementation has been extended to include effects of packet losses in Tabbara and Nesić (2008). Recently, a distributed implementation for this protocol has been conceived and successfully implemented in Christmann et al. (2014), but without evaluating the robustness of the implementation to information loss. In contrast, this chapter deals with state-based priorities for stochastic processes, where the priorities are allocated and evaluated in a distributed manner. The emphasis in this chapter is on performance, not just stability. Finally, the design and analysis methods presented in this chapter are shown to be easily extended to include packet losses from a real implementation.

A state-based priority can be viewed as an M-ary extension of binary-valued events. Consequently, many of the analytical results in this chapter build on results from the literature in stochastic event-triggered systems (Åström and Bernhardsson, 1999). Structural results for the closed-loop system that motivate the use of policies such as maximum-error-first have been explored in Molin and Hirche (2013) and in Chapter 3 of this thesis. The network interactions that result from state-based access methods introduce correlations between exogenous processes, as pointed out in Cervin and Henningsson (2008) and Rabi and Johansson (2009a). The state-based policy presented in this chapter circumvents these complications, while retaining the benefits of using the state to determine channel access. A related work from

the event-triggered literature is presented in Araujo et al. (2014), which explores a dynamic utilization policy for the Time Division Multiple Access (TDMA) slots of the IEEE 802.15.4 protocol.

The outline of the chapter is as follows. We formulate the problem in Section 6.2. We present the attention factor formulation and the tournament access mechanism in Section 6.3, and analyze the performance of this protocol in Section 6.3.3. We explain the implementation of tournaments in Section 6.4, motivate the choice of many parameters in the protocol. Finally, we illustrate the simulation and experimental results in Section 6.5.

6.2 Problem Formulation

We consider a network of M processes and their respective sensors, which communicate over a shared channel, as shown in Figure 6.1. Access to the network is determined by a state-based priority, and the priorities are evaluated in a distributed manner. Sensors that secure access transmit their packets in the chosen order. Each system in this network views the rest of the network through the model depicted in Figure 6.2. The blocks in this figure are explained below.

Plant: Each plant $\mathcal{P}_j$, for $j \in \{1, \ldots, M\}$ has state dynamics given by

$$x_{k+1}^j = Ax_k^j + Bu_k^j + w_k^j \,, \tag{6.1}$$

$$y_k^j = Cx_k^j + v_k^j \,, \tag{6.2}$$

where the state $x_k^j \in \mathbb{R}^n$ and the measurement $y_k^j \in \mathbb{R}^m$. The initial state x_0^j, the process noise w_k^j and the measurement noise v_k^j are i.i.d. zero-mean Gaussians with covariance matrices R_0, R_w and R_v, respectively. If the physical process is not part of a control loop, there is no control term in the state equation, i.e., $B = 0$. This discrete time model is defined with respect to a sampling period T for each plant, and the sampling instants are generated by a synchronized network clock. The different plants in the network are driven by exogenous noise processes. We comment on an extension to heterogeneous networks in Section 6.3.

Kalman Filter: A Kalman filter (KF) is implemented in each sensor to provide an estimate $\hat{x}_{k|k}^{s,j}$, where the superscript 's' denotes the estimator at the sensor, and is given by

$$\hat{x}_{k|k}^{s,j} = \hat{x}_{k|k-1}^{s,j} + K_{f,k}e_k^j \,,\ \hat{x}_{k|k-1}^{s,j} = A\hat{x}_{k-1|k-1}^{s,j} + Bu_{k-1}^j \,, \tag{6.3}$$

where $K_{f,k}$ denotes the Kalman gain, e_k^j denotes the innovation in the measurement, and $\hat{x}_{k|k-1}^{s,j}$ denotes the predicted estimate. The innovation is defined as

$$e_k^j = y_k^j - C\hat{x}_{k|k-1}^{s,j} \,. \tag{6.4}$$

The Kalman gain is defined as $K_{f,k} = P_{k|k-1}^s C^\top R_{e,k}^{-1}$, where $R_{e,k} = CP_{k|k-1}^s C^\top + R_v$ is the covariance of the innovation e_k^j. The prediction error covariance is given by

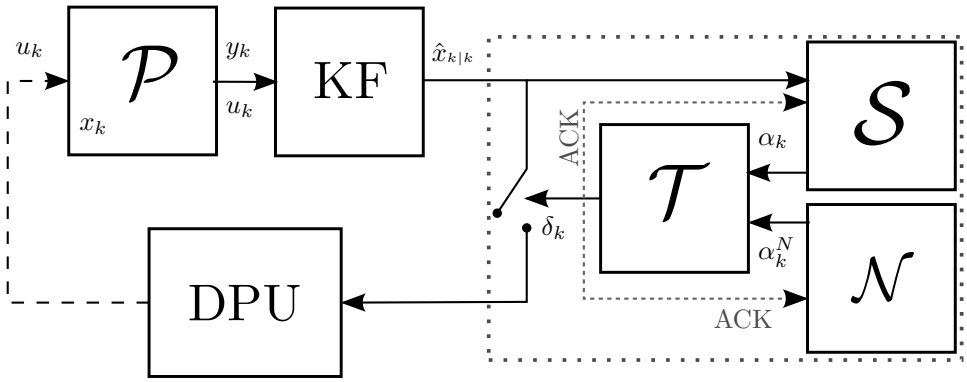


Figure 6.2: A mathematical model of a single system in the network. The system itself consists of a plant ($\mathcal{P}$), a Kalman filter (KF) at the sensor, a state-based priority synthesizer ($\mathcal{S}$) and a Data Processing Unit (DPU) across the network. The priority α_k is used to determine access to the shared network. The rest of the traffic sources in the network are abstracted away into the network block ($\mathcal{N}$), and this block produces an aggregate priority α_k^N, a vector of all the other priorities in the network. The priorities are evaluated in the Tournament block ($\mathcal{T}$), and $\delta_k \in \{0, 1\}$ indicates if the current packet is successfully transmitted across the network or not, respectively.

$P^s_{k|k-1} = AP^s_{k-1|k-1}A^\top + R_w$ and the filtered error covariance is given by $P^s_{k|k} = P^s_{k|k-1} - K_{f,k}R_{e,k}K^\top_{f,k}$.

State-based Priorities: There is a local scheduler $\mathcal{S}$, situated in the sensor node, between the plant and the network, which calculates the state-based priority, α^j_k, of the data packet. This block is formulated using a policy f, as given by

$$\alpha^j_k = f_k(\omega^{s,j}_k) \,, \tag{6.5}$$

where, $\omega^{s,j}_k \in \Omega^{s,j}_k$ and $\Omega^{s,j}_k$ is the σ-algebra generated by the information set at the scheduler, given by $\mathbb{I}^{s,j}_k = \left\{\{\hat{x}^{s,j}\}^{k|k}_{0|0}, \{y^j\}^{k-1}_0, \{\alpha^j\}^{k-1}_0, \{\delta^j\}^{k-1}_0, \{u^j\}^{k-1}_0\right\}$. The notation $\{c\}^b_a := \{c_a, \ldots, c_b\}$, for $a \leq b$.

Network: The network $\mathcal{N}$ generates other traffic, with an aggregate priority $\alpha^{N,j}_k$, which denotes a vector of all the other priorities in the network.

Tournament Block: The tournament $\mathcal{T}$ resolves contention between multiple simultaneous channel access requests. The channel access indicator $\delta^j_k \in \{0, 1\}$ is given by

$$\delta^j_k = \mathcal{T}(\alpha^j_k, \alpha^{N,j}_k) \,. \tag{6.6}$$

DPU: The DPU receives $z^j_k = \delta^j_k \hat{x}^{s,j}_{k|k}$, and utilizes the estimate of the state $\hat{x}^{s,j}_{k|k}$, when it receives it, in monitoring, control or detection applications. With no loss of generality, we assume that the DPU contains an estimator followed by a controller. The estimate at the DPU is denoted $\hat{x}^{c,j}_{k|k}$, where the superscript 'c' indicates the

controller, and is given by

$$\hat{x}^{c,j}_{k|k} = \delta^j_k \hat{x}^{s,j}_{k|k} + (1-\delta^j_k)\hat{x}^{s,j}_{k|\tau_k} \ , \tag{6.7}$$

where τ_k is the time index of the last received packet. The estimation cost is given by the average estimation error variance at the DPU,

$$J_E = \mathsf{tr}\{\mathbb{E}[P^{c,j}_{k|k}]\} \ , \tag{6.8}$$

where $P^{c,j}_{k|k}$ is the estimation error covariance at the controller and $\mathsf{tr}\{\cdot\}$ is the trace operator.

The controller implements a policy g_k, defined on the σ-algebra generated by the information set of the controller, given by $\mathbb{I}^{c,j}_k = \left\{\{z^j\}^k_0, \{\delta^j\}^k_0, \{u^j\}^{k-1}_0\right\}$. The control policy is typically chosen to minimize the infinite horizon Linear Quadratic Guassian (LQG) control cost, given by

$$J_C = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} \mathbb{E}\left[(x^j_k)^\top Q_1 x^j_k + (u^j_k)^\top Q_2 u^j_k\right] \ , \tag{6.9}$$

where the weighting matrices, Q_1 and Q_2, are non-negative and positive, respectively.

In this chapter, we present a formulation for the prioritization policy f_k and the tournament mechanism $\mathcal{T}$. We identify properties of the resulting access mechanism, and provide an analytical expression for the probability of a successful transmission $\mathrm{P}(\delta^j_k = 1)$. We also characterize the estimation and control costs, J_E and J_C, respectively. In addition to the above theoretical investigations, we design a modification to the existing IEEE 802.15.4 standard, to include tournaments, and implementing the resulting protocol on state-of-the-art low-rate wireless devices with IEEE 802.15.4 compliant radios. We validate the analysis using a wireless networked control experimental setup.

6.3 Protocol Design and Analysis

We now present the main theoretical results in this chapter. We begin by introducing a formulation for the priorities, and identifying some of its properties. Then, we present tournaments as a mechanism to evaluate priorities in a distributed manner. We also present a network-level performance analysis of the resulting access mechanism. We use the results of this analysis to evaluate upper bounds for the estimation and control costs for systems that use this access mechanism.

6.3.1 Attention Factor

In this section, we derive an expression for the state-based priorities and in doing so, identify the scheduling policy f_k in (6.5). The priorities are assigned by each sensor node to its own data packet, in isolation from the rest of the network.

The attention factor is an adaptive priority designed to call the attention of the DPU to the current data in the node, and to connote a penalty in not being able to transmit this data. At some time $k-1$, the j^{th} sensor delivers a measurement y^j_{k-1} to its local estimator (KF), which computes the estimate $\hat{x}^{s,j}_{k-1|k-1}$. Let us assume that the node is successful in sending this estimate to the DPU over the network. The estimator at the DPU (6.7) can generate future estimates as $\hat{x}^{c,j}_{k|k} = \hat{x}^{s,j}_{k|k-1}$,$\hat{x}^{c,j}_{k+1|k+1} = \hat{x}^{s,j}_{k+1|k-1}$, and so on. The motivation for allocating channel resources to the sensor in order to deliver the next packet to the DPU, is the innovation e^j_k in the measurement y^j_k, given by (6.4). The risk in not being able to deliver this packet can be assigned a value equal to a function of the difference between the predicted estimates $\hat{x}^{s,j}_{k+1|k-1} = A\hat{x}^{s,j}_{k|k-1} + Bu^j_k$ and $\hat{x}^{s,j}_{k+1|k} = A\hat{x}^{s,j}_{k|k} + Bu^j_k$. In fact, the quantity $\hat{x}^{s,j}_{k+1|k-1} - \hat{x}^{s,j}_{k+1|k}$, is an indicator of the minimum risk in not delivering the current packet, as the risk will only be larger if the packet at time $k-1$ did not reach the DPU. We have used the predicted estimate in the formulation so far to emphasize process dynamics and make the attention factor sensitive to unstable systems. To further emphasize the dynamics, the prediction horizon need only be extended further more.

Since we deal with estimation and control in this chapter, let us look at a distortion-like function of the quantity described above.

Definition 6.1 (Minimum Risk Indicator)**.** *The increase in the sample variance of the prediction error due to not receiving a packet at time k is denoted $\Delta P^j_{samp,k}$ and given by*

$$\Delta P^j_{samp,k} = f_{MRI}(y^j_k) := \mathit{tr}\{(\hat{x}^{s,j}_{k+1|k} - \hat{x}^{s,j}_{k+1|k-1}) \cdot (\hat{x}^{s,j}_{k+1|k} - \hat{x}^{s,j}_{k+1|k-1})^\top\} \,. \tag{6.10}$$

$\Delta P^j_{samp,k}$ is an empirical quantity based on knowledge of the measurement y^j_k, and can be rewritten as $\Delta P^j_{samp,k} = \mathit{tr}\{AK_{f,k}e^j_k(e^j_k)^\top K^\top_{f,k}A^\top\}$.

A prioritized transmission scheme based on the quantity $\Delta P^j_{\text{samp},k}$ should ideally result in

$$\delta^j_k = \mathcal{T}_{\text{ideal}}(\{\Delta P^i_{\text{samp},k}\}^M_{i=1}) := \begin{cases} 1 & j = \arg\max_i \Delta P^i_{\text{samp},k} \,, \\ 0 & \text{otherwise} \,, \end{cases} \tag{6.11}$$

for the j^{th} sensor node. Thus, the node with the largest priority gets to transmit. We use this idealized tournament to inspect the merits of the prioritization mechanism. First, note that the expected value of the minimum risk indicator can be shown to be proportional to the minimum increase in the variance of the prediction error at the DPU due to not possessing information about the measurement y^j_k, i.e.,

$$\mathbb{E}[\Delta P^j_{\text{samp},k}] = \mathsf{tr}\{AK_{f,k}\,\mathbb{E}[e^j_k(e^j_k)^\top]K^\top_{f,k}A^\top\} = \mathsf{tr}\{P^s_{k+1|k-1} - P^s_{k+1|k}\} \,.$$

Thus, the quantity $\Delta P^j_{\text{samp},k}$ can be said to be proportional to the increase in the per sample variance of the prediction error at the DPU. Then, let $\Delta P_{\text{sys},k} \triangleq$

$\sum_{j=1}^{M} \Delta P^{j}_{\text{samp},k}$ be the net increase in the sample variance of the prediction error due to not possessing information about measurements $\{\boldsymbol{y_k^j}\}_{j=1}^{M}$ from all the nodes in the network at time k. Using $\Delta P_{\text{sys},k}$ as an optimization index, we find that the resulting channel access mechanism is an optimal scheduling policy in a resource constrained network. In addition, the prioritization mechanism results in a few desirable system-level properties, such as separation between designs of the scheduler, observer and controller, and a minimum mean-squared error (MMSE) estimate with a simple, Kalman filter-like recursive update. All of these are compiled in the lemma below.

Proposition 6.1 (Properties of the Minimum Risk Indicator)**.** *For the network of M systems given by (6.1)–(6.6), with $f_k = f_{MRI}(y_k^j)$ for $1 \le j \le M$ as in (6.10) and $\mathcal{T} = \mathcal{T}_{ideal}(\{\Delta P^{i}_{samp,k}\}_{i=1}^{M})$ as in (6.11), it holds that:*

i. The resulting scheduling policy minimizes $\Delta P_{sys,k}$.

ii. The estimate (6.7) minimizes the mean-squared estimation error.

iii. The DPU estimates are correlated, but the network traffic remains independent.

iv. The optimal control policy for a closed-loop system in this network is certainty equivalent.

Proof. To prove Claim i, let us define the ratio $\eta_k^j \triangleq \Delta P^{j}_{\text{samp},k} / \Delta P_{\text{sys},k}$. In the prioritized access scheme in (6.11), the channel is allotted to the data packet which maximizes $\Delta P^{j}_{\text{samp},k}$. This also maximizes η_k^j, which implies that

$$\max_j \ \eta_k^j = \max_j \ \frac{\mathsf{tr}\{(\hat{x}^{s,j}_{k+1|k} - \hat{x}^{s,j}_{k+1|k-1}) \cdot (\hat{x}^{s,j}_{k+1|k} - \hat{x}^{s,j}_{k+1|k-1})^\top\}}{\sum_{j=0}^{M} \mathsf{tr}\{(\hat{x}^{s,j}_{k+1|k} - \hat{x}^{s,j}_{k+1|k-1}) \cdot (\hat{x}^{s,j}_{k+1|k} - \hat{x}^{s,j}_{k+1|k-1})^\top\}} \ .$$

Thus, the data packet (measurement) which results in a maximum reduction of the sample variance of the prediction error is allotted channel access. Thus, priorities based on $\Delta P^{j}_{\text{samp},k}$ result in a sample-wise optimal scheduling strategy given limited communication resources.

We now prove Claim ii. At any time k, let τ_k denote the time index of the last received packet by the j^{th} controller. Then, the MMSE estimate $\mathbb{E}[x_k^j | \mathbb{I}_k^{c,j}]$ is given by

$$\begin{aligned}\mathbb{E}[x_k^j | \mathbb{I}_k^{c,j}] &= \underbrace{A^{k-\tau_k} \hat{x}^{s,j}_{\tau_k|\tau_k} + \sum_{n=1}^{k-\tau_k} A^{n-1} B u^j_{k-n}}_{\hat{x}^{s,j}_{k|\tau_k}} + \mathbb{E}[x_k^j - \hat{x}^{s,j}_{k|\tau_k} | \{\boldsymbol{\delta^j}\}_{\tau_k+1}^{k} = 0] \\ &= \mathbb{1}_{\{\delta_k^j = 1\}} \hat{x}^{s,j}_{k|k} + \mathbb{1}_{\{\delta_k^j = 0\}} \hat{x}^{s,j}_{k|\tau_k} = \hat{x}^{c,j}_{k|k} \ ,\end{aligned}$$

where the second expression above is obtained by virtue of the symmetric prioritization policy. The term $\mathbb{E}[x_k^j - \hat{x}_{k|\tau_k}^{s,j} | \{\boldsymbol{\delta^j}\}_{\tau_k+1}^k = 0]$ vanishes when $\delta_k^j = 0$ due to the symmetric function of the innovations chosen for f_{MRI} in (6.10). Thus, the estimate in (6.7) is indeed the MMSE estimate.

To see how Claim iii holds, note that $\mathbb{E}[x_k^j | \mathbb{I}_k^{c,j}]$ is determined by δ_k^j, the channel access indicator. Since only a fixed number of nodes can access the channel, the variables $\{\delta_k^j\}_{j=1}^M$ are correlated. Thus, the network interaction induces a correlation in the estimates $\mathbb{E}[x_k^j | \mathbb{I}_k^{c,j}]$ across the network. However, the network traffic is determined by the priorities assigned to the nodes, which are functions of the independent innovations process. Thus, our choice of f_{MRI} in (6.10) results in independent traffic, despite the network interactions.

The choice of the scheduling policy f_k at each node is a symmetric function of the innovations process, and thus, the closed-loop system is free of a dual effect (from Chapter 3). This implies that the optimal control policy for a linear quadratic cost is certainty equivalent, as stated in Claim iv. □

Thus, our choice of the prioritization policy f_{MRI} results in many desirable properties, both for the network and the system. Importantly, f_{MRI} results in independent network traffic due to its dependence on the innovations e_k^j alone, simplifying the analysis and design considerably. A prioritization policy based on the *entire* prediction error, $(\hat{x}_{k+1|k}^{s,j} - \hat{x}_{k+1|\tau_k}^{s,j})$, rather than on e_k^j alone, might well result in better estimation and control costs. However, analyzing and designing such a scheme for a network of systems would certainly be more complex, and we comment on this later. Furthermore, our optimization index $\Delta P_{\text{sys},k}$ is a greedy index, emphasizing the current increase in per sample variance rather than the expected increase. Nevertheless, such a choice can still result in considerable benefits in terms of the estimation and control performance, as indicated by Theorem 6.8.

The tournament mechanism $\mathcal{T}_{\text{ideal}}$ is impractical as it compares real-valued priorities. To implement a prioritized access scheme, we require a fixed resolution priority. We obtain this by quantizing the minimum risk indicator in (6.10) to give us the attention factor, denoted as α_k^j.

Definition 6.2 (Attention Factor)**.** *For a dynamical system given by (6.1), the state-based priority $\alpha_k^j \in \mathbb{Z}$ for $1 \leq j \leq M$ is defined as*

$$\alpha_k^j = \textit{round}\left(\textit{tr}\{A K_{f,k} e_k^j (e_k^j)^\top K_{f,k}^\top A^\top\} \cdot \frac{A_{max}}{P_{s,max}} \right) , \tag{6.12}$$

where $0 \leq \alpha_k \leq A_{max}$ for some integer $A_{max} \in \mathbb{Z}$, and $P_{s,max} = \textit{tr}\{\kappa K_{f,k} R_{e,k} K_{f,k}^\top \kappa^\top\}$ is the maximum tolerable increase, as defined by the tolerance parameter $\kappa \in \mathbb{Z}^+$, in sample variance of the prediction error for a node attached to a plant with identity system matrix ($A = I$).

The tolerance parameter κ is used by the sensor node to dictate its own tolerance limits and influence the increase of α with deviating measurements. This

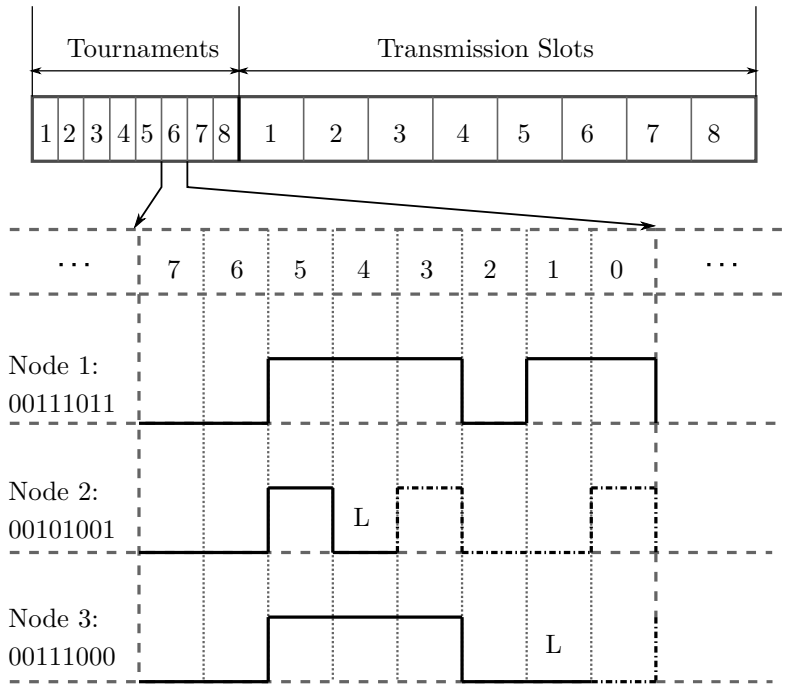


Figure 6.3: The Frame Structure of the tournament access mechanism, with a tournament comprising of 8 priority bits, for each of the $N_{\mathrm{T}} = 8$ transmission slots. An example of a tournament is depicted here, with three nodes competing for the sixth transmission slot. Nodes 2 and 3 lose the tournament to node 1, and the symbol L depicts the priority bit during which the tournament is lost. Node 1 gets to transmit its data packet in the sixth transmission slot.

constant determines the probability mass function (PMF) of the attention factor for the entire network, and we comment on its selection in later sections. Note that the merits of a prioritization scheme based on $\Delta P^{j}_{\mathrm{samp},k}$ (Proposition 6.1) are unaffected by the transformation to α^{j}_{k} for sufficiently large values of $A_{\max}$. Thus, (6.12) determines our choice of f_k in (6.5) at each node.

6.3.2 Evaluating Priorities Using Tournaments

Now that adaptive priorities have been assigned to the data packets, there remains the task of designing an arbitration policy to resolve contention. In other words, how should the data packets exchange priorities and decide who gets to transmit? We use tournaments as a distributed access mechanism that evaluates priorities in a non-coordinated manner and identify the function $\mathcal{T}$ in (6.6) corresponding to this mechanism.

Consider the frame structure presented in Figure 6.3. There are N_{T} transmission slots in each frame, and sensors that wish to transmit in this frame must generate an attention factor as described in Section 6.3.1. The data frame begins with N_{T} tournaments, followed by the same number of transmission slots. To transmit in one of the slots in this frame, a node must win the corresponding tournament. During the tournament, qualifying packets transmit their attention factors, starting with the most significant bit. Nodes transmit a suitably chosen pulse for a bit of value one and remain silent during the zero bit. As wireless transceivers cannot transmit and receive at the same instant, nodes can listen during the zero bits. A busy channel indicates that they have lost the tournament. The packet(s) with the largest attention factor wins the tournament.

As the attention factors are assigned by each node, more than one packet can have the same attention factor and win the tournament. Multiple winners are not aware of each other, and cause a collision. Using the same mechanism as in CSMA/CA, nodes are aware of a collision by the lack of an acknowledgment (ACK). Figure 6.3 illustrates the concept of a tournament between three nodes with attention factors of 59, 41 and 56 respectively. Nodes 2 and 3 lose the tournament after transmitting the fourth and first bits of their priorities, respectively, as they hear a busy channel during their recessive bits. Node 1 wins the tournament and transmits in the corresponding slot.

Thus, using this access mechanism, a node can either lose a tournament or win a tournament. A node that wins a tournament can either collide in the transmission slot or successfully transmit its data. We now define the tournament function $\mathcal{T}(\alpha_k, \alpha_k^N)$.

Definition 6.3 (Tournament Access Mechanism)**.** *Let the N_T highest values of attention factors, selected from the attention factors of all the nodes in the network, $\alpha_k^1, \ldots, \alpha_k^M$, be given by the set $\mathcal{A}_k = \{\bar{\alpha}_1, \ldots, \bar{\alpha}_{N_T}\}$, where $\bar{\alpha}_1 > \bar{\alpha}_2 > \cdots > \bar{\alpha}_{N_T}$. Each of these values may be the attention factor corresponding to one or more nodes in the network. It is clear that the values in the set $\mathcal{A}_k$ win the tournament, but if any of these values occur more than at one node, then, the corresponding packets are lost due to a collision. Thus, only the attention factors corresponding to unique values from the set $\mathcal{A}_k$ succeed in transmitting a packet. Let us denote the set of nodes that win a tournament at time k by $\mathfrak{W}_k$, and the set of nodes that successfully transmit in a tournament by $\mathfrak{T}_k$. We write these sets as*

$$\begin{aligned} \mathfrak{W}_k &:= \{j : \alpha_k^j \in \mathcal{A}_k\} \,, && \text{for } 1 \leq j \leq M \,, \\ \mathfrak{T}_k &:= \{j : \alpha_k^j \neq \alpha_k^s \;\forall\; s \in \mathfrak{W}_k \setminus j\} \,, && \text{for } j \in \mathfrak{W}_k \,. \end{aligned} \tag{6.13}$$

If $\mathfrak{A} = \{1, \ldots, M\}$, then clearly, the set of nodes that lose a tournament is given by $\mathfrak{L}_k := \mathfrak{A} \setminus \mathfrak{W}_k$, and the set of nodes that collide in a tournament is given by $\mathfrak{C}_k := \mathfrak{W}_k \setminus \mathfrak{T}_k$. Thus, the tournament function is given by

$$\mathcal{T}(\alpha_k^j, \alpha_k^{N,j}) = \begin{cases} 1 & j \in \mathfrak{T}_k \\ 0 & \text{otherwise} \,. \end{cases} \tag{6.14}$$

6.3.3 Analysis of the Tournament Access Mechanism

We now analyze the performance of a node that uses the attention factor in the tournament access mechanism. There are two performance metrics we identify - the probability of a successful transmission, and the distribution of the delay since the last successful transmission. These expressions will come of use in later sections to evaluate the estimation/control performance of each system in the network. Our analysis is presented for a homogenous network of M systems, but it will be shown that this analysis extends to networks with different types of processes. The analysis presented below depends on the statistical description of the system variables, which are identical in a homogenous network. Thus, we skip the index j in this section.

An important property of the attention factor is that we can find the PMF of each node's attention factor. For the general model described in Section 6.2, the innovations $e_k \in \mathbb{R}^m$ may be a vector. The sum of unnormalized squared Gaussian variables ($\mathsf{tr}\{AK_{f,k}e_k e_k^\top K_{f,k}^\top A^\top\}$) with unequal variances has a multivariate Gamma-type distribution, as discussed in Krishnamoorthy and Parthasarathy (1951). Then, the PMF is given by

$$\mathrm{P}(\alpha_k = a) = \begin{cases} \Phi(0.5\frac{P_{\mathrm{s,max}}}{A_{\max}}) & a = 0 \\ \Phi((\alpha+0.5)\frac{P_{\mathrm{s,max}}}{A_{\max}}) - \Phi((\alpha-0.5)\frac{P_{\mathrm{s,max}}}{A_{\max}}) & 0 < a < A_{\max} \\ 1 - \Phi((A_{\max}-0.5)\frac{P_{\mathrm{s,max}}}{A_{\max}}) & a = A_{\max} \end{cases}, \quad (6.15)$$

where, $\Phi(\cdot)$ is the cumulative distribution function corresponding to the multivariate Gamma-type distribution of $\mathsf{tr}\{AK_{f,k}e_k e_k^\top K_{f,k}^\top A^\top\}$. If $e_k \in \mathbb{R}$, or if $\mathbb{E}[e_k e_k^\top] = \sigma^2 I$, a diagonal matrix with equal values on the diagonal, then the above distribution becomes a first-order or higher-order Chi-Squared distribution, respectively.

It is easy for each node to characterize the probability of another node in the network with attention factor less than (p_{L}), less than or equal to (p_{LE}) and greater than (p_{G}) itself. In a homogenous network, these quantities are given by

$$p_{\mathrm{L}}(\alpha) = \sum_{a<\alpha} \mathrm{P}(a)\ , \quad p_{\mathrm{LE}}(\alpha) = \sum_{a\leq\alpha} \mathrm{P}(a)\ , \quad p_{\mathrm{G}}(\alpha) = 1 - p_{\mathrm{LE}}(\alpha)\ . \quad (6.16)$$

Let us denote $p_{\mathrm{C},n}(\alpha)$ as the probability that n nodes with attention factors greater than α collide in a single slot. This can be found as

$$p_{\mathrm{C},n}(\alpha) = \sum_{a=\alpha+1}^{A_{\max}-1} (\mathrm{P}(a))^n\ . \quad (6.17)$$

A combinatorial analysis using the above quantities gives us the probability of a node winning, successfully transmitting, losing or colliding in a tournament comprising of a number of tournament slots. We now present these results.

Theorem 6.2. *For a homogenous network of M systems given by (6.1)–(6.6), with f_k as given in (6.12) and $\mathcal{T}$ as defined in (6.14), the probability of successful*

transmission in N_T tournament slots for each node is given by

$$p_{\mathfrak{T}} := \mathrm{P}(\delta_k = 1) = \sum_{\alpha_k} \mathrm{P}(T_{N_T,M-1}|\alpha_k)\,\mathrm{P}(\alpha_k)\;, \tag{6.18}$$

where $\mathrm{P}(T_{N_T,M-1}|\alpha_k)$ is the conditional probability of a node with attention factor α_k succeeding in transmission in N_T slots against $M-1$ other nodes.

Proof. Let $\mathrm{P}(W_{N_\mathrm{T},M-1}|\alpha)$ denote the conditional probability of a node with attention factor α winning a tournament in N_T slots against $M-1$ other nodes. It is given by

$$\begin{aligned}\mathrm{P}(W_{N_\mathrm{T},M-1}|\alpha) \approx &\sum_{n=0}^{N_\mathrm{T}-1} C_n^{M-1} p_\mathrm{G}^n(\alpha) p_\mathrm{LE}^{M-1-n}(\alpha) \\ &+ \sum_{n=N_\mathrm{T}}^{M-1} C_n^{M-1}\Bigg(\sum_{r=n-N_\mathrm{T}+2}^{n} C_r^n p_\mathrm{G}^{n-r}(\alpha) p_{\mathrm{C},r}(\alpha)\Bigg) p_\mathrm{LE}^{M-1-n}(\alpha)\;,\end{aligned} \tag{6.19}$$

where $C_k^n = \frac{n!}{(n-k)!k!}$ refers to the binomial coefficient. The first term in this equation states that there can be only up to $N_\mathrm{T}-1$ packets with attention factors greater than α and that the rest must have attention factors less than or equal to α. This term does not take into account any collisions that might have occurred. The second term computes the probability of winning a tournament in N_T slots, given that one collision has occurred before the node with attention factor α gets to transmit. Thus, there can be more than $N_\mathrm{T}-1$ packets with attention factors greater than α, as long as sufficient numbers of these additional packets collide in the same slot. The exact expression for the quantity $\mathrm{P}(W_{N_\mathrm{T},M-1}|\alpha)$ must contain $N_\mathrm{T}-2$ additional terms, corresponding to $\{2,\ldots,N_\mathrm{T}-1\}$ collisions that occur before the node with attention factor α gets to transmit. However, these terms can be neglected when a collision is sufficiently rare.

The conditional probability of losing tournaments in all N_T slots against $M-1$ packets is then given by $\mathrm{P}(L_{N_\mathrm{T},M-1}|\alpha) = 1 - \mathrm{P}(W_{N_\mathrm{T},M-1}|\alpha)$. The conditional probability of succeeding in transmission in N_T slots against $M-1$ packets is given by

$$\begin{aligned}\mathrm{P}(T_{N_\mathrm{T},M-1}|\alpha) \approx &\sum_{n=0}^{N_\mathrm{T}-1} C_n^{M-1} p_\mathrm{G}^n(\alpha) p_\mathrm{L}^{M-1-n}(\alpha) \\ &+ \sum_{n=N_\mathrm{T}}^{M-1} C_n^{M-1}\Bigg(\sum_{r=n-N_\mathrm{T}+2}^{n} C_r^n p_\mathrm{G}^{n-r}(\alpha) p_{\mathrm{C},r}(\alpha)\Bigg) p_\mathrm{L}^{M-1-n}(\alpha)\;,\end{aligned} \tag{6.20}$$

which differs from (6.19) by requiring that the other packets have attention factors strictly less than the value α. Finally, the conditional probability of a collision under

these circumstances is given by $\mathrm{P}(C_{N_\mathrm{T},M-1}|\alpha) = \mathrm{P}(W_{N_\mathrm{T},M-1}|\alpha) - \mathrm{P}(T_{N_\mathrm{T},M-1}|\alpha)$. We can then define the probability of a successful transmission ($p_{\mathfrak{T}}$) for each node in the network using (6.18). □

We can now use this expression to characterize delay and throughput. We define the delay d_k suffered by a node as the number of sampling periods since the last successful transmission. Let τ_k denote the time-index of the last successful transmission. This can be defined as $\tau_k := \max\{t : \delta_t = 1\}$, for $-1 \leq t \leq k$ and $\delta_{-1} = 1$. Then, it follows that $d_k := k - \tau_k$.

Corollary 6.3. *For a homogenous network of M systems given by (6.1)–(6.6), with f_k as given in (6.12) and $\mathcal{T}$ as defined in (6.14), the probability distribution of the delay d_k for each node is given by*

$$\mathrm{P}(d_k = d) = p_{\mathfrak{T}}(1 - p_{\mathfrak{T}})^{d-1} \,. \tag{6.21}$$

Proof. The attention factor α_k is based on the innovations e_k, which is a white process. The results of Theorem 6.2 show that the outcomes of the tournaments can be treated as independent. Now, we use the fact that a successful transmission after a delay of d sampling instants implies a transmission failure for $d-1$ sampling instants. Using the expression for $p_{\mathfrak{T}}$ from Theorem 6.2, we obtain (6.21). □

We define throughput as the fraction of time that useful information is carried on the network. This is a measure of the efficiency of our multiple access mechanism.

Corollary 6.4. *For a homogenous network of M systems given by (6.1)–(6.6), with f_k as given in (6.12) and $\mathcal{T}$ as defined in (6.14), the throughput S_T is given by*

$$S_T \approx \frac{N_T \cdot Len(P)}{T_T} \,, \tag{6.22}$$

where $Len(P)$ is the packet payload size and T_T is the length of the access period.

Proof. For $M > N_\mathrm{T}$ and a negligible probability of collision, the definition yields the expression in (6.22). □

Extension to Heterogenous Networks: The combinatorial analysis performed above can still be used when the network is heterogenous. If the network contains two different types of physical processes with parameters A_i, B_i, etc. in (6.1) for $i = \{1, 2\}$, then the analysis requires two types of probabilities for each of the expressions given in (6.16)–(6.17). The expressions in (6.19)–(6.20) must be modified to include combinations of these two types of probabilities. Thus, the basic idea behind the analysis remains the same, but the number of cases to be taken into account are larger for heterogenous networks.

6.3.4 Estimation and Control Performance

We now make use of the performance analysis results of the previous section in order to evaluate the estimation and control performance. In this section as well, we skip the index j for the same reason as before. We begin with the estimation cost J_E in (6.8).

In the tournament access mechanism, the DPU can gain information about the statistics of the state even if the data packet is not received. This is because the conditional probabilities of losing or colliding in a tournament alter the probability density of the innovations. The altered probability density of the innovations can be written as

$$\psi(\bar{e}_k|\delta_k = 0) = \sum_{\alpha_k=0}^{A_{\max}-1} \psi(\bar{e}_k|\alpha_k) \frac{(1 - \mathrm{P}(T_{N_\mathrm{T},M-1}|\alpha_k))}{1 - p_{\mathfrak{T}}} \,, \tag{6.23}$$

where $\bar{e}_k = AK_{f,k}e_k$ and $\psi(\cdot)$ is the probability density function of $\bar{e}_k$, which is a multivariate normal distribution. Thus, the innovations no longer appear to be normally distributed to the DPU. In fact, if the DPU can 'listen' to the values of the priorities broadcast during the tournaments, then it has even more information to obtain a better posterior description of the innovations.

The posterior mean of the innovations cannot be improved in the above manner, due to the choice of a symmetric policy f_k in (6.12). However, the posterior variance of the innovations, and consequently of the estimation error, is altered. Finding an exact expression for this variance is difficult, as the density in (6.23) is non-Gaussian. Instead, we find an upper bound for the variance. The upper bound is the variance of the estimation error corresponding to a Bernoulli packet loss process with loss probability $(1 - p_{\mathfrak{T}})$. To see why, note that we can design the tournament access mechanism to ensure that the conditional probability of transmitting a packet is an increasing function of the attention factor. Then, the conditional probability of transmission is a non-decreasing function of the magnitude of the innovations. This will naturally result in a lower variance than a uniform probability of transmission across all attention factors. We show this formally below, beginning with the following property that reflects our design choice.

Property 6.4 (Non-increasing tail of the attention factor distribution)**.** *The distribution of the attention factor,* $\mathrm{P}(\alpha_k = a)$*, is said to possess the* non-increasing tail property *if* $\mathrm{P}(\alpha) \leq \mathrm{P}(\alpha - 1)$ *for all* $\alpha > 0$*.*

This is not a surprising property to obtain from a Gaussian innovations process, if it were not for the finite range of the attention factor. As the $\mathrm{P}(\alpha_k = A_{\max} - 1)$ in (6.15) is equal to the probability of the tail of the distribution, this value might not confirm with the above property. However, there is always a suitable value of the tolerance parameter κ in (6.12), for which the above property can be made to hold. Furthermore, such a value of κ is indeed a desirable design for the attention factor, as it reduces the probability of a collision for the highest value of attention factor.

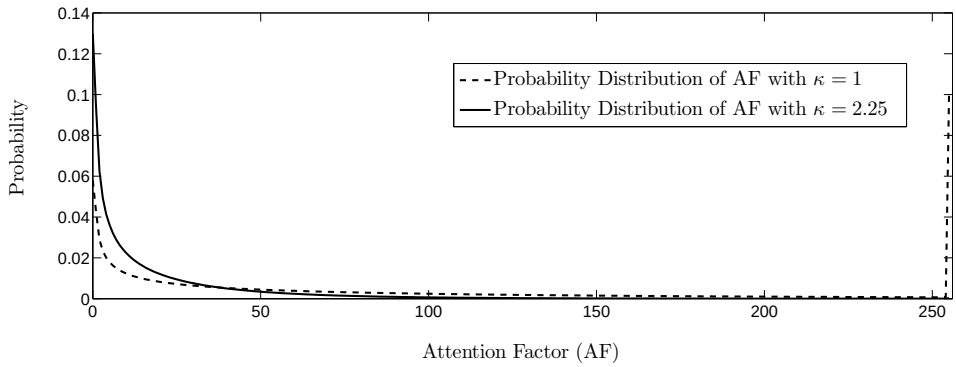


Figure 6.4: This plot compares the empirically obtained probability distributions of the attention factor for different values of the tolerance parameter κ. For $\kappa = 1$, the distribution given by the dashed line does not satisfy Property 6.4, as can be seen from the value of $\mathrm{P}(\alpha = A_{\max} - 1)$. However, for $\kappa = 2.25$, the solid line distribution clearly satisfies this property.

Impact of the tolerance parameter κ: The impact of κ can be noted from the attention factor distributions presented in Figure 6.4. The attention factors in this example are calculated for a process with parameters $A = 1$, $B = 1$, $C = 1$ and all initial variances equal to one. The maximum value of the attention factor is $A_{\max} = 256$. The parameter $\kappa = 2.25$ ensures Property 6.4, whereas $\kappa = 1$ does not, as shown in Figure 6.4. In general, lower values of κ result in a peak at the higher end of the probability distribution of the attention factor, while higher values of κ under utilize the range of the attention factor. Critical nodes must set as low a value for κ as possible to generate packets with high attention values for small deviations in measurements.

Lemma 6.5. *For a policy f_k that results in a non-increasing tail for the attention factor distribution, as defined in Property 6.4, the conditional probability of winning* $\mathrm{P}(W_{N_T,M-1}|\alpha)$ *is a non-decreasing function of α.*

The proof for this lemma is presented in Appendix C. Using this lemma, we derive a more interesting result below.

Lemma 6.6. *For a policy f_k that results in a non-increasing tail for the attention factor distribution, as defined in Property 6.4, the conditional probability of a successful transmission* $\mathrm{P}(T_{N_T,M-1}|\alpha)$ *is a non-decreasing function of α.*

Proof. From Property 6.4, we have $\mathrm{P}(\alpha) \leq \mathrm{P}(\alpha - 1)$, for $\alpha > 0$. This implies that $p_{\mathrm{G}}(\alpha) \leq p_{\mathrm{G}}(\alpha - 1)$ and $p_{\mathrm{L}}(\alpha) \geq p_{\mathrm{L}}(\alpha - 1)$. With these inequalities, we use

induction in Lemma 6.5 to show that $\mathrm{P}(W_{N_\mathrm{T},M-1}|\alpha_k)$ is a non-decreasing function of the attention factor. Then, to obtain the probability of a successful transmission $\mathrm{P}(T_{N_\mathrm{T},M-1}|\alpha_k)$, note that each of the occurrences of p^n_{LE} must be altered as $p^n_{\mathrm{LE}} \cdot (1 - \sum_{r=}^{n} C^n_r(\mathrm{P}(\alpha))^r)$, to discount the possibility of a collision for a node with attention factor α. Now, the second factor in this product term clearly increases as $\mathrm{P}(\alpha)$ increases, and hence does not alter the monotonic property of the other terms. Thus, the resulting expression $\mathrm{P}(T_{N_\mathrm{T},M-1}|\alpha_k)$ has the same property as $\mathrm{P}(W_{N_\mathrm{T},M-1}|\alpha_k)$. □

An immediate consequence of this result is given below.

Lemma 6.7. *For a homogenous network of M systems given by (6.1)–(6.6), with a policy f_k that results in a non-increasing tail for the attention factor distribution, as defined in Property 6.4, and a tournament $\mathcal{T}$, as defined in (6.14), the posterior variance of the innovations is less than its a priori value.*

The proof of this lemma is presented in Appendix C. We are now ready to present the main result of this section, which highlights the benefits of using state-based priorities.

Theorem 6.8. *For a homogenous network of M systems given by (6.1)–(6.6), with a policy f_k that results in a non-increasing tail for the attention factor distribution, as defined in Property 6.4, and a tournament $\mathcal{T}$, as defined in (6.14), it holds that:*

i. The variance of the estimation error at the DPU can be bounded from above as

$$J_E \leq \mathsf{tr}\{P_{loss}(p_{\mathfrak{T}})\} \,, \tag{6.24}$$

where $P_{loss}(p_{\mathfrak{T}})$ is the estimation error covariance obtained with a Bernoulli packet loss process of probability $1 - p_{\mathfrak{T}}$ (Schenato et al., 2007).

ii. The LQG control cost at the DPU can be bounded from above as

$$J_C \leq J_{loss}(p_{\mathfrak{T}}) \,, \tag{6.25}$$

where $J_{loss}(p_{\mathfrak{T}})$ is the LQG cost obtained with a Bernoulli packet loss process of probability $1 - p_{\mathfrak{T}}$ (Schenato et al., 2007; Gupta et al., 2007).

Proof. We use the fact that the variance of the innovations as seen by the DPU is less than its a priori value, shown in Lemma 6.7. Furthermore, from Proposition 6.1, we know that $\mathsf{tr}\{AK_{f,k}\,\mathbb{E}[e_k e_k^\top]K_{f,k}^\top A^\top\} = \mathsf{tr}\{P_{k+1|k-1} - P_{k+1|k}\}$. Thus, the reduction in variance implies that

$$\mathsf{tr}\{P^c_{k|k-1} - P^c_{k|k}\} \leq \mathsf{tr}\{P^s_{k|k-1} - P^s_{k|k}\} \,, \tag{6.26}$$

following a successful transmission at time $k-1$, where $P^c_{k|k} = P^s_{k|k}$ as this indicates a successful transmission at time k as well. Thus, we have the desirable inequality

$\mathsf{tr}\{P^c_{k|k-1}\} \leq \mathsf{tr}\{P^s_{k|k-1}\}$, following a successful transmission at time $k-1$. In fact, for any burst of non-transmissions of length $\ell \geq 0$, the inequality $\mathsf{tr}\{P^c_{k+\ell|k-1}\} \leq \mathsf{tr}\{P^s_{k+\ell|k-1}\}$ holds due to the evolution of the prediction error covariance, even if we assume that there are no further reductions in variance due to using the tournament access.

Now, note that a Bernoulli packet loss process does not provide any information about the innovations, and hence, the variance at the DPU is always given by $\mathsf{tr}\{P^s_{k|\tau_k}\}$, where τ_k is the time index of the last received packet. Thus, the above inequalities imply that the variance of the DPU with tournament access can be upper-bounded by the variance of the DPU with a Bernoulli packet loss process. Using these inequalities in the expression for the average estimation error variance gives us the desired result for estimation. We have

$$\begin{aligned}\mathsf{tr}\{\mathbb{E}[P^c_{k|k}]\} &= \sum_{d=0}^{\infty} \mathsf{tr}\{P^c_{k|k-d}\} \cdot \mathrm{P}(d_k = d) \\ &\leq \sum_{d=0}^{\infty} \mathsf{tr}\{P^s_{k|k-d}\} \cdot \mathrm{P}(d_k = d) = \mathsf{tr}\{P_{\mathrm{loss}}(p_{\mathfrak{T}})\} \,, \end{aligned} \tag{6.27}$$

where the average estimation error at any time k can be obtained by marginalizing over the delay due to the independent packet transmission process at each time instant.

Due to the certainty equivalence property from Proposition 6.1, the inequality for the control cost follows from the inequality for the estimation error variance and is given by (6.25). □

Any non-state-based random access mechanism can be modelled as a Bernoulli packet loss channel, with suitable assumptions on the operating time scales of the system and the protocol. Thus, the above result indicates that a well-designed tournament with state-based priorities can outperform any random access mechanism that results in the same probability of transmission, for the estimation and control costs considered in this chapter.

Extension to Packet Erasures due to a Lossy Medium: In the work presented so far, we have implicitly assumed that the physical medium does not drop packets. However, this may not be the case in general. If we model the losses due to the physical medium using a Bernoulli process with the probability of loss denoted by p_{loss}, then the probability of a successful transmission will be altered to be

$$\hat{p}_{\mathfrak{T}} := (1 - p_{\mathrm{loss}}) \sum_{\alpha_k} \mathrm{P}(T_{N_{\mathrm{T}},M-1}|\alpha_k)\, \mathrm{P}(\alpha_k) \,, \tag{6.28}$$

where the conditional probability of transmission $\mathrm{P}(T_{N_{\mathrm{T}},M-1}|\alpha_k)$, for $0 \leq \alpha_k \leq A_{\max}$, is multiplied by the compliment of the loss probability. Thus, note that p_{loss} is indiscriminate of the attention factor of the data packet, whereas the probability

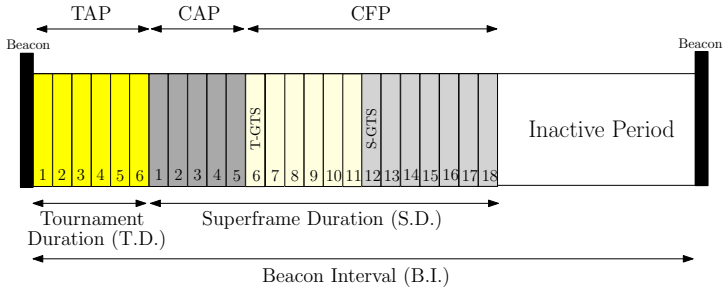


Figure 6.5: Example of a superframe structure of the tournament-based IEEE 802.15.4 MAC. The tournament occurs during the TAP, while the transmission of the winner nodes is assigned to the T-GTSs, during the CFP. Scheduled communication for other network nodes is performed during the S-GTS, while best-effort transmissions occur during the CAP.

of transmission that we obtain in the above analysis varies with the attention factor. By substituting the probability of transmission $p_{\mathfrak{T}}$ with $\hat{p}_{\mathfrak{T}}$ in the above analysis, we can include the effects of losses from the physical layer in evaluating the control and estimation performance.

6.4 Tournament Access Mechanism

We implement the tournament access mechanism as part of the current IEEE 802.15.4 (IEEE, 2006) standard. The IEEE 802.15.4 physical and medium access control (MAC) layers are used in some of the proposed protocols for control over wireless, e.g., WirelessHART (HART Communication Foundation, 2007) and ISA 100.11a (International Society of Automation, 2010). This protocol is particularly efficient for sensing applications as it provides a hybrid MAC layer that integrates both guaranteed time slots and contention-based slots in a single scheme. However, it is not optimized for wireless networked control systems (NCSs) as transmissions can only take place through random access and/or be dynamically scheduled, while incurring a fixed delay. The scheduled transmissions are delayed by at least one beacon interval, and can only be scheduled when guaranteed slots are available. This feature in the protocol was meant for slow monitoring applications that require a few continuous guaranteed transmissions, such as video or voice. Finally, adding prioritized access makes this protocol suitable for wireless control applications. Thus, the tournament access mechanism we propose is complimentary to existing features of this protocol. The compatibility of the proposed protocol stack with the IEEE 802.15.4 is desirable since it represents a de-facto standard at physical and MAC layer for sensor network solutions.

We focus on the beacon-enabled mode MAC specified in the standard. In such a setup, a centralized coordinator node, the network manager, is responsible for

synchronizing and configuring all the nodes. The synchronization and configuration messages take place periodically at each beacon message which defines the time bounds of the superframe structure. The Beacon Interval (BI) denotes the superframe length. The BI is further divided into active and inactive periods, as shown in Figure 6.5. The active period has a time interval defined by the Tournament Duration (TD) and the Superframe Duration (SD). The TD and SD can be divided in a maximum of 32 equally sized slots each. The length of the active period is given by the sum of the TD and SD, in which each can have a maximum of 32 equally sized slots. The TD contains the Tournament Access Period (TAP), which comprises of multiple tournament slots. In each tournament slot, nodes transmit priorities and receive acknowledgements if they are winners of the tournaments.

The SD contains the Contention Access Period (CAP) and the Collision Free Period (CFP). During the CAP, nodes transmit best effort messages using CSMA/CA. The CFP is intended to provide real-time guaranteed service, by allocating Guaranteed Time Slots (GTS) to the nodes using a TDMA scheme. In our protocol, two types of GTS slots are defined: Tournament GTS (T-GTS), where the transmissions of tournament winner nodes take place, and Standard GTS (S-GTS), which can be scheduled by the network manager for communication between specific nodes, in a TDMA fashion. An inactive period is defined at the end of the active period so that the network nodes and the network manager enter a low-power mode and save energy.

In the current implementation, we allow for the different channel access mechanisms to be used. Particularly, nodes which lose a tournament may be allowed to transmit their information in a best-effort manner during the CAP at the current superframe, or may be also scheduled for transmission by the network manager in S-GTSs in the following superframe CFP. Such mechanisms are application specific and are left to be defined by the user.

6.4.1 Tournament Access Mechanism Implementation

The implementation was performed for the Telos wireless platform (Polastre et al., 2005). These nodes are equipped with a Texas Instruments MSP430 16-bit, 8 Mhz microcontroller with 48 kB of Flash and 10 kB of RAM memory, 250 kbps 2.4GHz Chipcon CC2420 IEEE 802.15.4 compliant radio and on-board sensors. The operating system used is TinyOS (Levis et al., 2004). The implementation of the protocol is based on the IEEE 802.15.4 MAC TinyOS implementation (Hernandez, 2011). The code for these experiments is available for download at KTH wireless NCS code repository (2013). The implementation of the complete protocol requires 30 kB of memory.

The tournament access mechanism is implemented according to the tournament defined in Section 6.3.2. Each sensor node computes its attention factor according to (6.12) (decimal number), converting it afterwards to a binary sequence. The implementation is then carried out in the following manner. For a value of '1', an unmodulated carrier pulse is transmitted by the CC2420 radio. When a recessive bit

Table 6.6: Tournament access mechanism validation

Number of nodes	topology	distance to coordinator	false positive %
4	circle	5m	0
4	circle	2.5m	0
2	line	15m	0.7

(value '0') is present, the node refrains from transmitting an unmodulated carrier pulse, and detects unmodulated carrier pulses transmitted by other devices. This action is performed by the Clear Channel Accessment (CCA) mechanism of the CC2420 radio. When the CCA is issued, the average Received Signal Strength Indicator (RSSI) value is measured for a duration of $128\mu s$. Then, this value is compared to a pre-defined threshold in order to decide if the radio channel is busy or idle.

6.4.2 Tournament access mechanism validation

Several tests have been performed to validate the proposed tournament access mechanism. The results are summarized in Table 6.6. The tests were performed indoors. In each test a network manager is deployed in addition to the number of nodes specified in the table.

In the circle topology, all nodes were placed in a room at the same distance from the network manager and spaced 90 degrees apart. In the line topology, the nodes were placed in a long corridor stretch, with the network manager node in the middle. The devices priority was fixed through all tournaments and set to $\{160, 72, 37, 32\}$ for the 4 node case, and $\{160, 72\}$ in the 2 node case. A total number of 60000 tournaments was performed for each setup. As the results show, the tournament mechanism has no errors for short distances. However, a small percentage of false positives is verified for the case of a distance of 30 meters between the nodes.

6.5 Results

We provide two types of experimental results in this section. We first begin by validating our analysis in Section 6.3 with simulation results. Next, we present experimental results obtained from testing our protocol on state-of-the-art low power wireless devices.

6.5.1 Verification of Analytical Results using Simulations

We now present the results obtained from a Monte Carlo simulation, and use these to validate the analysis in Section 6.3. For this experiment, we simulated a network

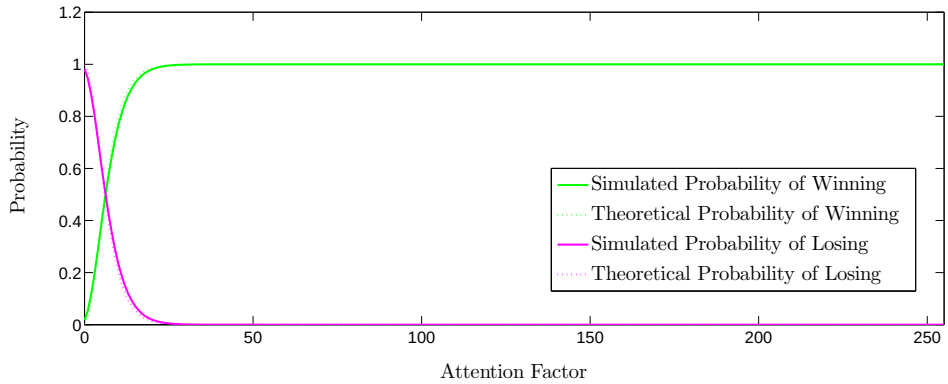


Figure 6.7: The conditional probabilities of winning and losing in the tournament access mechanism are plotted above. Note that the simulated probabilities of these complimentary events match the analytical values. The probability of winning is a non-decreasing function of the attention factor, as shown in Lemma 6.5.

of $M = 20$ control systems that use attention-based tournaments in Matlab. We considered the process to be controlled as a first-order plant with a scalar measurement. We chose the parameters in (6.1) to be $A = 1$, $B = 1$, $C = 1$, with all variances set to one. Additionally, we used $Q_1 = 1$ and $Q_2 = 1$ in the infinite horizon LQG cost in (6.9). Each sensor in the network generated a packet to transmit, and these packets vied for $N_{\mathrm{T}} = 10$ tournament slots. The maximum value of the attention factor $A_{\max} = 256$, which is sufficiently large to prevent frequent collisions while achieving sufficient throughput. The tolerance parameter $\kappa = 2.25$ is selected to ensure Property 6.4, as shown in Figure 6.4. Recall that lower values of κ result in a peak at the higher end of the probability distribution of the attention factor, while higher values of κ under utilize the range of the attention factor.

The simulated results matched the analysis closely, as shown in Figs. 6.7 and 6.8. Thus, neglecting terms that accounted for more than two collisions in (6.19) and (6.20) is justifiable when the probability of a collision is small. The conditional probabilities of winning and transmitting successfully are almost 1 for packets with high attentions. Furthermore, these plots show the non-decreasing nature of the conditional probabilities of winning and transmitting successfully, verifying the results of Lemmas 6.5 and 6.6, respectively. The peak in the conditional probability of collision (Figure 6.8) can be explained from the PMF of the attention factor, which indicates that there are few packets with large attention factors. These are most likely to win the tournament in the first few slots and transmit without collision. Hence, the curve falls to nearly 0 for high values of α. Packets with lower values of attention factor mostly win the tournament in the last few slots, and since there are many such packets, collisions are very likely. Finally, the packets with very low

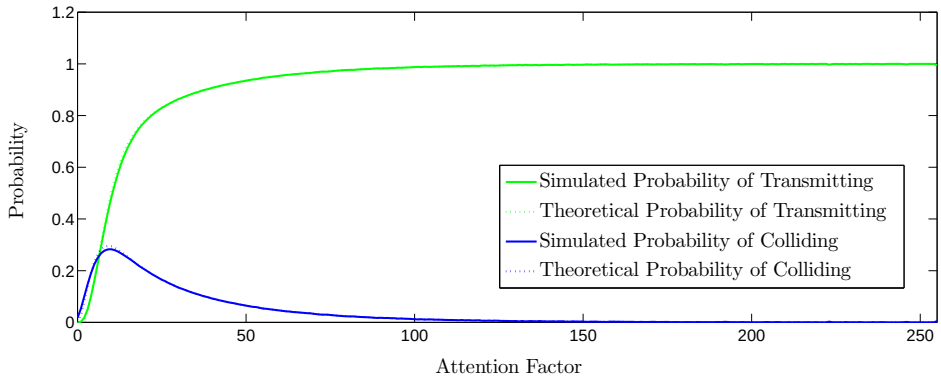


Figure 6.8: The conditional probabilities of transmitting successfully and colliding in the tournament access mechanism are plotted above. Note that the simulated probabilities match the analytical values. The probability of transmitting is a non-decreasing function of the attention factor, as shown in Lemma 6.6.

values of attention do not win the tournament often, and hence the probability of collision is low for these values.

In the proof of Lemma 6.5, we claim that the second term in the expression for the conditional probability of winning a tournament in (6.19) does not contribute much to the net value. This claim can be validated from Figure 6.9, where we compare the analytical values for the probability of winning and transmitting in a tournament from (6.19) and (6.20), respectively. For both expressions, we plot the first term alone, and compare with the full expression, to see that the effect of including even the second term is negligible. This is because the probability of collisions has been designed to be small enough to ignore.

In Table 6.10, we present a comparison of the estimation and control costs, J_E and J_C, for three setups. The entries in the first column, titled Tournament Access, correspond to values obtained from our Monte Carlo simulations of the network of systems described in this section using the tournament access mechanism along with attention factors for priorities. The average probability of transmission is found to be $p_{\mathfrak{T}} = 0.4403$ from the simulations. The entries in the second column, denoted Packet Loss, corresponds to the same network of systems using a multiple access mechanism or a physical medium that suffers from a Bernoulli packet loss process, with a probability of loss equal to $1 - p_{\mathfrak{T}}$. The third column corresponds to the network of systems and a packet loss scenario termed as 'ideal' because the probability of transmission is equal to the ratio of the available number of slots to the number of processes in the network, $N_{\mathrm{T}}/M = 0.5$. The entries in the packet loss columns are calculated using the expressions for the average estimation error variance in (6.27) and for the LQG cost from (Åström, 1970). The control cost expression uses the

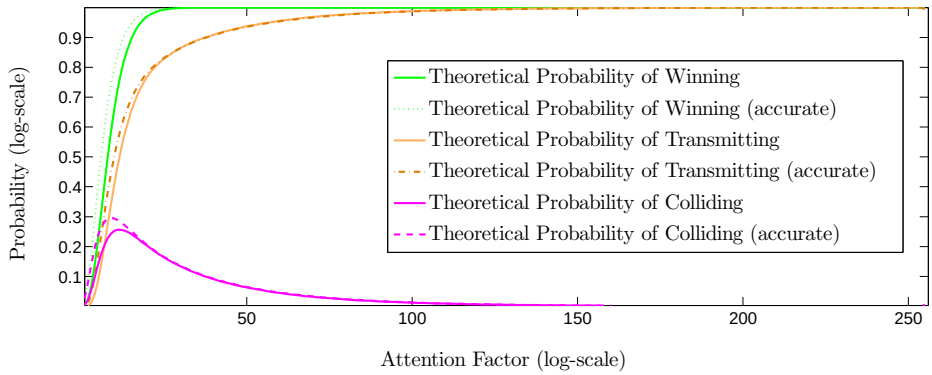


Figure 6.9: The conditional probabilities of winning, transmitting and colliding in the tournament access mechanism are plotted above. The solid line plots only the first term in (6.19) and (6.20), while the dashed lines plot the entire expressions. The second terms in both expressions contribute very little, as can be seen, due to the negligible probability of collision.

Table 6.10: A comparison of estimation and control costs

	Tournament Access	Packet Loss	Ideal Packet Loss
$p_{\mathfrak{T}}$	0.4403		0.5
$\mathbb{E}[P^c_{k\|k}]$	0.9765	1.8894	1.618
J	0.2576	0.3524	0.3252

estimation error variance found in (6.27). The values from the tournament access and packet loss columns in this table confirm the results from Theorem 6.8, showing that a well designed tournament access mechanism outperforms an agnostic packet loss mechanism that results in the same average probability of transmission. Furthermore, the values from the ideal packet loss column indicate that even a throughput achieving access mechanism cannot outperform the tournament access mechanism, despite the collisions in the latter mechanism.

6.5.2 Experimental Results

In order to validate the performance of the proposed protocol, we assembled a hardware-in-the-loop simulation of a network of wireless control systems, as depicted in Figure 6.11. The control systems, each comprising of a plant, sensor,

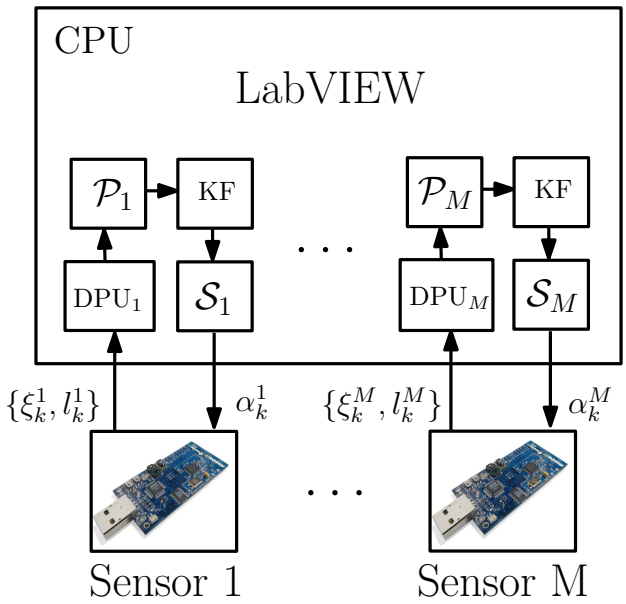


Figure 6.11: Experimental setup for the hardware-in-the-loop experiments. The plant, sensor, controller and actuator of each of the M control systems are simulated in LabVIEW. The tournaments are conducted in M wireless nodes that communicate with LabVIEW. The sensors send the attention factor (α_k^j) to their corresponding wireless node, and the wireless nodes conduct tournaments. They report whether they have won ($\xi_k^j \in \{0, 1\}$), and if so, the tournament slot number that they have won ($l_k^j \in \{1, \ldots, N_\mathrm{T}\}$).

controller and actuator, were simulated in a Matlab environment within LabView. However, the protocol was implemented on a wireless sensor node, which was connected to the computer using the serial port. The control systems generated attention factors in the simulated environment, which were sent to each node using the serial port. Then nodes conducted N_T tournaments, and returned their winning status (using the binary variable $\xi_k^j \in \{0, 1\}$), along with the tournament number that they won, to the computer. After discarding any tournament slots with multiple winners, the simulation environment closed the loop for the successful nodes.

In this setup, we considered $M = 2$ control systems, each with the same first-order linear process described in the Matlab simulations in Section 6.5.1. We also used the same parameters for the cost function as in Section 6.5.1. The data packets from these systems vied for $N_\mathrm{T} = 1$ tournament slot. The maximum value of the attention factor is $A_\mathrm{max} = 256$. We set the tolerance parameter $\kappa = 2.25$ again, to ensure a good design.

We selected the beacon interval BI as 937.5 ms, 1 TAP slots, 1 CAP slots, 1 T-GTS slots and 1 S-GTS slots which give a total of two superframe slots. The duration of each slot is 20 ms. The sampling period of the sensor node is then set to the value of the BI. Thus, the sensor node will attempt to transmit once per superframe.

The results obtained from the hardware-in-the-loop experiment are depicted in Figures 6.14–6.13. A trace of the innovations e_k^j and corresponding attention factors α_k^j, for $1 \leq j \leq 2$, are depicted in Figure 6.12. The attention factors were designed to be symmetric, quantized functions of the innovations (6.12), as can be noticed from this figure. Also, at three different time instants, the attention factors of both plants are identically zero. These result in collisions, as shown in Figure 6.13. Winners of tournaments are denoted by $\xi_k^j = 1$ and the winners who also get to transmit are denoted by $\delta_k^j = 1$. Losers of tournaments and those who do not get to transmit are denoted by the complimentary values $\xi_k^j = 0$ and $\delta_k^j = 0$, respectively. A trace of the attention factors, along with the indicators ξ_k^j and δ_k^j are depicted in this figure, for $1 \leq j \leq 2$. Notice that the highest attention factor always wins, as desired in tournaments. Also, equal attention factors from both the processes result in both of them winning, but none of them transmitting. This is because these data packets result in collisions. Thus, Figures 6.14–6.13 validate the implementation of the tournaments on sensor nodes.

A trace of the state x_k^j and the controls u_k^j, for $1 \leq j \leq 2$, are depicted in Figure 6.14. The winning and transmission records are superimposed onto the state and control plots, respectively. This indicates the intermittent nature of feedback control that is obtained using attention-based tournaments. The controls follow the state only when there is a transmission. One instance of this behaviour is noticeable at $k = 10$ in Figure 6.14. The state deviates considerably, giving rise to a large attention factor (see Figure 6.13). The sensor succeeds in transmitting the state to the controller, which reacts to this and successfully regulates the state at the next time instant. When there is no transmission, the control signals are generated in an open-loop manner.

6.6 Summary

We have presented a design for a distributed prioritized access mechanism using tournaments and state-based priorities called attention factors. There are two main merits of the tournament access mechanism. First, the priorities are assigned and evaluated in a distributed manner, thus rendering it suitable for wireless networks. Second, the state-based priorities result in better performance than agnostic access methods, as shown by our analysis.

A major concern in such a study is often the implementability of the proposed protocol. So, we have studied this issue in detail and implemented tournaments on sensor nodes. We have also validated our implementation with experiments using these nodes. To popularize the protocol, we have proposed modifications to

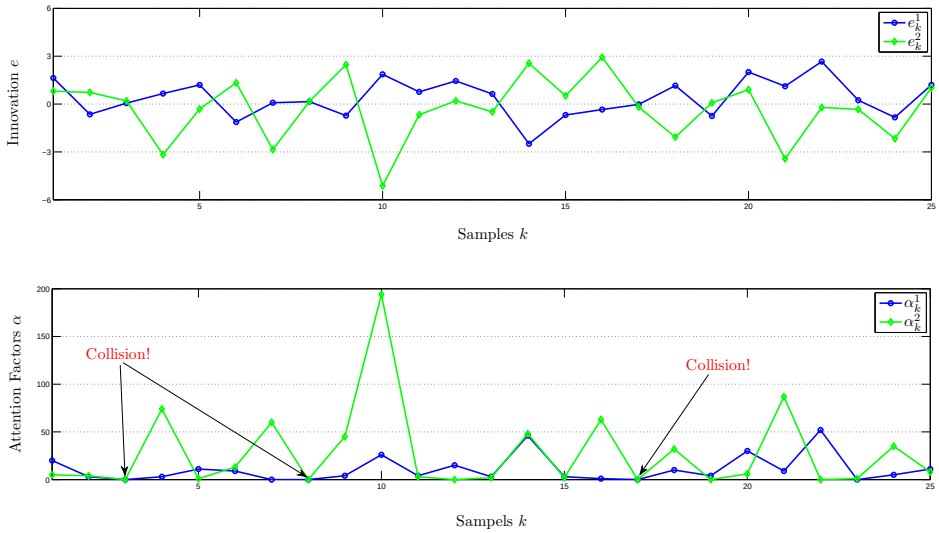


Figure 6.12: A trace of the innovations e_k^j (above) and attention factors α_k^j (below), for $1 \leq j \leq 2$. The attention factors are obtained by quantizing a distortion-like function of the innovations (6.12). Note that three collisions are imminent from this trace.

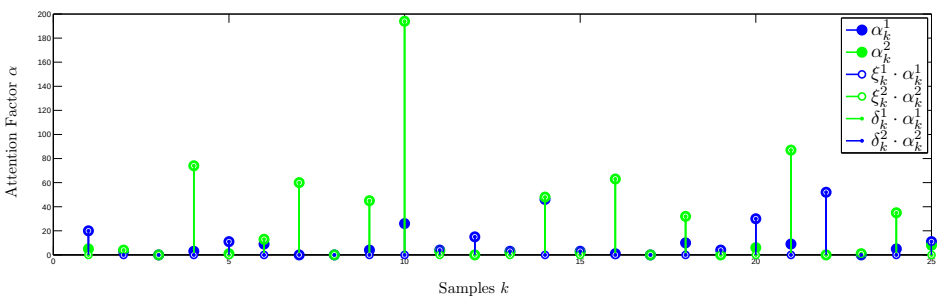


Figure 6.13: A trace of the attention factors, along with the indicators of winning and transmitting in the tournaments, as denoted by ξ_k^j and δ_k^j, respectively, for $1 \leq j \leq 2$.

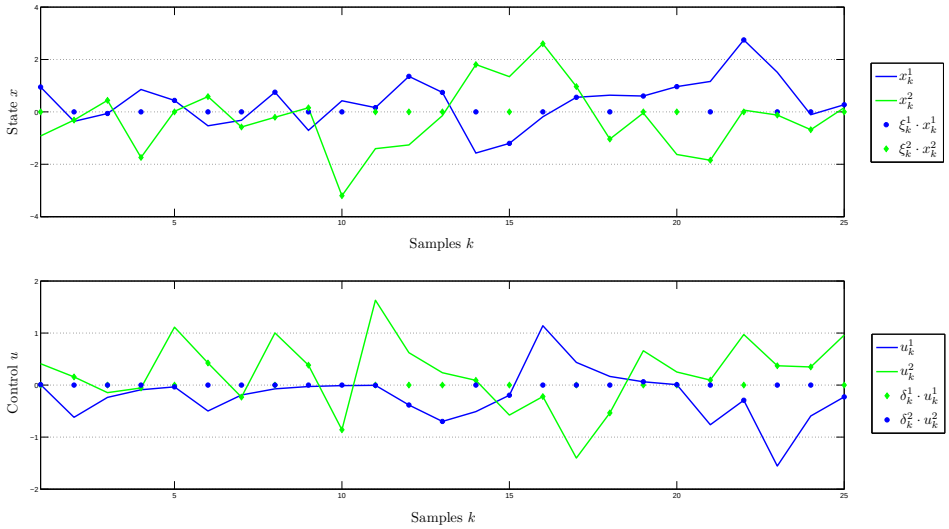


Figure 6.14: A trace of the state x_k^j (above) and controls u_k^j (below), for $1 \leq j \leq 2$, as obtained from the hardware-in-the-loop experiment. Winners of tournaments are denoted by $\xi_k^j = 1$, and depicted with a marker overlayed on the state plot. Nodes that get to transmit are denoted by $\delta_k^j = 1$, and depicted with a marker overlayed on the controls plot.

IEEE 802.15.4, which introduces priorities to a sensor network protocol, making it suitable for estimation and control of physical systems over a wireless network.

With this chapter, we conclude our presentation of state-based channel access mechanisms. In the next chapter, we consider a generalized NCS setting to examine the impact of a nonlinear dynamical measurement process on the structural properties of the control system.

Chapter 7

Stochastic Systems with Nonlinear Measurements

We consider discrete-time sequential decision problems for a control loop that has a communication bottleneck between the sensor and the controller (Figure 7.1). The design problem is to choose in concert an encoder and a controller. The encoder maps the sensor's raw data into a causal sequence of channel inputs. Depending on the channel model adopted in this chapter, the encoder performs either sequential quantization, sampling, or analog companding. The controller maps channel outputs into a causal sequence of control inputs to the plant. Such two-agent problems are generally hard because the information pattern is non-classical, as the controller has less information than the sensor (Witsenhausen, 1971). This gives scope for the controller to exploit any dual effect present in the loop, even when the plant is linear (Feldbaum, 1960). These two-agent problems are at the simpler end of a range of design problems arising in networked control systems (Borkar et al., 2001a; Baillieul and Antsaklis, 2007; Goodwin et al., 2008; Andrievsky et al., 2010). Naturally, one seeks formulations of these design problems as stochastic optimization problems whose solutions are tractable in some suitable sense.

The classical partially observed linear quadratic Gaussian (LQG) optimal control problem is a one-agent decision problem (Wonham, 1968). Given a linear, Gauss-Markov plant, one is asked for a causal controller, as a function of noisy linear measurements of the state, to minimize a quadratic cost function of states and controls. This problem has a simple and explicit solution, where the optimal controller 'separates' into two policies; one to generate a minimum mean-squared error estimate of the state from the noisy measurements, and the other to control the fully observed Gauss-Markov process corresponding to the estimate. A networked version of this problem is the following two-agent LQG optimal control problem (Borkar and Mitter, 1997). Given a linear Gauss-Markov plant and a channel model, one is asked for an encoder and controller to minimize a preformance cost which is a sum of a communication cost and a quadratic cost on states and controls. The communication cost is charged on decisions at the encoder, which are

chosen to satisfy constraints imposed by the channel model. No causal encoding or control policies are, in general, excluded from consideration. As in the one-agent version, a certain 'separated' design is optimal, as has been noted in various settings since the sixties (Larson, 1967; Sauer and Melsa, 1974; Fischer, 1982; Bansal and Başar, 1989; Mitter, 2001; Tatikonda et al., 2004; Matveev and Savkin, 2004; Wu and Arapostathis, 2005; Nair et al., 2007; Bao et al., 2011; Yüksel, 2012; Molin and Hirche, 2013; Yüksel, 2013). Precisely, the following combination is optimal: certainty equivalence controls with a minimum mean-squared estimator of the state, and an encoder that minimizes a distortion for state estimation at the controller. The distortion is the average of a sum of squared estimation errors with time-varying coefficients depending on the coefficients of the performance cost. This separation is different from that obtained in the classical, partially observed LQG problem, but it is still due to a linear evolution of the state, and the statistical independence of noises from all other current and past variables. As in the classical one-agent version (Root, 1969), the random variables do not need to be Gaussian.

7.1 Contributions and Related Work

The two-agent networked LQG problem has a long history. Different channel models have been treated, leading to different types of encoders. Thus we find in these works, the encoder is either a quantizer, or an analog, time-dependent compander, or an event-triggered sampler. But there is a common theme in these works. Several authors suggest that for what we call the dynamic encoder-controller design problem, separated design of encoder and controller is optimal, and that certainty equivalence control is optimal.

When one treats a discrete alphabet channel, one has to treat the encoder as a time-dependent quantizer. Quantized control has been explored since the sixties, and structural results for this problem have seen spirited discussions over the years (Larson, 1967; Marleau and Negro, 1972; Fischer, 1982). This problem was revisited by Borkar and Mitter (1997) in recent years, setting off a new wave of interest. A survey can be found by Nair et al. (2007) and Fu (2012).

When one treats an additive noise channel, one has to treat the encoder as a time-dependent, possibly non-linear, compander. The corresponding networked LQG problem has been studied by Bansal and Başar (1989), and more recently by Freudenberg et al. (2011). When one treats analog channels with channel use restrictions, one has to treat the encoder as an event-triggered sampler (Åström and Bernhardsson, 2002). The networked LQG problem for event-triggered sampling is studied by Molin and Hirche (2013).

The above papers suggested separated designs for the two-agent LQG problem with dynamic encoder and controller, and certainty equivalence controls. This is despite other results by Curry (1970) and Feng and Loparo (1997), confirming the dual effect in the two-agent networked control problem. Thus, there can be an incentive to the controller to influence the estimation error, and yet the optimal

controller chooses to ignore this incentive. Furthermore, for the two-agent LQG problem with event-triggered sampling, and with zero order hold control between samples, Rabi and Johansson (2009b) showed through numerical computations that it is suboptimal to apply controls affine in the minimum mean square error (MMSE) estimate. The optimal controls are nonlinear functions of the received samples. Thus, the literature does not tell us when separation holds, and when it does not, for the general class of two-agent problems.

We make three main contributions. Firstly, we show that for the combination of a linear plant and nonlinear encoder, the dual effect is present. This confirms the results of Curry (1970) and others (Feng and Loparo, 1997), by establishing through a counter example that there is a dual effect in the closed-loop system. In fact, each of the three models we allow for the channel endow the loop with the dual effect. The dual role of the controller lies in reducing the estimation error in the future, using the predicted statistics of the future state and knowledge of the encoding policy. Due to this dual role, we show that, in general, separated designs need not be optimal for linear plants with non-linear measurements, even with independent and identically distributed (IID) Gaussian noise and quadratic costs. Examples 4, 5, 6, and 8 show instances where the dual effect matters.

Our second contribution is a proof for separation in one specific design problem. We prove that for the dynamic encoder controller design problem, it is optimal to apply separation and certainty equivalence. We also notice that the result holds under a variety of schemes for charging communication costs. For example, it holds even when the encoder is an analog compander with hard amplitude limits. Our proof does not require the dual effect to be absent. Hence there is no contradiction with the fact separation and certainty equivalence are not optimal for other design problems concerning the same plant sensor combination. Our work also provides a direct insight to explain separation or the lack of it, in the form of a property of the optimal cost-to-go function. Furthermore, we show that when this property does not hold separation is no longer optimal.

Our third contribution points out some subtleties that arise when dynamic policies are involved. We explicitly demonstrate that with dynamic encoders for LQ optimal control, one cannot extend and apply a result of Bar-Shalom and Tse (1974), which mandates absence of dual effect for certainty equivalence to be optimal. The classical notion of a dual effect was introduced for static measurement policies, and the dual role of the controls has been motivated through the notion of a probing incentive (Feldbaum, 1960). We ask if the concept of probing applies unchanged for dynamic measurement policies. We point out some subtleties in answering this question.

The chapter is organized as follows. In Section 7.2, we present a basic problem formulation, pertaining to encoder and controller design for data-rate limited channels. In Section 7.3, we discuss the notion of a dual effect and certainty equivalence, and present a counterexample to establish that there is a dual effect in the considered networked control system (NCS). In Section 7.4, we present a proof for separation in the two-agent networked LQG problem. In Section 7.5, we extend

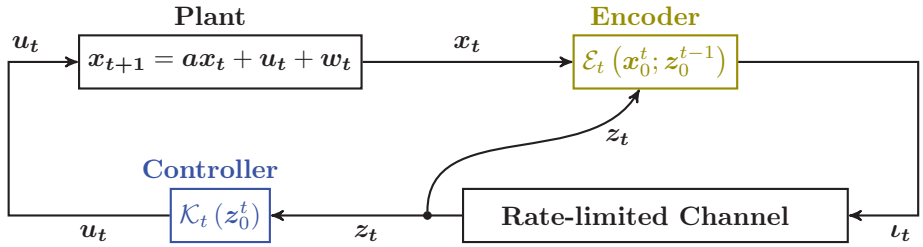


Figure 7.1: Control over a rate-limited channel that has a perfect feedback channel

our results to other channel models, including event-triggered samples and additive noise channels. In Section 7.6, we present a number of counterexamples to illustrate that in general, separation does not hold for constrained design problems.

7.2 Problem Formulation

In this section, we describe a version of the two-agent networked LQG problem, corresponding to a rate-limited channel model. We consider an instantaneous, error-free, discrete-alphabet channel and the logarithm of the size of the alphabet is the bit rate. A control system that uses such a channel to communicate between its sensor and controller is depicted in Figure 7.1, and comprises of four blocks. Each of these blocks, along with the performance cost, are described below, followed by a description of the design problems under consideration.

7.2.1 Plant

The plant state process $\{x_t\}$ is scalar, and its evolution law is linear:

$$x_{t+1} = ax_t + u_t + w_t, \tag{7.1}$$

for $0 \leq t \leq T$. Here $\{u_t\}$ is the controls process, and $\{w_t\}$ is the plant noise process, which is a sequence of independent random variables with constant variance σ_w^2, and zero means. The initial state x_0 has a distribution with mean $\overline{x}_0$ and variance σ_0^2. At any time t, the noise w_t is independent of all state, control, channel input, and channel output data up to and including time t. We assume that the state process is perfectly observed by the sensor.

7.2.2 Performance Cost

The performance cost is a sum of the quadratic cost charged on states and controls, and a communication cost charged on encoder decisions:

$$J = \mathbb{E}\left[x_{T+1}^2 + p\sum_{i=1}^{T} x_i^2 + q\sum_{i=0}^{T} u_i^2\right] + J^{\text{Comm}} \tag{7.2}$$

where $p > 0$ and $q > 0$ are suitably chosen scalar weights for the squares of the states and controls, respectively. The communication cost J^{Comm} is an average quantity that depends on the encoding and control policies, and the channel model adopted.

7.2.3 Channel Model

The channel model refers to an input-output description of the communication link from the sensor to the controller. We denote the channel input at time t by ι_t, the corresponding output by z_t, and the encoding map generating ι_t by $\mathcal{E}_t$. In Figure 7.1, we consider an ideal, discrete alphabet channel that faithfully reproduces inputs, and thus, $\iota_t = z_t$. The encoder's job is to pick at every time t, the encoding map $\mathcal{E}_t$ producing a channel output letter from the pre-assigned finite alphabet $z_t \in \{1, \ldots, N\}$, where the non-negative integer N is the pre-assigned size of the channel alphabet. Since the alphabet is fixed, we have a hard data-rate constraint at every time. Hence there is no explicit cost attached to communication, so $J^{\text{Comm}} \equiv 0$ in this case. In Section 7.5, we consider other channel models that permit the data-rate or energy needed for each transmission to be chosen causally by the encoder.

7.2.4 Controller

The control signal u_t is real valued and is to be computed by a causal policy based on the sequence of channel outputs. The controller has perfect memory, and thus remembers all of its past actions, and the causal sequence of channel outputs. Thus, in general, at every time t the controller's map takes the form:

$$\mathcal{K}_t \;:\; \left\{t, \{z_i\}_0^t, \{u_i\}_0^{t-1}\right\} \mapsto u_t.$$

7.2.5 Encoder

At all times, the encoder knows the entire set of control policies employed by the controller and the statistical parameters of the plant. With this prestored knowledge, the encoder works as a causal quantizer mapping the sequence of plant outputs. Thus, the encoder's map takes the form:

$$\mathcal{E}_t \;:\; \left\{t, \{x_i\}_0^t, \{z_i\}_0^{t-1}, \{\mathcal{K}_i(\cdot)\}_0^{t-1}\right\} \mapsto\; z_t.$$

Notice that we do not allow the encoder to directly view the sequence of inputs to the plant. This subtle point plays an important role in the examples we present in Section 7.7.

7.2.6 Design Problems

For a given information pattern, different design spaces may arise due to engineering heuristics, hardware or software limitations, etc. Any such design space is a subset of the set of all admissible encoder and controller pairs. We identify a few design problems, each associated with its own design space. For these design problems, an adopted channel model can be either the one described in Section 7.2.3, or any of the models from Section 7.5. First, we pose a single-agent design problem which has a classical information pattern.

Design problem 1 (Controller-only Design)**.** *The controller-only design problem is to pick a causal sequence of control policies* $\{\mathcal{K}_t\}_0^T$ *to minimize the performance cost (7.2).*

Next we pose a design problem where the design space is the largest possible non-randomized set of admissible encoder-controller pairs. We consider every causally time-dependent encoder and controller. In other words, for this type of design problem, regardless of the choices one makes for channel and communication cost, at any time, the controller can update the control signal using all of the channel outputs up till then.

Design problem 2 (Dynamic Encoder-Controller Design)**.** *The dynamic encoder-controller design problem is to pick causal sequences of encoding and control policies* $\{\mathcal{E}_t\}_0^T, \{\mathcal{K}_t\}_0^T$ *to minimize the performance cost (7.2).*

Next we pose a design problem where the controller and encoder must respect a restriction on selecting the control signals or encoding maps. At every time, the control values must be chosen from a restricted set $\mathcal{U}$, such as the interval $(-1, 1)$ or the finite set $\{-1, 0, 1\}$. Likewise, the encoding maps have to be chosen from within restricted sets. For example, the encoding maps may be constrained to consist of two quantization cells $(-\infty, \theta), (\theta, \infty)$, where the encoder threshold θ must be chosen from a restricted set Θ, say the interval $(-5, 5)$. Subject to these constraints, the controller and encoder policies are still to be dynamically chosen.

Design problem 3 (Constrained Encoder-Controller Design)**.** *The constrained encoder-controller design problem is to pick causal sequences of encoding and control policies* $\{\mathcal{E}_t\}_0^T, \{\mathcal{K}_t\}_0^T$*, subject to the constraints represented by* $\theta \in \Theta$ *and* $u_k \in \mathcal{U}$*, to minimize the performance cost (7.2).*

Next we pose a design problem where the controller must respect not only the information pattern in the dynamic encoder-controller design problem (Design problem 2), but must also respect a restriction on updating controls. Basically, the

control waveform is generated in a piece-wise 'open-loop' way, while epochs and encoding maps are picked using dynamic policies. Let $\epsilon_0, \epsilon_1 \geq 1$, be two random integers such that $\epsilon_0 + \epsilon_1 = T + 1$. Then the two epochs are $\{0, \ldots, \epsilon_0 - 1\}$ and $\{\epsilon_0, \ldots, T\}$. These epochs are chosen by the controller respecting the inequalities: $1 \leq \epsilon_0 < T + 1$ and $\epsilon_1 = T + 1 - \epsilon_0$, and hence have to be adapted to all the data available at the controller. Within an epoch, the controller must pick controls depending only on data at the start of the epoch. Precisely, given the condition that $t < \epsilon_0$, and given the initial observation z_0, the controls u_t must be a fixed function of (t, z_0) regardless of the data $\{z_1, \ldots, z_t\}$.

Design problem 4 (Hold-Waveform-Controller and Encoder Design)**.** *The hold-waveform-controller and encoder design problem is to pick a causal sequence of encoding polices $\{\mathcal{E}_t\}_0^T$ in concert with a causal sequence of policies for epochs and controls to minimize the performance cost (7.2). The controls are restricted to depend on the controller's data in the specific form:*

$$u_t = \begin{cases} \mathcal{K}_t^0\Big(z_0\Big) & \text{for } 0 \leq t \leq \epsilon_0 - 1, \text{ and,} \\ \mathcal{K}_t^1\Big(\{z_i\}_0^{\epsilon_0 - 1}, \{u_i\}_0^{\epsilon_0 - 1}\Big) & \text{for } \epsilon_0 \leq t \leq T. \end{cases}$$

A special case of a hold-waveform controller is that of zero order hold (ZOH) control where an additional restriction forces the control waveform be held constant over each epoch.

For all four design problems presented above, we assume the existence of measurable policies minimizing the associated costs. We avoid investigating the necessary technical qualifications except to say that if need be, one may allow randomized polices, or even reject the class of merely measurable policies in favour of the class of universally measurable policies (Bertsekas and Shreve, 1978).

7.3 Dual Effect and Certainty Equivalence

We begin by presenting a definition of the dual effect (Feldbaum, 1960) and certainty equivalence (Joseph and Tou, 1961). We then present a counterexample to establish that there is a dual effect of the controls in the NCS introduced in Section 7.2.

7.3.1 Dual Effect

In a feedback control loop, the dual effect is an effect that the controller may see in the rest of the loop. When it is present, the control laws affect not just the first moment, but also second, third and higher central moments of the controller's nonlinear filter for the state. Below, we state this formally for a controlled Markov process with partial observations available to the controller:

$$x_{t+1} = \Phi_t\Big(x_t, u_t, w_t\Big), \qquad z_t = \Psi_t\Big(x_t, u_t, \kappa_t\Big), \tag{7.3}$$

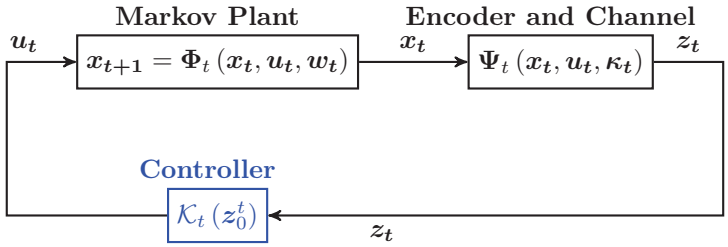


Figure 7.2: Setup for definitions in Section 7.3

where the sequences $\{x_t\}$ and $\{u_t\}$ are the real-valued plant state and control processes, respectively, see Figure 7.2. The sequence $\{z_t\}$ is the observation process and the sequences $\{w_t\}$ and $\{\kappa_t\}$ are the plant noise and observation noise processes, respectively. Assume that all the primitive random variables are defined on a suitable probability triple, $[\Omega, \mathcal{F}, \mathcal{P}]$. Now, consider two arbitrary admissible sets of control policies: $\{\mathcal{K}(t,\cdot)\}, \{\widetilde{\mathcal{K}}(t,\cdot)\}$. Once we pick one such set of control policies, they together with the measure $\mathcal{P}$ define the states, observations and controls as random processes. The choice of policies fixes their statistics. We can advertise this relationship by (1) specifying random variables, x_t for example, in the form $x_t(\boldsymbol{\omega}; \mathcal{K})$, (2) specifying a filtration, for example, the one generated by the z-process as $\mathcal{F}^{\mathcal{K},z}$, or (3) specifying an expected value of a functional, $\mathbb{E}[F_t]$ for example, in the form

$$\mathbb{E}_{\mathcal{P},\mathcal{K}}\left[F_t\Big(t, \{x_i(\boldsymbol{\omega};\mathcal{K})\}_0^t, \{z_i(\boldsymbol{\omega};\mathcal{K})\}_0^t, \{u_i(\boldsymbol{\omega};\mathcal{K})\}_0^t\Big)\right],$$

where $\boldsymbol{\omega}$ stands for any element of the sample space of the primitive random variables. To minimize the notational burden, we advertise the dependence on the set of control policies only as needed. We now define the dual effect by defining its absence.

Definition 7.1 (Dual effect)**.** *The NCS in Figure 7.2 is said to have* no dual effect of second order *if*

1. *for any two sets $\mathcal{K}, \widetilde{\mathcal{K}}$ of admissible control policies, and*
2. *for any two time instants t, s,*

we have $\mathcal{F}_t^{\mathcal{K},z} = \mathcal{F}_t^{\widetilde{\mathcal{K}},z}$ for every t, and that for any given event $X \in \mathcal{F}_t^{\mathcal{K},z}$,

$$\mathbb{E}_{\mathcal{P},\mathcal{K}}\left[\Big(x_t(\boldsymbol{\omega};\mathcal{K}) - \mathbb{E}_{\mathcal{P},\mathcal{K}}\left[x_t(\boldsymbol{\omega};\mathcal{K})\,\middle|\,\{z_i(\boldsymbol{\omega};\mathcal{K})\}_0^s, \boldsymbol{\omega}\in X\right]\Big)^2\,\middle|\,\{z_i(\boldsymbol{\omega};\mathcal{K})\}_0^s, \boldsymbol{\omega}\in X\right] =$$

$$\mathbb{E}_{\mathcal{P},\widetilde{\mathcal{K}}}\left[\Big(x_t\left(\boldsymbol{\omega};\widetilde{\mathcal{K}}\right) - \mathbb{E}_{\mathcal{P},\widetilde{\mathcal{K}}}\left[x_t\left(\boldsymbol{\omega};\widetilde{\mathcal{K}}\right)\,\middle|\,\left\{z_i\left(\boldsymbol{\omega};\widetilde{\mathcal{K}}\right)\right\}_0^s, \boldsymbol{\omega}\in X\right]\Big)^2\,\middle|\,\left\{z_i\left(\boldsymbol{\omega};\widetilde{\mathcal{K}}\right)\right\}_0^s, \boldsymbol{\omega}\in X\right].$$

Thus, we require equality of the two sets of covariances of filtering, prediction or smoothing errors, corresponding to any two choices of control strategies. In the definition above, by choosing one set of control policies, say $\widetilde{\mathcal{K}}$ as resulting in $u_t = 0$, for all t, we obtain the definition of Bar-Shalom and Tse (1974).

7.3.2 Certainty Equivalence

For the controlled Markov process (7.3), consider the general cost

$$J^{\text{general}} = \mathbb{E}\left[L\Big(\{x_i\}_0^T, \{u_i\}_0^T\Big)\right].$$

Imagine that a muse could at time t supply to the controller the exact values of all primitive random variables by informing the controller the exact element $\boldsymbol{\omega}$ of the sample space Ω. With such complete and acausal information, the controller could, in principle, solve the deterministic optimization problem

$$\inf_u J_t(u;\boldsymbol{\omega}) = \inf_u L\Big(\{x_i(\boldsymbol{\omega})\}_0^{\top}, \{u_i(\boldsymbol{\omega})\}_0^{t-1}, u, \{u_i(\boldsymbol{\omega})\}_{t+1}^T\Big).$$

Let $u_t^*(\boldsymbol{\omega})$ be an optimal control law for this deterministic optimization problem. We now state the definition of certainty equivalence from van de Water and Willems (1981):

Definition 7.2. *A certainty equivalence control law has the form:*

$$\mathbb{E}\left[u_t^*(\boldsymbol{\omega}) \,\middle|\, \{z_i(\boldsymbol{\omega})\}_0^t, \{u_i(\boldsymbol{\omega})\}_0^{t-1}\right].$$

Clearly, this law is causal. Notice also that its form is tied to the performance cost, and to the statistics of the state and observation processes. It is possible for certainty equivalence control laws to be nonlinear, and such laws can be optimal even when separated designs may not be. For linear plants, they can sometimes be linear or affine.

Lemma 7.1 (Affine CE laws for linear plants)**.** *For the plant 7.3 with $\Phi_t(\cdot) = ax_t + u_t + w_t$ and the quadratic performance cost (7.2) with $J^{Comm} = 0$, the following are certainty equivalence laws:*

$$u_t^{CE} = -k_t^{CE}\Big(a \cdot \mathbb{E}\left[x_t \,\middle|\, \{z_i\}_0^t, \{u_i\}_0^{t-1}\right] + \mathbb{E}\left[w_t \,\middle|\, \{z_i\}_0^t, \{u_i\}_0^{t-1}\right]\Big),$$

where, the gains $k_i^{CE} = \frac{\beta_{i+1}}{q+\beta_{i+1}}$, $\alpha_i = \beta_{i+1} + \alpha_{i+1}$, $\beta_i = p + \frac{a^2 q \beta_{i+1}}{q+\beta_{i+1}}$, and, $\alpha_{T+1} = 0$, $\beta_{T+1} = 1$.

For a proof, see van de Water and Willems (1981).

Definition 7.3 (Certainty equivalence property)**.** *The certainty equivalence property holds for a stochastic control problem if the optimal control law is given by the certainty equivalence control law.*

For the stochastic control problem described in Lemma 7.1, with non-linear measurements that do not result in a dual effect of the controls, Bar-Shalom and Tse (1974) showed that the certainty equivalence property holds.

We now consider a simple example, and show that there is a dual effect of the control signal in the closed-loop system presented in Section 7.2.

Example 7.1

For the plant (7.1), let $a = 1$, $x_0 = 2$, and $\sigma_0 = 0$. Let this information be known to the encoder and the controller, which simply means that $z_0 = x_0$. Let the variance $\sigma_w^2 = 0.7^2$. For the objective function, let the horizon end at $T = 1$, and let $p = q = 0.01$. Let the channel alphabet be $\{1, 2, 3\}$.

For the given threshold $\theta = 1.6$, let the encoder at $t = 1$ be:

$$\xi_1(x_1) = \begin{cases} 1 & \text{if } x_1 \in (-\infty, -\theta), \\ 2 & \text{if } x_1 \in (-\theta, \theta), \\ 3 & \text{if } x_1 \in (\theta, +\infty). \end{cases} \tag{7.4}$$

The optimal control law at $t = 1$ is $u_1 = -\frac{a}{q+1}\widehat{x}_{1|1}$, where $\widehat{x}_{1|1} = \mathbb{E}[x_1 | x_0, u_0, z_1]$.

Using the encoding policy ξ_1 and the optimal control signal u_1, the performance cost with $J^{\text{Comm}} = 0$ can be written as a function of u_0 :

$$J(u_0) = \sigma_w^2 + q u_0^2 + \left(p + \frac{qa^2}{q+1}\right) \mathbb{E}\left[x_1^2 | x_0, u_0\right] + \overbrace{\frac{a^2}{q+1}\mathbb{E}\left[\left(x_1 - \widehat{x}_{1|1}\right)^2 \middle| x_0, u_0, z_1\right]}^{\triangleq \Gamma}$$

In the above expression, Γ is the quantization distortion term, which is thus proportional to the conditional variance of the controller's minimum mean-squared estimation error of x_1. Notice that Γ is a function of u_0, thus resulting in a dual effect of the control signal in the plant-encoder-channel combination. Figure 7.3 shows how the quantization distortion Γ depends on u_0. The total cost J is also plotted and the optimal value u_0^* is shown to be different from the certainty equivalent value u_0^{CE}.

7.4 Dynamic Encoder-Controller Design

In this section we solve the dynamic encoder-controller design problem (Design problem 2) which allows both controls and encoders to be dynamic. We work out the details for the discrete alphabet channel with the fixed alphabet size N. We begin by examining a known structural property of optimal encoders. This states that it is optimal for the encoder to apply a quantizer on the state x_t, with the shape of the quantizer depending only on past quantizer outputs. Next, we present a structural property for encoders called *controls-forgetting*, which leads to separation. Finally, we show that one optimal encoder in design problem 2 does indeed possess this property, which leads to separation and certainty equivalence for this problem.

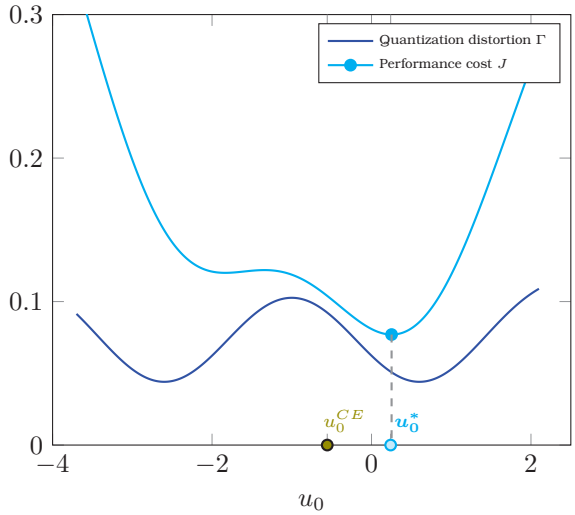


Figure 7.3: Plot of quantization distortion and performance cost for example 7.1

7.4.1 Known Structural Properties of Optimal Encoders

Let us now formulate the encoder's Markov decision problem. Fix the control policies to be the admissible laws:

$$u_t = \mathcal{K}_t^{\dagger}\left(\{z_i\}_0^t\right).$$

Then the optimization problem reduces to one of picking encoding policies. This is a single-agent, sequential decision problem, and hence one with a classical information pattern. The action space for this decision problem is the infinite dimensional function space of discrete-valued encoders. At time t, the encoder takes as input the current and previous states, all previous outputs, and all previous encoding maps. For convenience, we can view this encoding map as a function of only the current state but with the rest of the inputs considered as parameters determining the form of this function. Thus, without loss of generality the encoder can be described as the function

$$\xi_t(\cdot) : \mathbb{R} \to \{1, \ldots, N\}$$

having x_t as its argument with its shape determined by $\left(\{x_i\}_0^{t-1}\{z_i\}_0^{t-1}\{\xi_i(\cdot)\}_0^{t-1}\right)$. Hence the action space at times t can be described as:

$$\left\{\xi(\cdot) : \mathbb{R} \to \{1, \ldots, N\},\ \text{Borel measurable}\right\}.$$

Identifying encoders as decisions to be picked is not enough, as the signal x_t need not be Markov. By identifying an augmented state signal $\Xi_t = \Big(x_t,\ t,\ z_0 \cdot \mathbb{1}_{\{t>0\}},\ \ldots,\ z_i \cdot \mathbb{1}_{\{t>i\}},\ \ldots,\ z_T \cdot \mathbb{1}_{\{t>T\}} \Big)$, it is possible to obtain a controlled Markov process. Then, the encoder's problem can be rewritten as a Markov decision problem with controls taking values in a function space. For such decision problems, sufficient statistics have been laid down by Shiryaev (1964), and by Dynkin (1965) for countable state spaces, and by Striebel (1965) for states and controls taking values in finite dimensional vector spaces. Even though our action spaces are infinite dimensional spaces, Striebel's proof technique holds true here. Hence, the variables $\Big(\Xi_t,\, \xi_0(\cdot), \ldots, \xi_{t-1}(\cdot)\Big)$ are sufficient statistics for the encoder.

Hence, at every time t, performance is not degraded by the encoder choosing to quantize just x_t instead of quantizing the entire waveform $\{x_0, \ldots, x_t\}$. Of course the shape of the quantizer is allowed to vary with past encoder shapes, past encoder outputs, and on past control inputs. But given the sufficient statistics, the encoder can forget the data: $\{x_0, \ldots, x_{t-1}\}$.

7.4.2 The Common Information Approach:

The core problem we consider in this chapter has two decision makers that jointly minimize a given cost function. The information available to these decision makers is not the same, and neither is the information available to each agent a subset of the information available to the agent downstream in the loop. Thus, the information pattern here is neither classical nor nested. We apply the *common information approach* to our problem. This approach was first proposed by Witsenhausen (1971) as an unproven conjecture, to deal with multiple decision makers and non-classical information patterns in a general setting. This was shown to be true by Varaiya and Walrand (1978) for a special case. Our terminology is derived from Nayyar (2011), where this conjecture has been studied in detail.

Denote by $\mathcal{D}_{t^-}^{\text{con}}$ the data at the controller just after it has read the channel output z_t and just before it has generated the control value u_t. Similarly denote by $\mathcal{D}_{t^+}^{\text{con}}$ the data at the controller just after it has generated the control value u_t. Then

$$\mathcal{D}_{t^-}^{\text{con}} = \Big\{ \{z_i\}_0^t, \{\xi_i(\cdot)\}_0^t, \{u_i\}_0^{t-1} \Big\},$$
$$\mathcal{D}_{t^+}^{\text{con}} = \Big\{ \mathcal{D}_{t^-}^{\text{con}}, u_t \Big\} \ = \ \Big\{ \{z_i\}_0^t, \{\xi_i(\cdot)\}_0^t, \{u_i\}_0^{t-1}, u_t \Big\}.$$

Also let $\widehat{x}_{t|t} = \mathbb{E}\left[x_t \,\middle|\, \mathcal{D}_{t^-}^{\text{con}} \right]$.

The common information approach allows a designer to treat a problem with multiple decision makers as a classical control problem with a single decision maker that has access to partial state information. When applied to our setup, this approach leads to the following structural result at the encoder. The encoding policy $\xi_t(\cdot)$ is selected based on the information available to the controller at the pre-

vious time instant namely $\mathcal{D}^{\text{con}}_{(t-1)^+}$. At times t^-, t^+ respectively, the data $\mathcal{D}^{\text{con}}_{(t-1)^-}$, $\mathcal{D}^{\text{con}}_{(t-1)^+}$ comprise the common information in this problem. The encoding map $\xi_t(\cdot)$ is applied to the state x_t, which is private information available to the encoder. This approach has been used by many others, even within the context of quantized control (Borkar et al., 2001b; Walrand and Varaiya, 1983).

7.4.3 Controls-Forgetting Encoders and Separation

We now present a structural property of encoders which ensures separation in design. Recall that our plant evolution is linear: $x_{t+1} = ax_t + u_t + w_t$. Define the following control free part of the state:

$$\zeta_0 = x_0,$$
$$\zeta_{i+1} = x_{i+1} - \sum_{j=0}^{i} a^{i-1} u_j \text{ for } i \geq 0.$$

At the encoder, the change of variables

$$\left(x_t, \{z_i\}_0^{t-1}; \{\mathcal{K}_i(\cdot)\}_0^T\right) \longmapsto \left(\zeta_t, \{z_i\}_0^{t-1}; \{\mathcal{K}_i(\cdot)\}_0^T\right) \tag{7.5}$$

is causal and causally invertible. Hence the statistics $\left(\zeta_t, \{z_i\}_0^{t-1}; \{\mathcal{K}_i(\cdot)\}_0^T\right)$ are also sufficient statistics at the encoder. We now introduce the innovation encoding of Borkar and Mitter (1997).

Definition 7.4 (Innovation encoding)**.** *An encoding map with the inputs and outputs:*

$$\left(\zeta_t, \{z_i\}_0^{t-1}; \{\mathcal{K}_i(\cdot)\}_0^T\right) \longmapsto \iota_t$$

is admissible and is called 'innovation' encoding (Borkar and Mitter, 1997).

The control free part of the state is not affected by the control policies. It obeys the recursion:

$$\zeta_{t+1} = a\zeta_t + w_t.$$

The change of variables (7.5) also induces an one-to-one and onto map from the set of encoders ξ acting on $\left(x_t, \{z_i\}_{i=0}^{t-1}, \{\mathcal{K}_i(\cdot)\}_0^T\right)$ to the set of encoders ξ^{inn} acting on $\left(\zeta_t, \{z_i\}_{i=0}^{t-1}, \{\mathcal{K}_i(\cdot)\}_0^T\right)$. For any sequence of causal encoders, one can find an equivalent sequence of innovation encoders such that when these two sets operate on the same sequence of plant outputs, they produce two sequences of channel inputs that are equal with probability one. Hence, if for a plant and channel, the dual effect is present in a certain class of causal encoders, then the dual effect is also present in the equivalent class of innovation encoders (Feng and Loparo, 1997). This is what the following example illustrates:

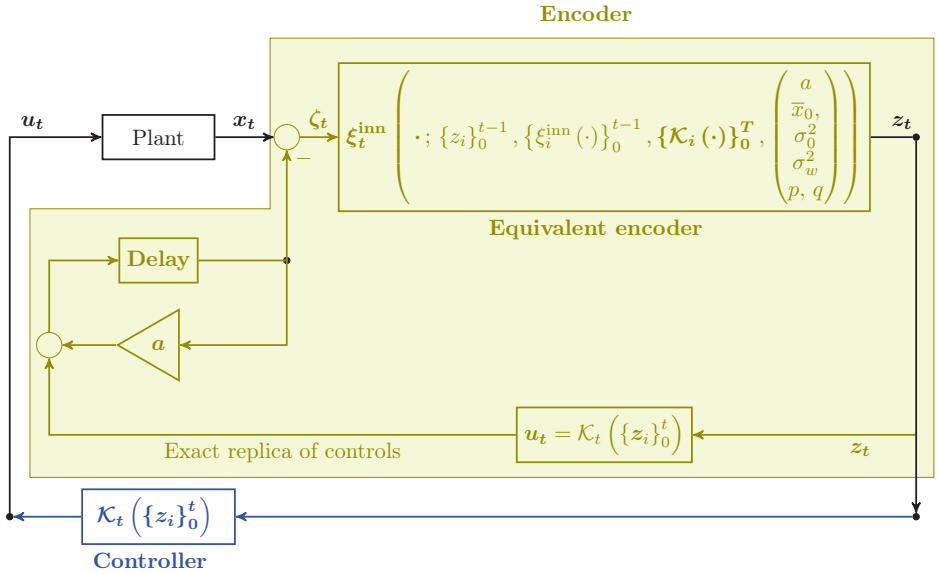


Figure 7.4: The block diagram of Figure 7.1 with innovation encoding

Example 7.2

For the linear plant (7.1), let the coefficient $a = 1$, and let the initial state $x_0 = 2$, and $\sigma_0 = 0$, and let this information be known to the encoder and the controller. This simply means that $z_0 = x_0$. Let the variance $\sigma_w^2 = 0.7^2$. Let the horizon end at $T = 1$, and let the coefficients p and q be equal to 0.01. Let the channel alphabet be $\{1, 2, 3\}$. For the given threshold $\theta = 1.6$, let the encoder at time $t = 1$ be the following innovation encoder:

$$\xi_1^{\text{inn}}(\zeta_1) = \begin{cases} 1 & \text{if } a\zeta_1 + \mathcal{K}_0(z_0) \in (-\infty, -\theta), \\ 2 & \text{if } a\zeta_1 + \mathcal{K}_0(z_0) \in (-\theta, \theta), \\ 3 & \text{if } a\zeta_1 + \mathcal{K}_0(z_0) \in (\theta, +\infty). \end{cases} \tag{7.6}$$

Let the control law at time $t = 1$ give: $u_1 = -\frac{a}{q+1}\widehat{x}_{1|1}$, where $\widehat{x}_{1|1} = \mathbb{E}[x_1 | x_0, u_0, z_1]$. Then the only choice yet to be made is the policy for control u_0.

Notice that all the parameters of this example are the same as those of example 7.1. The only possibly difference may be in the encoding policies. But looking at equations (7.4), (7.6), tells us that this innovation encoder ξ_t^{inn} is equivalent to the causal encoder ξ_t of example 7.1. For the same applied control policy $\mathcal{K}_0$, and for the same realizations of primitive random variables, we get: $\xi_1^{\text{inn}}(\zeta_1(\boldsymbol{\omega})) = \xi_1(x_1(\boldsymbol{\omega}))$. Hence, with probability one the two nonlinear filters for the state given x_0, z_1 are

the same. Thus for an event $X \in \mathcal{F}^{(x_0,z_1)}$, we have:

$$\mathbb{P}\left[x_1 \in X \left| x_0,\, z_1 = \xi_1^{\text{inn}}\left(\zeta_1\right)\right.\right] = \mathbb{P}\left[x_1 \in X \left| x_0,\, z_1 = \xi_1\left(x_1\right)\right.\right].$$

Hence the graphs in Figure 7.3 are identical to corresponding graphs for this example.

We now define special classes of controllers and encoders.

Definition 7.5 (Controls affine from time τ)**.** *We call a controller affine from time τ if it takes the following multiplexed form:*

$$\mathcal{K}_i^{mult,\,\tau}\left(\mathcal{D}_{i^-}^{con}\right) = \begin{cases} u_i^{\dagger}, & \text{if } \; i < \tau, \\ u_i^{aff} = k_i \widehat{x}_{i|i} + d_i, & \text{if } \; i \geq \tau, \end{cases} \tag{7.7}$$

where, the controls $u_i^{\dagger}$ are generated by an admissible strategy $\{\mathcal{K}_i^{\dagger}(\cdot)\}_{i=0}^{T}$, and the controls u_i^{aff} are generated by an affine strategy $\{\mathcal{K}_i^{aff}(\cdot)\}_{i=0}^{T}$. The gains $\{k_i\}_0^T$ and offsets $\{d_i\}_0^T$ must be computed offline and stored in the memory of the controller.

Definition 7.6 (controls-forgetting encoder)**.** *From time τ, denote by $\rho_{\tau|\tau-1}^{\zeta}(\cdot)$ the conditional density of ζ_τ given the data $\mathcal{D}_{(\tau-1)^-}^{con}$. We call an admissible encoding strategy controls-forgetting from time τ if it takes the form:*

$$\xi_t^{CF,\,\tau}\left(x_t; \mathcal{D}_{(t-1)^-}^{con}\right) = \begin{cases} \xi_t^{\dagger}\left(x_t; \mathcal{D}_{(t-1)^-}^{con}\right), & \text{if } t \leq \tau, \\ \epsilon_t\left(\zeta_t;\, \rho_{\tau|\tau-1}^{\zeta}\left(\cdot\right),\, \{z_i\}_{\tau}^{t-1}, \{\epsilon_i\left(\cdot\right)\}_{\tau}^{t-1}\right), & \text{if } t \geq \tau+1, \end{cases}$$

where: (1) $\xi_t^{\dagger}\left(\cdot; \mathcal{D}_{(t-1)^-}^{con}\right)$ is any admissible policy for encoding at time t, (2) for times $t \geq \tau+1$ the policies $\epsilon_t\left(\cdot; \rho_{\tau|\tau-1}^{\zeta}\left(\cdot\right), \{z_i\}_{\tau}^{t-1}, \{\epsilon_i\left(\cdot\right)\}_{\tau}^{t-1}\right)$ are adapted to the data

$$\mathcal{D}_{(t-1)^+}^{CF,\,\tau} = \left(\rho_{\tau|\tau-1}^{\zeta}\left(\cdot\right), \{z_i\}_{\tau}^{t-1}, \{\epsilon_i\left(\cdot\right)\}_{\tau}^{t-1}\right) \;\subset\; \mathcal{D}_{(t-1)^+}^{con}, \quad \text{for } t \geq \tau,$$

and (3) for fixed values of the data $\mathcal{D}_{(t-1)^+}^{CF,\,\tau}$, the map $\epsilon_t(\cdot)$ produces the same output no matter what the controls $\{u_i\}_{\tau}^{t}$ are, and no matter what the control policies $\{\mathcal{K}_i(\cdot)\}_{t+1}^{T}$ are.

Clearly such controls-forgetting encoders exist. For example: a set of encoders that quantize in sequence $\zeta_{\tau+1}, \ldots, \zeta_T$ to minimize the estimation distortion

$$\sum_{i=\tau+1}^{T} \mathbb{E}\Big[\left(\zeta_i - \widehat{\zeta}_{i|i}\right)^2\Big],$$

where $\widehat{\zeta}_{i|i} = \mathbb{E}\left[\zeta_i \left| \mathcal{D}_{(i-1)^+}^{CF,\,\tau}\right.\right]$. Let the non-negative function $\psi(\cdot)$ represent some notion of cost. For example: $\psi(x) := x^2$.

Lemma 7.2 (Distortions incurred by CF encoders also forget controls)**.** *Fix the distortion measure ψ. If the encoder is controls-forgetting from time τ, then for times $i \geq \tau + 1$, the conditional expected distortions*

$$\mathbb{E}\left[\psi\left(x_i - \widehat{x}_{i|i}\right) \middle| \mathcal{D}_{i^-}^{con}\right]$$

are statistically independent of the partial set of controls $\{u_i\}_{i=\tau}^{T}$.

Proof. The unconditional statistics of $\{\zeta_t\}$ are independent of the entire control waveform, no matter what the encoder is. For times $i \geq \tau + 1$ and for sets $X \in \mathcal{F}_i^z$ the conditional probabilities

$$\mathbb{P}\left[\zeta_i \in X \middle| \mathcal{D}_{i^-}^{\text{con}}\right]$$

are independent of $\{u_i\}_{i=\tau}^{T}$ because the encoding maps ξ_i are controls-forgetting from time τ. Since $\zeta_t - \widehat{\zeta}_{t|t} = x_t - \widehat{x}_{t|t}\ \forall t$, the lemma follows. □

7.4.4 Preliminary Lemmas

The main result ahead is Theorem 7.7 that states that it is optimal for the design problem 2 to apply a separated design and certainty equivalence controls. In this subsection, we do some necessary ground work towards proving that theorem.

Once we are prescribed an admissible encoder, the controls $\{u_j\}_{j=i}^{T}$ affect only the cost to go: $\mathbb{E}\left[x_{T+1}^2\right] + \sum_{j=i}^{T} \mathbb{E}\left[px_j^2 + qu_j^2\right]$. In the classical single agent LQ problem, the 'prescribed encoder' is simply the linear observation process with prescribed signal-to-noise ratios. There, this cost to go can be expressed as a quadratic function of $\{u_j\}_{j=i}^{T}, \{x_j\}_{j=i}^{T}$ and $\left\{\widehat{x}_{j|j}\right\}_{j=i}^{T}$. But in our two agent LQ problem, because of the dual effect, the cost to go may have a non-quadratic dependence on the controls $\{u_j\}_{j=i}^{T}$. However we show that by restricting the controls-forgetting encoders and affine controls the above cost to go does get a quadratic form. We use this reasoning and dynamic programming to show that for time $t = i$ going backwards from T:

- it is optimal at time $t = i$ to apply as control a linear function of $\widehat{x}_{i|i}$, and,
- it is optimal at time $t = i$ to apply as encoding map one that is controls-forgetting from time $i - 1$.

Lemma 7.3 (Optimal control at time $t = T$)**.** *The best control policy at time $t = T$ is the linear law: $u_T^* = -\frac{a}{1+q}\widehat{x}_{T|T}$, and the optimum cost to go $V_T^*\left(\mathcal{D}_{T^-}^{con}\right) = \min_{u_t} \mathbb{E}\left[x_{T+1}^2 + qu_T^2 \middle| \mathcal{D}_{T^-}^{con}\right]$ is the expected value of a quadratic in x_T and $\widehat{x}_{T|T}$.*

Proof. At time T^-, one is given $\mathcal{D}_{T^-}^{\text{con}}$, and is asked to pick u_T to minimize the cost to go

$$\begin{aligned} V_T\left(u_T; \mathcal{D}_{T^-}^{\text{con}}\right) &= \mathbb{E}\left[x_{T+1}^2 + q u_T^2 \,|\mathcal{D}_{T^-}^{\text{con}}\right], \\ &= \sigma_w^2 + \mathbb{E}\left[a^2 x_T^2 + 2a x_T u_T + (1+q)\, u_T^2 \,|\mathcal{D}_{T^-}^{\text{con}}\right], \\ &= \sigma_w^2 + \frac{a^2}{1+q}\mathbb{E}\left[\, q\, x_T^2 + \left(x_T - \widehat{x}_{T|T}\right)^2 \middle| \, \mathcal{D}_{T^-}^{\text{con}}\right] \\ &\qquad + (1+q)\left(u_T - \frac{a}{1+q}\widehat{x}_{T|T}\right)^2, \end{aligned}$$

and this lets us prove the lemma. □

Lemma 7.4 (Optimal ξ_i for separated, quadratic cost to go)**.** *Consider the dynamic encoder-controller design problem (Design problem 2), for the linear plant eqn. (7.1), and the performance cost of eqn. (7.2). Suppose that we apply an admissible controller $\widetilde{\mathcal{K}}$ along with an encoder $\xi_t^{CF,i}$ that is controls-forgetting from time i. Furthermore, suppose that the partial sets of policies:*

$$\begin{gathered} \left\{\xi_{i+1}^{CF,i}(\cdot), \ldots, \xi_T^{CF,i}(\cdot)\right\} \\ \left\{\widetilde{\mathcal{K}}_i(\cdot), \widetilde{\mathcal{K}}_{i+1}(\cdot), \quad \ldots, \widetilde{\mathcal{K}}_T(\cdot)\right\} \end{gathered}$$

are chosen such that the following three properties hold:

1. *the cost to go at time i takes the separated form:*

$$\mathbb{E}\left[x_{T+1}^2 + p\sum_{j=i}^{T} x_i^2 + q\sum_{j=i}^{T} u_i^2 \,\middle|\, \mathcal{D}_{i^+}^{con}\right] = \mathbb{E}\left[\, J_i^{con}(u_i, x_i)\right| \mathcal{D}_{i^+}^{con}] + \mathbb{E}\left[\Gamma_{i+1}\right| \mathcal{D}_{i^+}^{con}],$$

   where, $J_i^{con}(u_i, x_i) = \overline{\alpha} + \alpha\,\sigma_w^2 + \overline{\beta}\, x_i + \widetilde{\beta}\, x_i^2 + \overline{\nu}\,\widehat{x}_{i|i} + \widehat{\nu}\, x_i \widehat{x}_{i|i} + \widetilde{\nu}\,\widehat{x}_{i|i}^2$,

   and the term Γ_{i+1} is a weighted sum of future distortions and depends only on the random sequence $\left\{x_j - \widehat{x}_{j|j}\right\}_{j=i+1}^{T}$,

2. *the coefficients of the quadratic J_i^{con} may depend on the control policies $\left\{\widetilde{\mathcal{K}}_j(\cdot)\right\}_i^T$ but not on the partial set of encoding maps: $\left\{\xi_j^{CF,i}(\cdot)\right\}_{i+1}^T$ and,*

3. *the term Γ_{i+1} depends on the encoding maps $\left\{\xi_j^{CF,i}(\cdot)\right\}_{i+1}^T$ but not on the partial set of control policies $\left\{\widetilde{\mathcal{K}}_j(\cdot)\right\}_i^T$.*

Then, it follows that the best choice for the encoding map at time $t = i$ does not depend on the data: $\left(u_{i-1},\, \left\{\widetilde{\mathcal{K}}_j\left(\cdot\right)\right\}_i^T\right)$. It also follows that the shapes of the encoding maps $\left\{\xi_j^{CF,i}\left(\cdot\right)\right\}_{i+1}^T$ and their performance do not depend on the control u_{i-1}.

Proof. The proof exploits three facts: Firstly the special form of $J_i^{\mathrm{con}}\left(u_i, x_i\right)$ makes the encoder's performance cost at time i a sum of a quadratic distortion between x_i and $\widehat{x}_{i|i}$, and a term gathering distortions at later times. Secondly the minimum of the sum distortion depends only on the intrinsic shape of the conditional density $\rho_{i|i-1}\left(\cdot\right)$ and not on its mean. Thirdly, these facts and the controls-forgetting nature of later encoding maps allows the encoder to 'ignore' the control u_{i-1}. We now start by writing the cost to go as:

$$\begin{aligned}
\mathbb{E}\left[\, J_i^{\mathrm{con}}\left(u_i, x_i\right) + \Gamma_{i+1} \middle|\, \mathcal{D}_{i^+}^{\mathrm{con}}\right], &= \mathbb{E}\left[\, \overline{\alpha} + \alpha\, \sigma_w^2 + \overline{\beta}\, x_i + \widetilde{\beta}\, x_i^2 + \overline{\nu}\, \widehat{x}_{i|i} \middle|\, \mathcal{D}_{i^+}^{\mathrm{con}}\right], \\
&\quad \mathbb{E}\left[\, + \widehat{\nu}\, x_i\, \widehat{x}_{i|i} + \widetilde{\nu}\, \widehat{x}_{i|i}^2 \middle|\, \mathcal{D}_{i^+}^{\mathrm{con}}\right], + \mathbb{E}\left[\, \Gamma_{i+1} \middle|\, \mathcal{D}_{i^+}^{\mathrm{con}}\right], \\
&= \overline{\alpha} + \alpha\, \sigma_w^2 + \mathbb{E}\left[\, \left(\overline{\beta} + \overline{\nu}\right) x_i + \left(\widehat{\nu} + \widetilde{\nu} + \widetilde{\beta}\right) x_i^2 \middle|\, \mathcal{D}_{i^+}^{\mathrm{con}}\right] \\
&\quad - \left(\widehat{\nu} + \widetilde{\nu}\right) \mathbb{E}\left[\, x_i^2 - \widehat{x}_{i|i}^2 \middle|\, \mathcal{D}_{i^+}^{\mathrm{con}}\right] + \mathbb{E}\left[\, \Gamma_{i+1} \middle|\, \mathcal{D}_{i^+}^{\mathrm{con}}\right], \\
&= \overline{\alpha} + \alpha\, \sigma_w^2 + \mathbb{E}\left[\, \left(\overline{\beta} + \overline{\nu}\right) x_i + \left(\widehat{\nu} + \widetilde{\nu} + \widetilde{\beta}\right) x_i^2 \middle|\, \mathcal{D}_{i^+}^{\mathrm{con}}\right] \\
&\quad - \left(\widehat{\nu} + \widetilde{\nu}\right) \mathbb{E}\left[\, \left(x_i - \widehat{x}_{i|i}\right)^2 \middle|\, \mathcal{D}_{i^+}^{\mathrm{con}}\right] + \mathbb{E}\left[\, \Gamma_{i+1} \middle|\, \mathcal{D}_{i^+}^{\mathrm{con}}\right].
\end{aligned}$$

Given the data $\mathcal{D}_{(i-1)^+}^{\mathrm{con}}$ the part of the cost above that depends on the encoding map $\xi_i\left(\cdot\right)$ is

$$- \left(\widehat{\nu} + \widetilde{\nu}\right) \mathbb{E}\left[\, \left(x_i - \widehat{x}_{i|i}\right)^2 \middle|\, \mathcal{D}_{i^+}^{\mathrm{con}}\right] + \mathbb{E}\left[\, \Gamma_{i+1} \middle|\, \mathcal{D}_{i^+}^{\mathrm{con}}\right].$$

Notice that the first term is the quantization variance of the quantizer $\xi_i\left(\cdot\right)$. This reduction of the encoder's performance cost to a sum of current and future quantization distortions is possible because the term $J_i^{\mathrm{con}}\left(u_i, x_i\right)$ has been assumed to be quadratic in x_i and $\widehat{x}_{i|i}$. The reduced performance cost of the encoder is a function only of the quantizer $\xi_i\left(\cdot\right)$ and the conditional density $\rho_{\,i|i-1}\left(x \middle| \mathcal{D}_{(i-1)^-}^{\mathrm{con}}\right)$. Indeed, given the data $\mathcal{D}_{(i-1)^-}^{\mathrm{con}}$ this cost is the following average:

$$\begin{aligned}
&\Gamma_i\left(\xi_i\left(\cdot\right); \mathcal{D}_{(i-1)^+}^{\mathrm{con}}\right) \\
&\quad = \sum_{\text{cells } \Delta} \mathbb{P}\left[x_i \in \Delta \middle| \mathcal{D}_{(i-1)^-}^{\mathrm{con}}, u_{i-1}\right] \cdot \left\{\mathbb{E}\left[\Gamma_{i+1}\left(\mathcal{D}_{i^-}^{\mathrm{con}}\right) \middle| \mathcal{D}_{(i-1)^+}^{\mathrm{con}}, x_i \in \Delta\right]\right\} \\
&\quad + \sum_{\text{cells } \Delta} \mathbb{P}\left[x_i \in \Delta \middle| \mathcal{D}_{(i-1)^-}^{\mathrm{con}}, u_{i-1}\right] \cdot \left\{\lambda\, \mathbb{E}\left[\left(x_i - \widehat{x}_{i|i}\right)^2 \middle| \mathcal{D}_{(i-1)^+}^{\mathrm{con}}, x_i \in \Delta\right]\right\},
\end{aligned}$$

where $\lambda = -(\widehat{\nu} + \widetilde{\nu})$. The cost Γ_i does depend on both $\xi_i(\cdot)$ and u_i, but for given data $\mathcal{D}^{\text{con}}_{(i-1)^-}$ and control u_{i-1}, the minimum of Γ_i over all admissible quantizers $\xi_i(\cdot)$ may possibly depend on $\mathcal{D}^{\text{con}}_{(i-1)^-}$ but not on the control u_{i-1}. To see this consider two arbitrary possible values $u, \widetilde{u}$ for u_{i-1}. Suppose that one is given the quantizer

$$\xi(x) = \begin{cases} 1 & \text{if } x \in (-\infty, \delta_1), \\ 2 & \text{if } x \in (\delta_1, \delta_2), \\ \vdots & \vdots \\ N & \text{if } x \in (\delta_{N-1}, +\infty), \end{cases}$$

meant for quantizing a random variable with the density $\rho_{i|i-1}\left(x \left| \mathcal{D}^{\text{con}}_{(i-1)^-}, u_{i-1} = u \right.\right)$. Consider the quantizer $\widetilde{\xi}$ constructed by taking each cell $\Delta = \left(\underline{\delta}, \overline{\delta}\right)$ in ξ, and generating a new cell $\widetilde{\Delta} = \left(\underline{\delta} - u + \widetilde{u}, \overline{\delta} - u + \widetilde{u}\right)$, and stipulating that the new quantizer $\widetilde{\xi}$ assigns to the cell $\widetilde{\Delta}$ the same channel input that the quantizer ξ assigns to Δ.

Because of the linear evolution: $x_i = ax_{i-1} + u_{i-1} + w_{i-1}$, and because the random variable w_{i-1} is independent of the data $\mathcal{D}^{\text{con}}_{(i-1)^+}$, we have the convolution relations:

$$\rho(x) = \rho_{i|i-1}\left(\frac{\cdot - u}{a}\right) \circledast \rho_w(\cdot)\Big|_x \text{ and,}$$
$$\widetilde{\rho}(x) = \rho_{i|i-1}\left(\frac{\cdot - \widetilde{u}}{a}\right) \circledast \rho_w(\cdot)\Big|_x,$$

leading to the following symmetry w.r.t. translations:

$$\rho_{i|i-1}\left(x - u \left| \mathcal{D}^{\text{con}}_{(i-1)^-}, u_{i-1} = u \right.\right) = \rho_{i|i-1}\left(x - \widetilde{u} \left| \mathcal{D}^{\text{con}}_{(i-1)^-}, u_{i-1} = \widetilde{u} \right.\right). \tag{7.8}$$

Then we get the following equalities for each pair of cells $\Delta, \widetilde{\Delta}$

$$\mathbb{P}\left[x_i \in \Delta \left| \mathcal{D}^{\text{con}}_{(i-1)^-}, u_{i-1} = u \right.\right] = \mathbb{P}\left[x_i \in \widetilde{\Delta} \left| \mathcal{D}^{\text{con}}_{(i-1)^-}, u_{i-1} = \widetilde{u} \right.\right],$$
$$\Gamma_{i+1}\Big(\mathcal{D}^{\text{con}}_{(i-1)^-}, u_{i-1} = u, , x_i \in \Delta\Big) = \Gamma_{i+1}\Big(\mathcal{D}^{\text{con}}_{(i-1)^-}, u_{i-1} = \widetilde{u}, , x_i \in \widetilde{\Delta}\Big),$$
$$\mathbb{E}\left[\left(x_i - \widehat{x}_{i|i}\right)^2 \left| \mathcal{D}^{\text{con}}_{(i-1)^-}, u_{i-1} = u, , x_i \in \Delta \right.\right] =$$
$$\mathbb{E}\big[\left(x_i - \widehat{x}_{i|i}\right)^2 | \mathcal{D}^{\text{con}}_{(i-1)^-}, u_{i-1} = \widetilde{u}, x_i \in \widetilde{\Delta}.\big].$$

Then the performance of any quantizer ξ designed for $u_{i-1} = u$ can be matched by $\widetilde{\xi}$ for $u_{i-1} = \widetilde{u}$, and vice versa. Hence, we can conclude that for any $u, \widetilde{u}$,

$$\inf_{\xi} \Gamma_i\Big(\xi(\cdot); \mathcal{D}^{\text{con}}_{(i-1)^-}, u_{i-1} = u, \Big) = \inf_{\xi} \Gamma_i\Big(\xi(\cdot); \mathcal{D}^{\text{con}}_{(i-1)^-}, u_{i-1} = \widetilde{u}, \Big).$$

Notice that this optimal encoder now become controls-forgetting from time $i-1$. □

As the optimal control u_T^* is a linear function on $\widehat{x}_{T|T}$, the encoder ξ_T begets a performance cost that is quadratic in $x_T, \widehat{x}_{T|T}$. Then the above lemma renders the optimal encoding map ξ_T^* to be controls-forgetting from time $T-1$. This reduction also holds at earlier times.

Lemma 7.5 (Encoder separation for affine controls)**.** *If: (1) for any admissible control strategy, an admissible encoder strategy minimizing the performance cost (7.2) exists, and (2) we apply as control strategy one affine from time τ: $\mathcal{K}_i^{mult,\tau}\left(\mathcal{D}_{i^+}^{con}\right)$ (from defn. 7.5), then: (a) an encoder that is controls-forgetting from time τ minimizes the partial LQ cost:*

$$\mathbb{E}\left[x_{T+1}^2 + p\sum_{i=\tau+1}^{T} x_i^2 + q\sum_{i=\tau}^{T} u_i^2 \,\middle|\, \mathcal{D}_{\tau^+}^{con}\right],$$

and, (b) the shapes of the minimizing encoding maps from time τ and their performance are independent of the data: $\left\{u_{\tau-1}^{\dagger}, \{k_i\}_{i=\tau}^{T}, \{d_i\}_{i=\tau}^{T}\right\}$.

Proof. We prove by mathematical induction. For a given control strategy, define:

$$W_T = \mathbb{E}\left[x_{T+1}^2 + px_T^2 + qu_T^2 \middle|\, \mathcal{D}_{(T-1)^+}^{\text{con}}\right], \qquad W_T^* = \inf_{\xi_T(\cdot)} W_T,$$

$$W_i = \mathbb{E}\left[px_i^2 + qu_i^2 \middle|\, \mathcal{D}_{(i-1)^+}^{\text{con}}\right] + \mathbb{E}\left[W_{i+1}^*\left(\mathcal{D}_{i^+}^{\text{con}}\right) \middle| \mathcal{D}_{(i-1)^+}^{\text{con}}, \xi_i(\cdot)\right], \quad W_i^* = \inf_{\xi_i(\cdot)} W_i.$$

Induction hypothesis. Induction hypothesis*[for time i] For some time $t = i$ such that $\tau \leq i < T$, we have the following three facts: (1) for every $j \geq i+1$, the optimal value function $W_j^*\left(\mathcal{D}_{(j-1)^-}^{con}\right)$ takes the form:*

$$\alpha_j\sigma_w^2 + \overline{\alpha}_j + \widetilde{\beta}_j\mathbb{E}\left[{x_j}^2 \middle| \mathcal{D}_{j^-}^{con}\right] + \overline{\beta}_j\,\widehat{x}_{j|j} + \mathbb{E}\left[\widetilde{\Gamma}_{j+1}^*\left(\mathcal{D}_{(j+1)^-}^{con}\right) \middle| \mathcal{D}_{j^-}^{con}\right] + \widetilde{\lambda}_j\mathbb{E}\left[\left(x_j - \widehat{x}_{j|j}\right)^2 \middle| \mathcal{D}_{j^-}^{con}\right],$$

where the $\alpha_j, \overline{\alpha}_j, \widetilde{\beta}_j, \overline{\beta}_j, \widetilde{\lambda}_j$ are known non-negative real numbers for $j \geq i+1$, (2) for each such j, the non-negative function $\widetilde{\Gamma}_{j+1}^*\left(\mathcal{D}_{j^-}^{con}\right)$ is assumed to be independent of the partial waveform $\{u_j, u_{j+1}, \ldots, u_T\}$, and (3) the optimal partial set of encoding maps $\left\{\xi_j^*(\cdot)\right\}_{i+1}^{T}$ is a set that is controls-forgetting from time i.

We will now show: if this hypothesis holds for time i, then it holds for time $i-1$.

Assuming that the partial set of encodings maps $\left\{\xi_j^*(\cdot)\right\}_{i+1}^{T}$ are employed, we get:

$$\begin{aligned}
W_i &= \mathbb{E}\left[px_i^2 + qu_i^2 \middle| \mathcal{D}_{(i-1)^+}^{\text{con}}\right] + \mathbb{E}\left[W_{i+1}^*\left(\mathcal{D}_{i^+}^{\text{con}}\right) \middle| \mathcal{D}_{(i-1)^+}^{\text{con}}, \xi_i(\cdot)\right], \\
&= p\,\mathbb{E}\left[x_i^2 \middle| \mathcal{D}_{(i-1)^+}^{\text{con}}\right] + q\,\mathbb{E}\left[u_i^2 \middle| \mathcal{D}_{(i-1)^+}^{\text{con}}\right] + \alpha_{i+1}\sigma_w^2 + \overline{\alpha}_{i+1} \\
&\quad + \widetilde{\beta}_{i+1}\mathbb{E}\left[{x_{i+1}}^2 \middle| \mathcal{D}_{(i+1)^-}^{\text{con}}\right] + \overline{\beta}_{i+1}\mathbb{E}\left[x_{i+1} \middle| \mathcal{D}_{(i+1)^-}^{\text{con}}\right] \\
&\quad + \mathbb{E}\left[\widetilde{\Gamma}_{i+1}^*\left(\mathcal{D}_{(i+1)^-}^{\text{con}}\right) \middle| \mathcal{D}_{i^-}^{\text{con}}\right], \\
&= \alpha_i\sigma_w^2 + \overline{\alpha}_i + \widetilde{\beta}_i\mathbb{E}\left[x_i^2 \middle| \mathcal{D}_{i^-}^{\text{con}}\right] + \overline{\beta}_i\mathbb{E}\left[x_i \middle| \mathcal{D}_{i^-}^{\text{con}}\right] \\
&\quad + \mathbb{E}\left[\widetilde{\Gamma}_{i+1}^*\left(\mathcal{D}_{(i+1)^-}^{\text{con}}\right) \middle| \mathcal{D}_{i^-}^{\text{con}}\right] + \widetilde{\lambda}_i\mathbb{E}\left[\left(x_i - \widehat{x}_{i|i}\right)^2 \middle| \mathcal{D}_{i^-}^{\text{con}}\right],
\end{aligned}$$

where, the coefficients:

$$\begin{aligned}
\alpha_i &= \alpha_{i+1} + \widetilde{\beta}_{i+1}, \\
\overline{\alpha}_i &= \overline{\alpha}_{i+1} + \widetilde{\beta}_{i+1}d_i^2 + q\,d_i^2 + \overline{\beta}_{i+1}d_i, \\
\overline{\beta}_i &= 2\left(q\,k_i d_i + a\widetilde{\beta}_{i+1}d_i + \widetilde{\beta}_{i+1}k_i d_i\right), \\
\widetilde{\beta}_i &= p_i + a^2\widetilde{\beta}_{i+1} + k_i^2\widetilde{\beta}_{i+1} + 2ak_i\widetilde{\beta}_{i+1} + q\,k_i^2\widetilde{\beta}_{i+1}, \\
\widetilde{\lambda}_i &= q\,k_i^2 + k_i^2\widetilde{\beta}_{i+1} + 2ak_i\widetilde{\beta}_{i+1}.
\end{aligned}$$

We have thus: $W_i = \mathbb{E}\left[\text{A quadratic in } x_i, \widehat{x}_{i|i}\right] + \mathbb{E}\left[\text{ Future distortions }\right]$. This and the fact that the encoder is controls-forgetting from time $t = i$ meet the requirements of lemma 7.4. Then we get the optimal encoding map ξ_i^* to be controls-forgetting from time $t = i - 1$, and

$$\widetilde{\Gamma}_i = \min_{\xi}\mathbb{E}\left[\widetilde{\Gamma}_{i+1}^*\left(\mathcal{D}_{(i+1)^-}^{\text{con}}\right) \middle| \mathcal{D}_{i^-}^{\text{con}}\right] + \widetilde{\lambda}_i\mathbb{E}\left[\left(x_i - \widehat{x}_{i|i}\right)^2 \middle| \mathcal{D}_{i^-}^{\text{con}}\right]$$

is independent of the partial set of controls $\{u_j\}_{j=i-1}^{T}$. From this it follows that the induction hypothesis is also true for time $i - 1$. □

Lemma 7.6 (CE controls for CF encoders)**.** *If the encoder is preassigned to be one that is controls-forgetting from time τ, then the partial LQ cost*

$$\mathbb{E}\left[x_{T+1}^2 + p\sum_{i=\tau+1}^{T} x_i^2 + q\sum_{i=\tau}^{T} u_i^2 \middle| \mathcal{D}_{\tau^-}^{con}\right],$$

is minimized by the following control laws with a linear form: For $i \geq \tau$: $u_i^* = k_i^*\,\widehat{x}_{i|i}$.

Proof. Define the following cost to go at time $t = T - 1$:

$$V_{T-1} = \mathbb{E}\left[W_T\left(\epsilon_T(\cdot); \mathcal{D}_{(T-1)^+}^{\text{con}}\right)\right].$$

Because of lemma 7.3,

$$V_{T-1} = \sigma_w^2 + \left(p + \frac{a^2 q}{q+1}\right) \mathbb{E}\left[x_T^2 \left| \mathcal{D}^{\text{con}}_{(T-1)^-}, u_{T-1}\right.\right] + \mathbb{E}\left[\left(x_T - \widehat{x}_{T|T}\right)^2 \middle| \mathcal{D}^{\text{con}}_{T^-}\right].$$

Because the encoder is controls-forgetting from time τ, the last term, which is the distortion due to the encoder ξ_T, is independent of the partial set of controls $\{u_i\}_{i=\tau+1}^{T}$. Hence the only part of V_{T-1} that depends on the control u_{T-1} is the quadratic

$$\begin{aligned}
&q\,u^2 + \left(p + \frac{a^2 q}{q+1}\right) \mathbb{E}\left[x_T^2 \left| \mathcal{D}^{\text{con}}_{(T-1)^-}, u_{T-1}\right.\right] \\
&= q\,u^2 + \left(p + \frac{a^2 q}{q+1}\right) \left\{a^2 \mathbb{E}\left[x_T^2 \left| \mathcal{D}^{\text{con}}_{(T-1)^-}, u_{T-1}\right.\right]\right\} \\
&+ \left(p + \frac{a^2 q}{q+1}\right) \left\{2a\,\widehat{x}_{T-1|T-1}\; u_{T-1} + u_{T-1}^2 + \sigma_w^2.\right\}
\end{aligned}$$

Hence the best control law is: $u^*_{T-1} = -\frac{a\left(p+\frac{a^2 q}{q+1}\right)}{q+p+\frac{a^2 q}{q+1}} \widehat{x}_{T-1|T-1}$, and the resulting value function:

$$\begin{aligned}
V^*_{T-1} = &\left(1 + p + \frac{a^2 q}{q+1}\right) \sigma_w^2 + \frac{a^2 q\left(p+\frac{a^2 q}{q+1}\right)}{q+p+\frac{a^2 q}{q+1}} \mathbb{E}\left[x_T^2 \left| \mathcal{D}^{\text{con}}_{(T-1)^-}, u_{T-1}\right.\right] \\
&+ \frac{a^2\left(p+\frac{a^2 q}{q+1}\right)}{q+p+\frac{a^2 q}{q+1}} \mathbb{E}\left[\left(x_{T-1} - \widehat{x}_{T-1|T-1}\right)^2 \middle| \mathcal{D}^{\text{con}}_{(T-2)^+}\right] \\
&+ \mathbb{E}\left[\left(x_T - \widehat{x}_{T|T}\right)^2 \middle| \mathcal{D}^{\text{con}}_{T^-}\right].
\end{aligned}$$

Repeating this procedure backwards in time, we get for times $i \geq \tau$, the optimal control laws are: $u_i^* = -k_i^* \widehat{x}_{i|i}$, where $k_i^* = a\frac{\beta_{i+1}}{q+\beta_{i+1}}$, $\beta_i = p + \frac{a^2 q \beta_{i+1}}{q+\beta_{i+1}}$, $\beta_{T+1} = 1$. □

7.4.5 Main Theorem

Lemma 7.5 implies that for a pre-assigned controller affine from time zero, there exist optimal encoding maps that are controls-forgetting from time zero. Lemma 7.6 is complementary. It implies that for a pre-assigned encoder that is controls forgetting from time zero, the optimal control laws have linear forms.

For Design problem 2 an optimal pair of strategies have a similar simplified structure. It is optimal to apply a combination of controls-forgetting encoding and control laws linear in $\hat{x}_{i|i}$. In general, this controls forgetting encoder does not minimize the aggregate squared estimation error. The goal accomplished by an optimal encoder is slightly different. It is to minimize a sum of state estimation errors with the time-varying weights λ_i.

Theorem 7.7 (Optimality of separation and certainty equivalence)**.** *For Design problem 2, with the discrete alphabet channel of constant alphabet size, the quadratic performance cost (7.2) is minimized by applying the linear control laws*

$$u_t^* = -k_t^* \, \widehat{x}_{\,t|t} \tag{7.9}$$

in combination with the following encoder which is controls-forgetting from time 0*:*

$$\epsilon_t^* \left(\zeta_t \, ; \{z_i\}_0^{t-1}, \{\epsilon_i(\cdot)\}_0^{t-1} \right) = \arg\inf_{\epsilon(\cdot)} \Gamma_i \Big(\epsilon(\cdot) \, ; \{z_i\}_0^{t-1}, \{\epsilon_i(\cdot)\}_0^{t-1} \Big), \tag{7.10}$$

where, $k_i^* = a \frac{\beta_{i+1}}{q+\beta_{i+1}}, \beta_i = p + \frac{a^2 q \beta_{i+1}}{q+\beta_{i+1}}, \beta_{T+1} = 1,$ *and* $\lambda_i = \frac{a^2 \beta_{i+1}^2}{q+\beta_{i+1}}$ *and where,*

$$\Gamma_t = \lambda_t \, \mathbb{E}\left[\left(\zeta_t - \widehat{\zeta}_{t|t} \right)^2 \Big| \epsilon_t(\cdot), \mathcal{D}^{con}_{(t-1)^+} \right] + \mathbb{E}\left[\Gamma_{t+1}^* \left(\overline{x}_0, \sigma_0^2, \{z_i\}_0^t, \{\epsilon_i(\cdot)\}_0^t \right) \right],$$

$$\Gamma_T = \mathbb{E}\left[\left(\zeta_T - \widehat{\zeta}_{T|T} \right)^2 \Big| \epsilon_T(\cdot), \overline{x}_0, \sigma_0^2, \{z_i\}_0^{T-1}, \{\epsilon_i(\cdot)\}_0^{T-1} \right],$$

$$\Gamma_t^* = \inf_{\epsilon(\cdot)} \Gamma_t(\epsilon).$$

Moreover, this control law is a certainty equivalence law.

Proof. Starting with the result of lemma 7.3 as a seed, repeatedly apply in sequence theorems 7.5, 7.6. This proves optimality of the above combination. Lemma 7.1 implies that the controls laws of 7.9 are indeed certainty equivalence control laws as per van de Water and Willems (1981). □

The optimal controller splits into a least square estimator computing $\widehat{x}_{\,t|t}$ and a time-dependent gain. Computing $\widehat{x}_{\,t|t}$ is intrinsically hard because quantization is a nonlinear operation. If one ignores this computational burden, then, at least formally, the optimal controller resembles that for the classical LQG optimal control problem.

Note that in general the sequence of weights $\{\lambda_i\}_0^T$ depends on the parameters of the performance cost including the control penalty coefficient q. In the two special cases:

1. the coefficients $q = 0$, $p = 1$, or
2. the quantity $p + a^2 q - q > 0$ and the following equality holds:

$$p + a^2 q - q + \sqrt{(p + a^2 q - q)^2 + 4pq} = 2,$$

it turns out that the weights $\beta_i \equiv 1 \, \forall i$, and hence the weights $\lambda_i \equiv \frac{a^2}{q+1} \, \forall i$. Thus in these special cases, optimal encoders 'ignore' the parameters of the performance cost and simply minimize the usual aggregate squared error in state estimation.

7.5 Dynamic Designs for Other Channel Models

Clearly, our results for Design problem 2 extend to the case where we allow deterministic, time-varying coefficients for the plant equation and of the quadratic performance costs. These results also apply to the case where the quantizer word-lengths at different times are deterministic but time-varying. In this section, we sketch how these results also apply to some wider settings. These include the case where the quantizer word-lengths are chosen real-time by the encoder, as well as situations where the bandwidth constraint takes different forms.

7.5.1 Kinds of Channel Models

We study a handful of channel models, but all coming from within three broad classes of messaging a sequence of real numbers. These are: (1) quantized messaging, (2) unquantized but irregular, event-triggered sampling, and (3) unquantized but noise added messaging. What is common to our channel models is that the dynamic LQ design problem gets a separated optimal solution. To obtain this design simplification, we also assume that at all times, the channel output is perfectly visible to the encoder. Thus in each one of our channel models, there will be an ideal, delay-free feedback channel copying the actual inputs for the controller back to the encoder.

Ideal, discrete-alphabet channel

The first model is the instantaneous, error-free, discrete-alphabet channel. The logarithm of the size of the alphabet is the bit rate. A variation is a channel which, on each channel use provides instantaneous, error-free transmission of words of varying but bounded length.

Ideal continuous-valued channel with limited use

The second model is suitable only with systems working in real-time, since it has infinite capacity in the Shannon sense. It provides instantaneous, error-free transmission of any input real number. Any arbitrarily large set of random quantities can be encoded into a single real number and perfectly reconstructed by using a suitable coding-decoding scheme. Hence with a single channel use, an arbitrarily large number of bits can be transmitted. However we need not worry because in real-time loops, one can do no better than to transmit the latest sample of the state vector. To make this channel model represent a bottleneck, one must limit how often the channel can be used over prescribed time intervals. This we do by charging a communication cost for transmissions. This channel model is suitable for loops with event-triggered sampling.

Noisy linear channel

This model is a generalization of the classical AWGN channel. Since we let the channel noise be coloured and non-Gaussian, we call this the AN channel. It accepts a real valued input and delivers the input added with noise. This model asks for an encoder that maps raw sensor measurements into a sequence of real valued channel inputs; such a map must be causal, but can be nonlinear and memory-based. To allow only finite-rate communications, we charge energy costs for inputs to the channel.

7.5.2 Modified Performance Cost

The performance cost now has same form as before, but the communication cost will take a positive functional form depending on the channel model:

$$J = \mathbb{E}\left[x_{T+1}^2 + p\sum_{i=1}^{T} x_i^2 + q\sum_{i=0}^{T} u_i^2\right] + J^{\text{Comm}} \tag{7.11}$$

To show these extensions for all the other channel models we study, we only need to find the appropriate versions of lemma 7.4. Once this is done, all the steps in the proofs for theorems 7.5, 7.7 can be repeated with no hitch. For each of the channel models we consider, an encoder that is controls forgetting from time 0 will be optimal in combination with the certainty equivalence control laws of 7.9. In what follows we will use the following symbols $\eta, \eta_t, \phi(\cdot), \varphi(\cdot)$ and attach different meanings to them, depending on the channel model.

7.5.3 Quantizer with its Rate Chosen Real-Time

We describe below the Design problem 2 for quantized control where the quantization rate is to be chosen real-time. We describe the situation where the rate has an expense attached, and when there may be both a common upper bound on the sizes of individual codewords and a separate upper bound on the average data rate over the entire horizon.

Communication cost

The channel is a discrete alphabet channel with a variable sized alphabet. With each channel use, the size of the alphabet must be chosen causally, and this choice is to be made by the encoder. At every time t, the channel input and output are the same: $\iota_t = z_t$. And

$$z_t = (\eta_t, \nu_t),$$

where the positive number η_t is the size of the alphabet, and the codeword $\nu_t \in \{1, 2, \ldots, \eta_t\}$. Let $\phi(\cdot)$ be a non-negative and increasing function of positive integers,

such that $\phi(0) = 0$. We use ϕ to measure data-rates. An example is: $\phi(\eta) = \log_2 \eta$. Let the positive integer $\overline{\eta}$ denote an upper limit on the alphabet size at any time. Then the communication cost incurred at time t can be described thus:

$$\varphi_t(\eta_t) = \begin{cases} \phi(\eta_i) & \text{if } \eta_i \leq \overline{\eta}, \\ +\infty & \text{if } \eta_i > \overline{\eta}. \end{cases}$$

Let the positive real number $\mathcal{R} \leq \overline{\eta}$ denote an upper limit on the average data rate over the entire horizon. We define the communication cost as follows:

$$J^{\text{Comm}} = \begin{cases} m \cdot \mathbb{E}\left[\sum_{i=0}^{T} \varphi(\eta_i)\right] & \text{if } \sum_{i=0}^{T} \varphi(\eta_i) \leq \mathcal{R} \times (T+1), \\ +\infty & \text{if } \sum_{i=0}^{T} \varphi(\eta_i) > \mathcal{R} \times (T+1), \end{cases}$$

where the non-negative real m is a Lagrange multiplier. It is easy to see that the signals x_t, $\{\iota_j\}_0^{t-1} \{z_j\}_0^{t-1}, \{\xi_j\}_0^{t-1}$ are sufficient statistics for encoding decisions, where of course $z_i = (\eta_t, \nu_t)$. All that is left to do now is to present a suitable version of lemma 7.4.

Lemma 7.8 (Variable rate CF encoder optimal for affine controls)**.** *Fix time $t = i$ and apply control laws affine from time i. Suppose that for all times $j > i$ the best encoding policies $\mathcal{E}_j^*(\cdot)$ (rules for variable alphabet sizes η_j as well as actual quantization maps) and their performances are independent of the partial control waveform $\{u_i, \ldots, u_T\}$. Then, for all times $j > i-1$ the best encoding policies $\mathcal{E}_j^*(\cdot)$ and their performances are independent of the slightly longer waveform $\{u_{i-1}, u_i, \ldots, u_T\}$.*

Sketch of proof. Consider the encoder choice at time $t = i$. For any fixed alphabet size η, let $\mathcal{E}^{\eta *}(\cdot)$ be the encoder possessing the two properties: (1) its alphabet size equals η, and (2) this encoder in combination with optimal policies for the later encoders $\left\{\mathcal{E}_j^*\right\}_{j=i+1}^{T}$ (meaning policies for variable alphabet sizes and quantization maps) achieves the lowest possible values for the performance costs. Where of course by performance cost of the encoder we mean those parts of the performance cost that, once affine control policies are fixed, depend on these encoders.

For every fixed η, we know that $\mathcal{E}^{\eta *}(\cdot)$ and the statistics of its outputs are independent of the policy for control u_{i-1}. Hence when this quantizer is used in combination with an optimal set of later encoders, the quantization distortion at time $t = i$, and the statistics of channel outputs at all times $j \geq i$ become independent of the control value u_{i-1}. Likewise the communication costs incurred at times $j \geq i$ become independent of the control value u_{i-1}. Since every admissible choice of η_t leads to this property, the lemma is proved. □

We present without proof the main theorem:

Theorem 7.9 (Optimality of separation and certainty equivalence)**.** *For Design problem 2, with the discrete alphabet channel of variable alphabet size, the performance cost (7.11) is minimized by applying the linear control laws*

$$u_t^* = -k_t^* \, \widehat{x}_{t|t}$$

in combination with the following encoder which is controls-forgetting from time 0*:*

$$\epsilon_t^*\left(\zeta_t\,;\{z_i\}_0^{t-1},\{\epsilon_i\,(\cdot)\}_0^{t-1}\right)=\underset{\epsilon(\cdot)}{\arg\inf}\;\Gamma_i\Big(\epsilon\,(\cdot)\;;\;\{z_i\}_0^{t-1},\{\epsilon_i\,(\cdot)\}_0^{t-1}\Big),$$

where, $k_i^*=a\frac{\beta_{i+1}}{q+\beta_{i+1}},\beta_i=p+\frac{a^2q\beta_{i+1}}{q+\beta_{i+1}},\beta_{T+1}=1,$ *and* $\lambda_i=\frac{a^2\beta_{i+1}^2}{q+\beta_{i+1}}$ *and where,*

$$\begin{aligned}\Gamma_t&=\lambda_t\;\mathbb{E}\left[\left(\zeta_t-\widehat{\zeta}_{t|t}\right)^2+m\cdot\varphi\,(\eta_t)\,\middle|\epsilon_t\,(\cdot)\,,\mathcal{D}_{(t-1)^+}^{con}\right]\\&\quad+\mathbb{E}\left[\Gamma_{t+1}^*\left(\overline{x}_0,\sigma_0^2,\{z_i\}_0^t,\{\epsilon_i\,(\cdot)\}_0^t\right)\right],\\\Gamma_T&=\mathbb{E}\left[\left(\zeta_T-\widehat{\zeta}_{T|T}\right)^2+m\cdot\varphi\,(\eta_T)\,\middle|\epsilon_T\,(\cdot)\,,\overline{x}_0,\sigma_0^2,\{z_i\}_0^{T-1},\{\epsilon_i\,(\cdot)\}_0^{T-1}\right],\\\Gamma_t^*&=\inf_{\epsilon(\cdot)}\;\Gamma_t\,(\epsilon)\,.\end{aligned}$$

Moreover, this control law is a certainty equivalence law.

7.5.4 Event-Triggered Sampling

We now summarize parallel developments for event-triggered messaging. Here the encoder transmits unquantized real numbers at selected event-triggered times. The best signal to sample and transmit is the state signal x_t.

Communication cost

The channel is an ideal, delay-free continuous valued one with no amplitude constraints. We will stipulate that the input to the channel is either a special silence symbol or a real number. In either case, the output will be a faithful reproduction of the input. Hence $\iota_t\equiv z_t$. Let η_i denote the random number of state samples transmitted up to and including time $t=i$. Then the encoder for event-triggered sampling can be represented by the following map from plant output to channel input

$$z_i=\begin{cases}x_i & \text{if } x_i\notin\mathcal{S}_i\\ \text{SILENCE} & \text{if } x_i\in\mathcal{S}_i,\end{cases}$$

where policies for the silence sets $\mathcal{S}_i$ have to be measurable w.r.t. the filtration generated by the data $\mathcal{D}_{(i-1)^+}^{\text{con}}$. Then we can write: $\eta_t=\sum_{i=0}^t\mathbb{1}_{\{x_t\notin\mathcal{S}_i\}}$. Let the non-negative number $N_0\leq T+1$ denote an initial budget of samples. This initial budget is a hard limit and the total number of samples taken over the entire horizon can never exceed N_0. Then we define the communication cost as follows:

$$J^{\text{Comm}}=\begin{cases}m\cdot\mathbb{E}\,[\eta_T] & \text{if } \eta_T\leq N_0,\\ +\infty & \text{if } \eta_T>N_0,\end{cases}$$

where the non-negative real m is a Lagrange multiplier. It is easy to see that the signals

$$x_t, \{z_j\}_0^{t-1}, \{\xi_j\}_0^{t-1}, \{\eta_j\}_0^{t-1}$$

are sufficient statistics for sampling decisions. Note also that the record of sample counts $\{\eta_j\}_0^{t-1}$ can be causally deduced from the record of channel outputs $\{z_j\}_0^{t-1}$.

If we set N_0 to be a finite number less than the horizon length $T+1$ and set the multiplier m to zero, then we get a design problem with a fixed budget N_0 and no cost attached to any number of samples within the budget. If instead we set the multiplier m to be some positive number and set the bound N_0 to be $T+1$, then we get a design problem with no budget constraint but with a communication cost growing linearly with the number of samples taken over the entire horizon. These two kinds of design problems and their hybrids will all be simultaneously studied by examining the general case where m can be any nonnegative number, and N_0 any positive number.

Lemma 7.10 (CF sampler optimal for affine controls)**.** *Fix time $t = i$ and apply control laws affine from time i. Suppose that for all times $j > i$ the best silence sets $\mathcal{S}_j^*(\cdot)$ and their performances are independent of the partial control waveform $\{u_i, \ldots, u_T\}$. Then, for all times $j > i-1$ the best silence sets $\mathcal{S}_j^*(\cdot)$ and their performances are independent of the slightly longer waveform $\{u_{i-1}, u_i, \ldots, u_T\}$.*

Sketch of proof. As with proving lemmas 3,4 we carry out two steps. First we show that because the cost to go is quadratic, the quantizer's objective at time i is to minimize a sum Γ_i of current and future estimation distortions. Second we show that the minimum of this sum distortion is independent of the control u_{i-1}. Thus the encoder becomes controls-forgetting from time $i-1$. □

The main theorem for event-triggered sampling is the same as Theorem 7.9 except for cosmetic changes to do with the nomenclature of the running communication cost.

7.5.5 Messaging over a Noisy Linear Channel

This channel accepts real valued inputs ι_t and delivers outputs z_t with noise added. For $0 \leq t \leq T$:

$$z_t = \iota_t + \chi_t,$$

where the channel noise process $\{\chi_i\}$ is IID with mean zero and variance $\sigma_\chi^2 < \infty$. At time t, the noise χ_t is independent of the state, controls and process noises upto and including time t. For this style of messaging, we describe a model that allows the encoder to choose the SNR for each message. Naturally the model will also specify costs incurred for choosing message SNRs.

Communication cost

Let the real-valued even function $\phi(\cdot)$ increase with increasing magnitude of argument, and let $\phi(0) = 0$. An example is the function $\phi(\iota) = \iota^2$. Let the positive real $\overline{\iota}$ denote a constant, hard upper limit on inputs to the channel. Then the communication cost incurred at a time t can be described thus:

$$\varphi_t = \begin{cases} \phi(\iota_t) & \text{if } |\iota_t| \leq \overline{\iota}, \\ +\infty & \text{if } |\iota_t| > \overline{\iota}. \end{cases}$$

Let the positive real $\mathcal{P} \leq \phi(\overline{\iota})$ denote an upper limit on the average power of channel inputs over the entire horizon. We define the communication cost from time t to the horizon end as follows:

$$J^{\text{Comm}} = \begin{cases} m \cdot \mathbb{E}\left[\sum_{j=t}^{T} \varphi\left(\iota_j\right)\right] & \text{if } \sum_{j=0}^{T} \varphi\left(\iota_j\right) \leq \mathcal{P} \times (T+1) \\ +\infty & \text{if } \sum_{j=0}^{T} \varphi\left(\iota_j\right) > \mathcal{P} \times (T+1)\,. \end{cases}$$

where the non-negative real m is a Lagrange multiplier.

Sufficient statistics and scope for the dual effect

It is straightforward to see that

$$x_t,\, \{\iota_j\}_0^{t-1}, \{\xi_j\}_0^{t-1}, \{z_j\}_0^{t-1}$$

are sufficient statistics at the encoder. As with quantized and event-triggered messaging, here too there is scope for the dual effect since the encoding map may be nonlinear.

Clearly there is no dual effect introduced if the upper limit on inputs is removed, and the encoder implements an affine encoder. But in general, there is scope for introducing the dual effect. If the encoder implements the quadratic encoder:

$$\xi_t^{quadratic} = \eta x_t^2,$$

then there is a second order dual effect. Another example of an admissible encoder that introduces the dual effect in the loop is one that implements the piecewise-constant encoder:

$$\xi_t = \begin{cases} -\overline{\iota} & \text{if } x_t \in (-\infty, -\theta)\,, \\ 0 & \text{if } x_t \in (-\theta, +\theta)\,, \\ \overline{\iota} & \text{if } x_t \in (+\theta, -\infty)\,, \end{cases}$$

where the threshold θ is fixed. In fact, this encoder has nearly the same input-output behaviour as the encoders considered in examples 7.1, 7.2. Using this parallel, one can setup an example of a loop with an AN channel such that the dual effect is present. And, when there is a finite, hard limit on amplitudes of channel inputs,

then the dual effect is present for any encoder other than the trivial ones of the form: $\xi_t \equiv constant\ \forall t$.

As in the case of other types of messaging, we can show that even though the dual effect is present, the dynamic problem has a separated solution and certainty equivalence controls are optimal. In general, it is still hard to design optimal encoders, especially when there is a finite hard limit on channel input amplitudes. With no such hard limits, and with Gaussian plant and channel noises, optimal encoders are linear (Sauer and Melsa, 1974; Bansal and Başar, 1989; Breun and Utschick, 2008; Freudenberg et al., 2011). We now outline how one can prove optimality of separated encoder design and certainty equivalence controls for the dynamic problem.

Lemma 7.11 (CF encoder optimal for affine controls)**.** *Fix time $t = i$ and apply control laws affine from time i. Suppose that for all times $j > i$ the best encoding policies $\mathcal{E}_j^*(\cdot)$ and their performances are independent of the partial control waveform $\{u_i, \ldots, u_T\}$. Then, for all times $j > i-1$ the best encoding policies $\mathcal{E}_j^*(\cdot)$ and their performances are independent of the slightly longer waveform $\{u_{i-1}, u_i, \ldots, u_T\}$.*

Sketch of proof. As with proving lemmas 3,4 we carry out two steps. First we show that because the cost to go is quadratic, the quantizer's objective at time i is to minimize a sum Γ_i of current and future estimation distortions. Second we show that the minimum of this sum distortion is independent of the control u_{i-1}. Thus the encoder becomes controls-forgetting from time $i-1$. □

The main theorem for communication over a noisy linear channel is the same as Theorem 7.9 expect for the necessary changes in nomenclature of communication cost.

We might also add that for all of he above channel models, the results for Design problem 2 can also be extended to the case of vector valued states with only partial, noisy linear observations available at the sensor (encoder). Such a situation is no more complicated than that one where the encoder observes the state perfectly. In the partially observed case, the role of the 'state' falls on the estimate produced by the encoder's Kalman filter.

7.6 Constrained Encoder-Controller Design

We now use our understanding of the dynamic encoder-controller design problem (Design problem 2) to examine the constrained encoder-controller design problem (Design problem 3) and the hold-waveform-controller and encoder design problem (Design problem 4). In this section, we show that, in general, separation in design of encoder and controller is not optimal for these design problems. We do this by presenting a counterexample for each of these design problems. Some of these counterexamples illustrate that the distortion term in the cost-to-go lacks symmetry w.r.t. translations (7.8). Recall that this property was instrumental in

ensuring separation in the dynamic encoder-controller design problem (see proof of Lemma 7.4).

Thus, we begin with Example 7.3, which illustrates, through explicit calculations, that symmetry w.r.t. translations does indeed occur in the dynamic encoder-controller design problem. Next, we impose a set of constraints on the decision makers of the closed-loop system in Examples 7.4-7.6, which have the effect of removing the symmetry w.r.t. translations. For these cases, we show that separation in design is no longer optimal. In Example 7.8, we illustrate that separation is not optimal when the control signals are held constant over random epochs.

7.6.1 Symmetry w.r.t. Translations Leads to Separation

We present a simple example of a dynamic encoder-controller design problem; the encoder is specified in a parametric form, but the choice of the parameters can be dynamic, with no restrictions on the set of parameters. We show that the optimal controller uses the certainty equivalence law.

Example 7.3

For the linear plant (7.1), with initial state x_0 given by a zero mean Gaussian with variance σ_x^2, and process noise w_k given by a zero mean Gaussian with finite variance σ_w^2, let the horizon length be $T = 2$. Let the cost coefficients p and q remain unspecified. Let the channel alphabet be $\{1, 2\}$. The controller receives a quantized version of the state, denoted z_k and given by

$$z_k = \begin{cases} 1 & \text{if } x_k \leq \delta_k \ , \\ 2 & \text{otherwise} \ . \end{cases}$$

The quantizer thresholds δ_0 and δ_1 are to be chosen along with the control signals u_0 and u_1, to jointly minimize the two-step horizon control cost.

We use dynamic programming to find the optimal values for u_1, δ_1 and u_0, and δ_0, in the specified order. From Lemma 7.3, we know that u_1^* is given by the certainty equivalence law as $-\frac{a}{q+1}\hat{x}_{1|1}$, where the MMSE estimate of x_1 is given by $\hat{x}_{1|1} = \mathbb{E}\left[x_1 \middle| \{z_i\}_0^1\right]$.

Then, let us consider the cost-to-go at the previous time step,

$$V_0 = \min_{u_0, \delta_1} \mathbb{E}\left[a^2(p + a^2)x_0^2 + (q + p + a^2)u_0^2 + 2a(p + a^2)x_0 u_0 - \frac{a^2}{q+1}\hat{x}_{1|1}^2 \ \middle| \ z_0\right] + \kappa \ , \tag{7.12}$$

where $\kappa = (1 + p + a^2)\sigma_w^2$. The above cost-to-go is to be minimized by selecting a suitable u_0 and δ_1 simultaneously. To do this, we first need to find an expression for $\mathbb{E}\left[\hat{x}_{1|1}^2 \middle| z_0\right]$. The encoder outputs at times $0, 1$ tell us the quantization cells in which x_0 and x_1 lie. We use this information to find an expression for the estimate

$\hat{x}_{1|1}$, as shown in Appendix D, and rewrite the cost-to-go as

$$V_0 = \min_{u_0,\delta_1} \mathbb{E}\left[a^2(p+a^2)x_0^2 + \overbrace{(q+p+a^2\frac{q}{q+1})u_0^2 + 2a(p+a^2\frac{q}{q+1})x_0u_0}^{\text{function of } u_0} \Big| z_0 \right] - \underbrace{\frac{a^2}{q+1}\frac{\sum_{j=1}^{N}\vartheta^2\left(\frac{\delta_{j-1}-u_0}{\sigma_2},\frac{\delta_j-u_0}{\sigma_2}\right)}{\mathbb{P}\left(x_0\in(\theta_{l-1},\theta_l)\right)}}_{\triangleq\Gamma_1:\quad \text{function of } u_0 \text{ and } \mathcal{E}_1} + (1+p+a^2)\sigma_w^2 \,, \tag{7.13}$$

where $\sigma_2^2 = \sigma_w^2 + a^2\sigma_x^2$. The term $\vartheta(\underline{r},\bar{r})$ in the above equation is given by

$$\vartheta(\underline{r},\bar{r}) = \Bigg[-a\sigma_x\phi\left(\frac{\theta_l}{\sigma_x}\right)\Phi\left(r\frac{\sigma_2}{\sigma_w} - \theta_l\frac{a}{\sigma_w}\right) - \sigma_2\phi(r)\Phi\left(\frac{\theta_l}{\sigma_1} - r\frac{a\sigma_x}{\sigma_w}\right) + a\sigma_x\phi\left(\frac{\theta_{l-1}}{\sigma_x}\right)\Phi\left(r\frac{\sigma_2}{\sigma_w} - \theta_{l-1}\frac{a}{\sigma_w}\right) + \sigma_2\phi(r)\Phi\left(\frac{\theta_{l-1}}{\sigma_1} - r\frac{a\sigma_x}{\sigma_w}\right)\Bigg]_{r=\underline{r}}^{\bar{r}}, \tag{7.14}$$

where $\sigma_1^2 = \sigma_x^2\sigma_w^2/\sigma_2^2$ and $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

The quantization distortion term Γ_1 in (7.13) possesses symmetry w.r.t. translations, as defined in (7.8). Thus, for any value of the control signal u_0, the minimum value is given by $\Gamma_1^*(\mathcal{E}_1)$, a term that depends only on the encoder. Then, the cost-to-go with respect to the control signal u_0 comprises of only the terms in the first row in (7.13). Hence, we obtain separation. Furthermore, the optimal control signal is given by the certainty equivalence law, $u_0^{\text{CE}} = -\frac{a(p+a^2\frac{q}{q+1})}{p+q+a^2\frac{q}{q+1}}\hat{x}_{0|0}$. Thus, the certainty equivalence property holds for this setup.

We illustrate symmetry w.r.t. translations in Figure 7.5. For the choice of parameters $a = 1$, $p = 1$ and $q = 1$, we evaluate the quantization distortion term Γ_1 from the above example and show that the minimum that this function attains over the range of the quantizer threshold δ_1 is invariant for different values of u_0. To evaluate the cost-to-go, we make an arbitrary choice for the quantizer threshold at time $k = 0$ as $\delta_0 = 0$, and compute the estimates and probabilities using this choice.

7.6.2 Optimal Constrained Encoder

We now impose a restriction on the choice of encoder parameters. The one-bit quantizer that we consider in the previous example selects two semi-infinite intervals as the quantizer cells, $\Delta_1 = (-\infty, \delta_k]$ and $\Delta_2 = (\delta_k, \infty)$. We restrict the choice of the quantizer threshold to a constraint set, such that $\delta_k \in \Theta$. In the following example, we see that separation is lost for this constrained optimization problem.

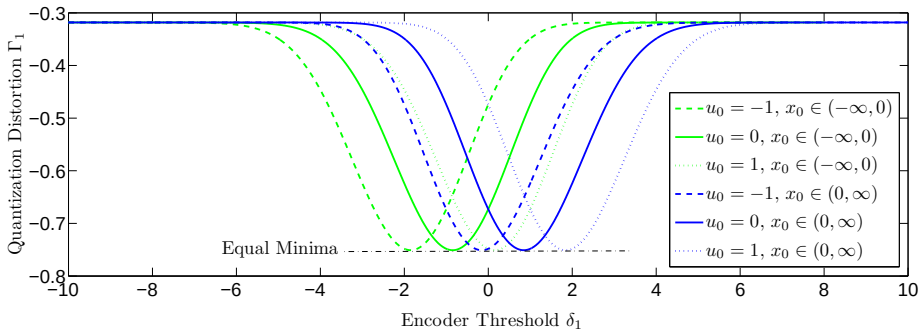


Figure 7.5: This plot illustrates the *symmetry w.r.t. translations* of the quantization distortion term Γ_1 in (7.13). Different values of u_0 result in the same minimum value for Γ_1 at different values of δ_1, thus resulting in separation and certainty equivalence in Example 7.3.

Example 7.4

Consider the same setup as in Example 7.3, with the restriction that the quantizer threshold be chosen from the set $\Theta = (-1, 1)$. The quantizer thresholds $\delta_0 \in \Theta$ and $\delta_1 \in \Theta$ are to be chosen along with the control signals u_0 and u_1, to jointly minimize the two-step horizon control cost.

We follow the same procedure as before. The optimal control signal u_1 is given by the certainty equivalence law as $u_1^* = u_1^{\text{CE}}$. This gives us the same cost-to-go V_0 from (7.12). Evaluating Γ_1 for the parameters $a = 1$, $p = 1$ and $q = 1$, we plot it over a range of quantizer thresholds $\delta_1 \in \Theta$, for three arbitrary choices of u_0, in Figure 7.6. By restricting the range of quantizer thresholds to Θ, we do not permit all the curves to reach their minima from Figure 7.5. In particular, the minima for $u_0 = -1$, when $x_0 \in (-\infty, 0)$, and $u_0 = 1$, when $x_0 \in (0, \infty)$ are higher than before. Thus, the minimum value of Γ_1 obtained over the range of δ_1 now varies depending on the choice of u_0. Consequently, there is no longer a symmetry w.r.t. translations, and separation cannot be achieved using the proof of Theorem 7.5. Furthermore, the optimal control signal u_0^* must be chosen along with δ_1^* to optimize the entire cost-to-go including the term Γ_1. Thus, u_0^* does not just minimize a quadratic expression in this problem, and cannot be chosen independently of the encoding policy. Hence, separation in design of the controller and encoder is no longer optimal.

7.6.3 Optimal Constrained Controller

We now remove the restriction on the encoder parameters, and instead impose the following restriction on the controller: the controls are required to have limited

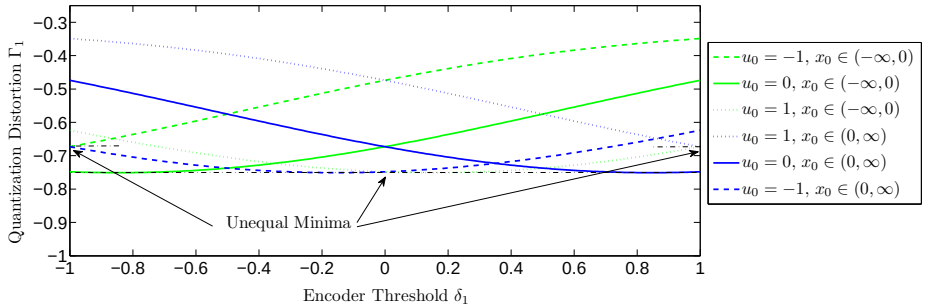


Figure 7.6: This plot illustrates the lack of *symmetry w.r.t. translations* of Γ_1, when the quantizer thresholds are restricted to be chosen from an interval, such as in Example 7.4. Different values of u_0 do not result in the same minimum value for Γ_1 over the range of δ_1, thus resulting in a lack of separation and certainty equivalence.

range. Specifically, the control values at ever time step must come from a specified constraint set $\mathcal{U}$. We present two versions of this constraint: in case 1, our constrained control set $\mathcal{U}$ is discrete, and in case 2, the constrained control set is an interval $\mathcal{U} = (u_{\min}, u_{max})$.

Example 7.5

Consider the same setup as in Example 7.3, with the restriction that the control signal be chosen from a discrete set $\mathcal{U} = \{-1, 0, 1\}$. The quantizer thresholds δ_0 and δ_1 are to be chosen along with the control signals $u_0 \in \mathcal{U}$ and $u_1 \in \mathcal{U}$, to jointly minimize the two-step horizon control cost.

The unconstrained minimizer for the cost-to-go at the terminal time is given by the certainty equivalent value u_1^{CE}. The best we can do, given the constraint set $\mathcal{U}$, is to choose the control value from the discrete set $\mathcal{U}$ that results in the lowest cost-to-go. Using this principle, we find the optimal control signal u_1^* to be

$$u_1^* = \begin{cases} -1 & \hat{x}_{1|1} \geq \frac{q+1}{2a}\ , \\ 0 & \frac{q+1}{2a} \geq \hat{x}_{1|1} \geq -\frac{q+1}{2a}\ , \\ 1 & \hat{x}_{1|1} \leq -\frac{q+1}{2a}\ . \end{cases}$$

The optimality regions are identified by comparing $\min_{u_1 \in \mathcal{U}} V_1(u_1)$ evaluated at each permissible value of u_1, and determining the switching points.

The cost-to-go V_0, obtained by averaging over the three different cost-to-go

functions obtained at time $k = 1$, is given by

$$V_0 = \min_{u_0,\delta_1} \mathbb{E}\Big[a^2(p+a^2)x_0^2 + (q+p+a^2)u_0^2 + 2a(p+a^2)x_0u_0 \\ + (-2a\hat{x}_{1|1} + q + 1)\mathbb{1}_{\left\{\hat{x}_{1|1} \geq \frac{q+1}{2a}\right\}} \\ + (2a\hat{x}_{1|1} + q + 1)\mathbb{1}_{\left\{\hat{x}_{1|1} \leq -\frac{q+1}{2a}\right\}} \Big| z_0\Big] + (1+p+a^2)\sigma_w^2 \, .$$

We denote the terms in the above cost-to-go that directly depend on the choice of the encoder threshold δ_1 as Γ_1^{RC}. Using the expression for $\hat{x}_{1|1}$ and the posterior density for x_1 from Appendix D, we compute Γ_1^{RC} as

$$\Gamma_1^{\mathrm{RC}} = \mathbb{E}\Big[(-2a\hat{x}_{1|1} + q + 1)\mathbb{1}_{\left\{\hat{x}_{1|1} \geq \frac{q+1}{2a}\right\}} + (2a\hat{x}_{1|1} + q + 1)\mathbb{1}_{\left\{\hat{x}_{1|1} \leq -\frac{q+1}{2a}\right\}} \Big| z_0\Big] \\ = \sum_{j=1}^{N} \frac{\mathbb{P}\left(x_0 \in (\theta_{l-1}, \theta_l), x_1 \in (\delta_{j-1}, \delta_j)\right)}{\mathbb{P}\left(x_0 \in (\theta_{l-1}, \theta_l)\right)} \Big((-2a\hat{x}_{1|1} + q + 1)\mathbb{1}_{\left\{\hat{x}_{1|1} \geq \frac{q+1}{2a}\right\}} \\ + (2a\hat{x}_{1|1} + q + 1)\mathbb{1}_{\left\{\hat{x}_{1|1} \leq -\frac{q+1}{2a}\right\}} \Big) \, .$$

Evaluating the above expression for parameters $a = 1$, $p = 1$ and $q = 1$, and some arbitrary choice of quantizer threshold δ_0, we plot Γ_1^{RC} over a range of quantizer thresholds δ_1, for different choices of u_0 from the set $\mathcal{U}$, in Figure 7.7. Notice that the minimum values of Γ_1^{RC} obtained over the range of δ_1 vary depending on the choice of u_0. In other words, there is no symmetry w.r.t. translations. Consequently, a separation in design of the controller and encoder is no longer optimal.

We now present a slight variation in the restriction on the controller, and reconfirm that separation in design of controller and encoder is not optimal.

Example 7.6

Consider the same setup as in Example 7.3, with the restriction that the control signal be chosen from an interval $\mathcal{U} = (u_{\min}, u_{\max})$. The quantizer thresholds δ_0 and δ_1 are to be chosen along with the control signals $u_0 \in \mathcal{U}$ and $u_1 \in \mathcal{U}$, to jointly minimize the two-step horizon control cost.

As in the solution to the previous example, note that the unconstrained minimizer for the cost-to-go V_1 is the certainty equivalent value u_1^{CE}. The best we can do, given the constraint set $\mathcal{U}$, is to choose the control signal closest to the unconstrained value. This follows from the convexity of the quadratic cost-to-go. Using this principle, we find the optimal control signal u_1^* to be

$$u_1^* = \begin{cases} u_{\min} & u_1^{\mathrm{CE}} \leq u_{\min} \, , \\ u_1^{\mathrm{CE}} & u_{\min} \leq u_1^{\mathrm{CE}} \leq u_{\max} \, , \\ u_{\max} & u_1^{\mathrm{CE}} \geq u_{\max} \, . \end{cases}$$

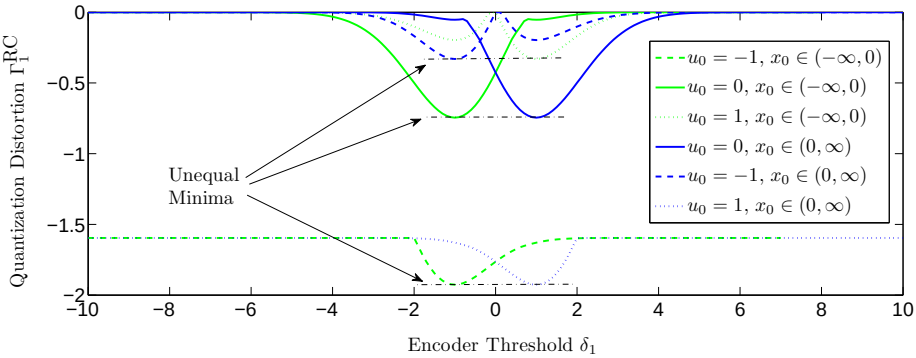


Figure 7.7: This plot illustrates the lack of *symmetry w.r.t. translations* for Γ_1^{RC}, when the controls are restricted to be chosen from a discrete set $\mathcal{U}$, such as in Example 7.5. Different values of u_0 do not result in the same minimum value for Γ_1^{RC} over the range of δ_1, thus resulting in the lack of separation and certainty equivalence.

Evaluating the cost-to-go V_1 using u_1^*, and reusing quantities derived in Appendix D, we can write up the cost-to-go V_0 as before. More interesting to us are the terms in this expression that directly depend on the choice of the quantizer threshold δ_1, as given by

$$\begin{aligned}\Gamma_1^{\mathrm{IC}} =& \mathbb{E}\Bigg[(2a\hat{x}_{1|1}u_{\min} + (q+1)u_{\min}^2)\mathbb{1}_{\left\{\hat{x}_{1|1} \geq -\frac{q+1}{a}u_{\min}\right\}} \\ &- \frac{a^2}{q+1}\hat{x}_{1|1}^2\mathbb{1}_{\left\{-\frac{q+1}{a}u_{\max} \leq \hat{x}_{1|1} \leq -\frac{q+1}{a}u_{\min}\right\}} \\ &+ (2a\hat{x}_{1|1}u_{\max} + (q+1)u_{\max}^2)\mathbb{1}_{\left\{\hat{x}_{1|1} \leq -\frac{q+1}{a}u_{\max}\right\}}\Bigg| z_0\Bigg] .\end{aligned}$$

Evaluating this expression for parameters $a = 1$, $p = 1$, $q = 1$, $u_{\min} = -2$ and $u_{\max} = 2$, and some arbitrary choice of quantizer threshold δ_0, we plot Γ_1^{IC} over a range of quantizer thresholds δ_1, for different choices of u_0 from the set $\mathcal{U}$, in Figure 7.8. Notice that the minimum value of Γ_1^{IC} obtained over the range of δ_1 varies depending on the choice of u_0. Thus, there is no symmetry w.r.t. translations, and a separation in design is no longer optimal.

In both the above examples, the constrained set $\mathcal{U}$ did not contain the certainty equivalent values of the control signal u_1 for at least some values of δ_1. The resulting cost-to-go V_0 was altered, such that the symmetry w.r.t. translations was lost. Consequently, separation no longer holds. The restriction removed the certainty equivalence property during time step $k = 1$, but the resulting cost and the information pattern resulted in the lack of separation itself at time step $k = 0$. A similar

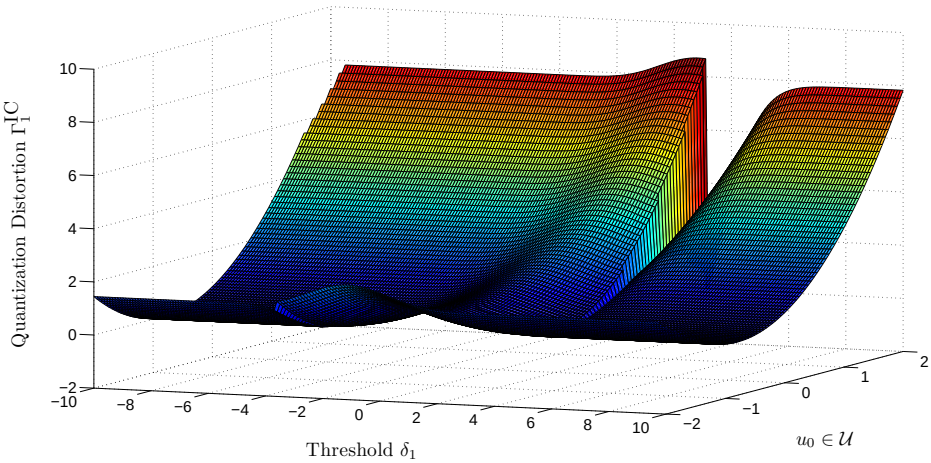


Figure 7.8: This plot illustrates the lack of *symmetry w.r.t. translations* of Γ_1^{IC}, when the controls are restricted to be chosen from an interval, such as in Example 7.6. Different values of u_0 do not result in the same minimum value for Γ_1^{IC} over the range of δ_1, thus resulting in lack of separation and certainty equivalence.

problem setup has been explored by Bernhardsson (1989), where the control gain is restricted to be chosen from two given values. The dual effect has been shown for this problem setup as well.

7.6.4 ZOH and Event-Triggered Sampling

We study numerically two cases of control under event-triggered sampling. Basically these are problems with a sampling budget of exactly one. For the controller, we must design a whole waveform to be applied up to the time when the first sample is received. We are already given the control law to be applied from this random sampling time to the end time. For the encoder, we must design an envelope to generate exactly one sample between time $t = 1$ and $t = T$.

We study two examples, and in both of them, the encoder is allowed to be dynamic. In the first example, the control waveform up to the first sample time is pre-assigned, and it has a particular linear dependence on the Kalman predictor. In the second example, the control waveform upto the first sample time must be a zero order hold waveform.

Example 7.7

[Fixed linear control law upto an event-triggered sample] For the scalar linear plant (7.1), let the coefficient $a = 1$, and let the initial state $x_0 = 2$, and $\sigma_0 = 0$, and let this information be known to the encoder and the controller. This simply

means that $z_0 = x_0$. This information is prestored at the controller. Let the variance $\sigma_w^2 = 0.5^2$. Let the horizon end $T = 4$, and let $p = 1, q = 0.2$. The control law is fixed to be:

$$u_t = \begin{cases} k_t^* \, \mathbb{E}\left[x_t \,|x_0, \{u_i\}_0^{t-1}\right], & \text{for } 0 \le t \le \tau - 1, \\ k_t^* \, \mathbb{E}\left[x_t \,|x_\tau, \{u_i\}_\tau^{t-1}\right], & \text{for } \tau \le t \le T, \end{cases}$$

where the gains k_t^* are the ones from the CE law (7.9), and τ satisfies $1 \le \tau \le T$ and is the first and only sample time, which is chosen by encoder. Choose a policy (sampling envelope) which comprises silence sets $\{\mathcal{S}_1, \ldots, \mathcal{S}_T\}$ giving:

$$\tau = \min\Big\{T, \min_{t \ge 1}\{t : x_t \notin \mathcal{S}_t\}.\Big\}$$

Next we consider an example of a design problem with a ZOH control. Here we specialize to the case where the control's hold epochs are forced to be exactly the inter-sample intervals.

Example 7.8

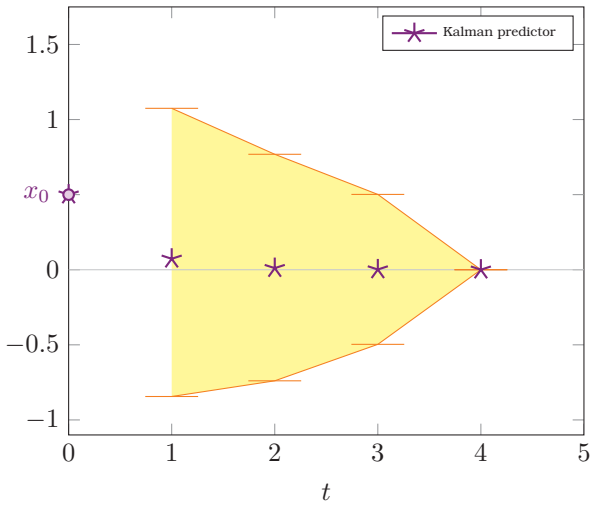
[ZOH control upto an ET sample] For the scalar linear plant (7.1), let the coefficient $a = 1$, and let the initial state $x_0 = 2$, and $\sigma_0 = 0$, and let this information be known to the encoder and the controller. This simply means that $z_0 = x_0$. This information is prestored at the controller. Let the variance $\sigma_w^2 = 0.5^2$. Let the horizon end $T = 4$, and let the coefficients $p = 1, q = 0.2$. There are exactly two epochs; and they must be precisely $\{0, 1, \ldots \tau - 1\}$ and $\{\tau, \ldots, T\}$, where τ is the first and only sample time, and is chosen to occur at or later than time $t = 1$. The control laws over the second epoch are fixed to have the form: $u_t = k_i^* \, \mathbb{E}\left[x_\tau \,|x_{\tau_0}\right]$, for $\tau \le t \le T$, where the gains k_t^* are the ones from the CE law (eqn 7.9). Pick: (1) a control law for the first epoch having the zero-order hold form:

$$u_t = \mathcal{K}^0\left(x_0\right), \text{ for } 0 \le t \le \tau - 1,$$

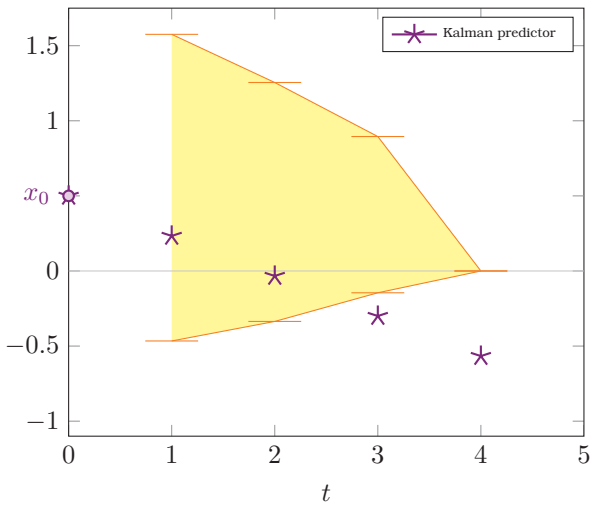
and (2) a sampling envelope which comprises silence sets $\{\mathcal{S}_1, \ldots, \mathcal{S}_T\}$ for generating the sample time:

$$\tau = \min\Big\{T, \min_{t \ge 1}\{t : x_t \notin \mathcal{S}_t\}.\Big\}$$

The optimal sampling envelope of the ZOH control example (Figure 7.8) shown in Figure 7.9(b). This is clear pictorial evidence that the dual effect is present in the loop. This becomes clear from the reasoning below. Supposing the dual effect were absent, then the encoder's goal would have been to pick the sample time τ to minimize a weighted sum of squared estimation errors upto time $\tau - 1$. The



(a) Optimal sampling envelope for Example 7.7



(b) Optimal sampling envelope for Example 7.8

Figure 7.9: Event-triggered sampling with exactly one sample after time $t = 0$

envelope optimal for that objective will be a sequence of silence set symmetric about the means $\mathbb{E}\left[x_t \left| x_0, \{u_i\}_0^{\tau-1}\right.\right]$. When the plant noise is Gaussian, Hajek and others (Hajek, 2002; Hajek et al., 2008; Lipsa and Martins, 2011; Nayyar et al., 2013) predict a symmetric sequence of silence is optimal. They also imply that a sequence of silence sets that are not symmetric about the respective means $\mathbb{E}\left[x_t \left| x_0, \{u_i\}_0^{\tau-1}\right.\right]$ will lead to suboptimal state estimation. Since the optimal envelope computed numerically is clearly non-symmetric about the means $\mathbb{E}\left[x_t \left| x_0, \{u_i\}_0^{\tau-1}\right.\right]$, there must be a dual effect in the loop, which is exploited by this optimal pair of sampler and ZOH controller.

7.7 Discussion

An interesting aspect of our results is that we have shown that separation and certainty equivalence are optimal for Design problem 2, despite the dual effect being present in the NCS of Section 7.2. To understand this result, we examine two implementations of the optimal encoder-controller pair for this design problem, and using these, we draw out an some subtle points concerning dual effect and optimality of separation and certainty equivalence.

Bar-Shalom and Tse (1974) consider the loop shown in Figure 7.2, for a linear plant with $\Phi_t(\cdot) = ax_t + u_t + w_t$. At the sensor, instead of our dynamic encoder Ψ_t, they place a nonlinear map. This sensor map is time-varying but memoryless and its exact functional form is given. For this setup, they have a result stating the mutual exclusivity of the dual effect and optimality of certainty equivalence controls. In their setting, if the linear 'plant' is such that the effect of controls is never felt at the observation signal y_t, then clearly there is no dual effect. This happens in the case where the so-called 'plant' has a sub-system that produces the 'plant' output after explicitly removing the effect of controls.

However, for our setup (Figure 7.1), the sensor has a dynamic encoder even after one performs the equivalence transformation by subtracting out the effect of controls. The use of 'innovation coding' leads to a closed loop shown in Figure 7.4. The crucial difference from the setup of Bar-Shalom and Tse is that rather than being a memoryless nonlinear map, the encoder $\widetilde{\xi}_t$ is a dynamical system. Hence their result does not apply. But it springs the following question: Does the plant sensor combination in the closed loop 7.4 have a dual effect if an encoder is used that is optimal for the dynamic design problem ? To answer this question, one needs to interpret carefully what it means to implement an optimal encoder. For different interpretations, one gets different answers.

Assume that we are implementing the feedback loop Figure 7.1 with the optimal encoder and any admissible controller. One possible implementation is the following: The encoder stores the actual set of control policies used by the controller, and uses this to carry out the innovation encoding, and on the result applies the sequential quantizer $\xi_t^*\Big(\bullet\,;\, \{z_i\}_0^t, \{\xi_i(\cdot)\}_0^{t-1}, \{u_i = 0\}_0^{t-1}\Big)$. Clearly, because of exact cancellation of controls, there is no dual effect in this loop and the answer to our

question is: No.

The second implementation is the following: The encoder assumes that the controller is applying the certainty equivalence laws (7.9). It subtracts out the effect of the these certainty equivalence control laws. To the residue $\widetilde{\zeta}_t$, it applies the sequential quantizer $\xi_t^*\left(\bullet;\{z_i\}_0^t,\{\xi_i(\cdot)\}_0^{t-1},\{u_i=0\}_0^{t-1}\right)$. This encoder is not controls-forgetting. But yet when used in combination with the certainty equivalence laws of (7.9), it leads to minimum performance cost.

On the other hand, when this encoder is used in combination with a general admissible control law, there is potential mismatch between the encoder's assumption and the actual controller behaviour. The effect of the controls is not absent in the input to the sequential quantizer $\xi_t^*\left(\bullet;\{z_i\}_0^t,\{\xi_i(\cdot)\}_0^{t-1},\{u_i=0\}_0^{t-1}\right)$. Clearly, there is a dual effect. And for this interpretation, the common answer to our question is: Yes.

This leads to an interesting consequence. If a pair of encoding and control strategies is optimal, then the individual strategies that are components of the pair must be person-by-person optimal. Since the combination of certainty equivalence controls and the corresponding optimal encoder is optimal, it follows that the certainty equivalence controls must be optimal for the single-agent control problem obtained by fixing the encoder to be the optimal one. Since the second interpretation of implementing the optimal encoder is perfectly valid, it turns out that certainty equivalence controls can be optimal even though the dual effect is present in the loop. Thus we can conclude that the theorem of Bar-Shalom and Tse cannot generalize to the scenario where sensors implement dynamic encoders.

7.8 Summary

In this chapter, we delve into the structural properties of stochastic systems with nonlinear measurements such as those found in NCSs. We have seen through examples that the dual effect is present in the plant-encoder-channel combination. Hence in general, it is suboptimal to apply a controls-free encoder, or to apply an affine controller. It has long been known that for the design problem with a static encoder, separation is not optimal, and that the optimal control laws are nonlinear (Curry, 1970). Recent interest in the dynamic design problem was due to Borkar and Mitter (1997) who describe advantages obtained by applying controls-forgetting encoders. Many papers state that the separated design is optimal for the dynamic design problem for the various channel models we have treated. We have shown by dynamic programming that these statements are indeed correct. This is an instance of the optimal decision policies 'ignoring' the presence of the dual effect. But a separated design need not be optimal for other design problems. In particular, for event-triggered sampling the dynamic design problem has a separated design, but the ZOH control design problem does not have a separated solution. This is at least partly surprising because, separated design is optimal for the classical LQG partially observed control with or without the ZOH control restriction.

In this thesis, we have dedicated two chapters to structural analyses of linear stochastic systems with nonlinear measurement policies. In Chapter 3, we examined structural properties of a control system for a given measurement policy. We found that there is a dual effect for event-triggering policies in general, and this implied that separation and certainty equivalence are not necessarily optimal for these control systems. In this chapter, we examined structural properties of a control system where both the encoder and controller had to be simultaneously chosen. We found a dual effect for three different channel models here, including the channel with event-triggered sampling. However, we found that the consequences of a dual effect for dynamic measurement policies need not be the same as for static measurement policies. Thus, we were able to establish separation and certainty equivalence for at least one design problem even in the presence of a dual effect.

Chapter 8

Conclusions and Future Work

This thesis examined the concept of a state-based channel access policy for networked control systems (NCSs). We modelled a multiple access network on the sensing link of the control system, and designed an access mechanism to guarantee stability for each control system as well as the network. Along with this, we studied structural properties of control systems with dynamic, nonlinear measurement policies, such as those found in NCSs. In this final chapter, we present a short summary of the contributions of this thesis, and directions for future work.

8.1 Conclusions

In this thesis, we showed that state-based channel access policies enable network access to be adapted to the needs of the plant, while also enabling the contribution from each plant to be adapted to the total traffic in the network. We examined two realizations of state-based policies, using event-triggering and prioritization policies, while ensuring a distributed implementation for each to facilitate easy deployment on wireless sensor nodes. Our main contributions were the following:

Innovations-based Policies: In Chapter 3, we found that there is a dual effect when the scheduler uses the state of the plant to initiate a transmission in the network. To simplify the control design, we suggested the use of static functions of the innovations process in the scheduler. This resulted in certainty equivalence for the optimal controller. This scheduling policy, along with Bianchi's assumption, also resulted in a Markov model for the interference from other users in the network in Chapter 4.

Dual Predictor Architecture: In Chapter 3, we used the results of our structural analysis to identify a dual predictor architecture that resulted in separation in design of the scheduler, observer and controller. This architecture specified a symmetric scheduling criterion based on the innovations, along with a simple observer and a certainty equivalent controller. We used this architecture for both the

event-based formulation in Chapters 4 and 5, and the prioritization framework in Chapter 6.

Interference Model for Event-based Systems: In Chapter 4, we analyzed the interactions in a network of control systems that each implement the dual predictor architecture along with a contention resolution mechanism (CRM). Using Bianchi's assumption, we constructed a Markov model for the steady state network interactions and validated this assumption for various network configurations.

Analysis and Design of Event-based Systems: Once we identify an interference model, each event-based system in a network can be analyzed in isolation. Using this technique, in Chapter 4 we analyzed the network reliability and delay for an event-based system, and in Chapter 5 we analyzed the stability of an event-based system. We also used the results of our stability analysis to synthesize an event-triggering policy that guarantees stability for the plant and the network.

Analysis and Design of Attention-based Tournaments: In Chapter 6, we designed and implemented a distributed state-based prioritization policy for a network of control systems. We presented a performance analysis for this access mechanism and characterized the estimation and control performance of a system in this network with upper bounds. Our results showed that attention-based tournaments can result in better estimation and control costs than any agnostic random access protocol with the same average channel access probability.

Joint Optimization of Measurement and Control: In Chapter 7, we found that it is optimal to apply separation to the optimal dynamic state-based measurement and control policies for the networked LQG problem, despite the presence of a dual effect in the NCS. We found that the dual role of the controller due to the probing incentive can be effected by both the control and measurement policies, under some circumstances, leading to separation and certainty equivalence. On the other hand, we identified a number of constrained design problems where separation in design was not optimal. In this work, we also highlighted some subtleties that arise when dynamic policies are encountered in stochastic control systems.

A general conclusion from this thesis is that state-based channel access policies can outperform agnostic policies. To enable the use of wireless control in future applications, such solutions must be explored to obtain performance guarantees from control systems using resource-constrained networks. However, state-based channel access policies may not be as useful in heterogenous networks where the models of some network users are unknown to the designer. Existing multiple access methods are better suited for such networks.

8.2 Future Work

There are several directions to further develop the work presented in this thesis. In general, the design methodology presented in this thesis must be applied to real

experiments and evaluated in industrial settings. We have presented an initial attempt in this direction in Chapter 6. However, our experiments are limited in scope and scale. Furthermore, our design methodology may prove to be quite intensive when applied to large-scale networks. A small change in any part of the network would require a complete overhaul in design. A universal channel access design may counter this drawback. A few specific problems for future work are discussed below.

Multiple Access on Other Links: This thesis explored the design of a channel access policy for the sensing link. A similar study can be performed for the actuation link as well. There are some dualities in the problem setup which can be exploited, but there are differences as well. If we consider multiple access on the actuation link alone, with dedicated connections on the sensing link, much of the underlying formulation will remain the same. This is due to the generic nature of the causal policies that we seek in both formulations. However, the policies we obtain must be re-imagined and re-interpreted as control, channel access or observer policies. The resulting structural properties can be quite different based on the interpretation we draw. A more interesting and challenging problem arises when there is a multiple access channel on both the sensing and actuation links. In this case, the controller must expand its role to act as a relay, which attempts to communicate the information it receives on the sensing link to a smart actuator. This is related to the signalling incentive that arises from the non-classical information pattern in this problem, as illustrated by Witsenhausen (1968) in his famous counterexample. Despite its inherent difficulties, this problem must be tackled to present a complete wireless networked control solution.

In this thesis, we assumed that ACKs are always available and never lost. However, this is not realistic. A challenging problem arises if there are losses in the acknowledgement channel, because the channel access policy can no longer keep track of the information available at the controller. Now, even a channel access policy chosen as a static function of the innovations may no longer result in separation or certainty equivalence. This is because the controller gains an incentive to signal to the access policy on the measurements that it has received.

Optimal Controller Design: The results in this thesis established the suboptimality of certainty equivalent controllers for a closed-loop system with a state-based scheduler, when the input arguments to the scheduling criterion include the applied controls. The optimal controller can be found by solving the expression in (3.18) for a 2-step horizon LQG cost. This could provide a starting point for investigation into the design of the optimal controller.

Interference Model for Event-based Systems with advanced CRMs: In Chapter 4, we identified an interference model for event-based systems that use p-persistent CSMA to resolve contention. In practise, carrier sense multiple access with collision avoidance (CSMA/CA) is the popular choice in protocols such as IEEE 802.15.4 and IEEE 802.11. Thus, an extension of this analysis to include CSMA/CA would prove very useful in analyzing realistic network setups. In the case of a synchronized network, where all the nodes initiate a transmission of their events at the same time, there is no direct extension to analyze CSMA/CA. This is

because the conditional probability of a busy channel will no longer be a constant, but vary with each backoff attempt. Modelling this variation is quite challenging. However, analyzing the performance of asynchronous networks with CSMA/CA might prove to be a simple extension, even if we only obtain an average performance over all possible initial sampling instants.

Extensions of the Stability Analysis: In Chapter 5, we identified an upper bound for the evolution of the estimation error covariance of an event-based system in a multiple access network. A direct extension would be to identify tighter upper bounds, especially for networks with low traffic levels. Furthermore, Bianchi's assumption has been shown to hold only under certain conditions, and identifying these might well provide a set of necessary conditions for stability in our problem.

In Chapter 5, the event-triggering policies we presented required an infinite number of parameters, in terms of event-thresholds or event-probabilities, to be completely characterized. In practice, it is only possible to synthesize policies with finite parameterizations. The constant-probability event-triggering policy we presented was one such example. Identifying other realizations that can be easily synthesized using the stability conditions we have is an interesting direction for future work. At the moment, we do not know which realizations are better suited to different network configurations, such as networks with high or low traffic. Analyzing and identifying design principles for event-triggering policies in multiple access networks are important to engineer designs of such networks.

Flow Control with Events and Tournaments: Event-triggering policies make it possible to regulate the flow of traffic from a node in a network of control systems, through appropriate choice of the scheduling threshold. In the results presented in this thesis, the scheduling threshold was chosen to guarantee stability, not performance. What is needed is an algorithm to choose the scheduling threshold, possibly in a socially optimal manner. This would provide a strategy to perform flow control for NCS. Attention-based tournaments also offer a simpler setup to develop a flow control mechanism for NCSs.

Chapter A

Appendix to Chapter 3

Here are the complete derivations for the estimates and the Bellman equations for the optimal controller for Example 3.6.2.

Derivation of Estimates

From (3.21), we get

$$\hat{x}_{0|0} = \begin{cases} x_0 & \delta_0 = 1 \\ \mathbb{E}[x_0 | x_0 < 0.5] & \delta_0 = 0 \end{cases} . \tag{A.1}$$

As $x_0 \sim \mathcal{N}(0,1)$, we can find the expected value

$$\bar{x}_{\delta 0} := \mathbb{E}[x_0 | x_0 < 0.5] = \int_{-\infty}^{0.5} x \phi_{x_{\delta 0}}(x) dx \ ,$$

where $\phi_{x_{\delta 0}}$ is the conditional probability distribution function (pdf) of x_0, conditioned on $x_0 < 0.5$. Thus, $\phi_{x_{\delta 0}}(x) = \phi_{x_0}(x)/Pr(x_0 < 0.5)$, where ϕ_{x_0} is the pdf of x_0. The probability of a non-transmission, $Pr(x_0 < 0.5)$, is given by

$$Pr(x_0 < 0.5) = \int_{-\infty}^{0.5} \phi_{x_0}(x) dx \ .$$

The estimation error $\tilde{x}_{0|0}$ is thus given by

$$\tilde{x}_{0|0} = \begin{cases} 0 & \delta_0 = 1 \\ x_0 - \bar{x}_{\delta 0} & \delta_0 = 0 \end{cases} . \tag{A.2}$$

The pdf of $\tilde{x}_{0|0}$ is $\phi_{\tilde{x}_0}(x) = \phi_{x_{\delta 0}}(x + \bar{x}_{\delta 0})$. The estimation error variance is given by

$$P_{0|0} = \begin{cases} 0 & \delta_0 = 1 \\ R_{\tilde{x}_0} & \delta_0 = 0 \end{cases} , \quad \text{where,} \quad \begin{aligned} R_{\tilde{x}_0} &= \mathbb{E}[(x_0 - \bar{x}_{\delta 0})^2 | x_0 < 0.5] \\ &= \int_{-\infty}^{0.5 - \bar{x}_{\delta 0}} x^2 \phi_{x_{\delta 0}}(x + \bar{x}_{\delta 0}) dx \ . \end{aligned} \tag{A.3}$$

Let us denote e_1 as the unknown part of x_1 before y_1 is received:

$$e_1 = \begin{cases} w_0 & \delta_0 = 1 \\ ax_0 + w_0 & \delta_0 = 0 \end{cases}, \quad \text{and} \quad \phi_e(\epsilon) = \begin{cases} \phi_{w_0}(\epsilon) & \delta_0 = 1 \\ \phi_{e_{\delta 0}}(\epsilon) & \delta_0 = 0 \end{cases},$$

where, ϕ_e is the pdf of e_1. The variable e_1 is the sum of two random variables if $\delta_0 = 0$, and its pdf is denoted $\phi_{e_{\delta 0}}$, and given by

$$\begin{aligned} \phi_{e_{\delta 0}}(\epsilon) &= \int_{-\infty}^{0.5} \phi_{x_{\delta 0}}(x)\phi_{w_0}(\epsilon - ax)dx \\ &= \frac{e^{-\epsilon^2/2(1+a^2)}}{\sqrt{2\pi(1+a^2)}} \left[\frac{Pr(t < \frac{1+a^2-2a\epsilon}{2\sqrt{1+a^2}})}{Pr(x_0 < 0.5)} \right], \end{aligned}$$

where, $t \sim \mathcal{N}(0,1)$. Then, at the next time instant, we get

$$\hat{x}_{1|1} = \begin{cases} x_1 & \delta_1 = 1 \\ \begin{cases} ax_0 + bu_0 + \bar{w}_0 & \delta_0 = 1 \\ bu_0 + \bar{e}_{\delta 0} & \delta_0 = 0 \end{cases} & \delta_1 = 0 \end{cases}. \tag{A.4}$$

As $w_0 \sim \mathcal{N}(0,1)$, we can find the expected value

$$\bar{w}_0 = \mathbb{E}[w_0 | w_0 < 0.5 - ax_0 - bu_0] = \int_{-\infty}^{0.5-ax_0-bu_0} w\phi_{w_0}(w)dw,$$

where ϕ_{w_0} is the pdf of w_0. Similarly, using the expression for $\phi_{e_{\delta 0}}$, we can derive

$$\begin{aligned} \bar{e}_{\delta 0} &= \mathbb{E}[ax_0 + w_0 | x_0 < 0.5, ax_0 + w_0 < 0.5 - bu_0] \\ &= \frac{1}{Pr(e_1 < 0.5 - bu_0)} \int_{-\infty}^{0.5-bu_0} \epsilon\phi_{e_{\delta 0}}(\epsilon)d\epsilon, \end{aligned}$$

where $Pr(e_1 < 0.5 - bu_0)$ is the probability of no transmission at time $k = 1$. We know that

$$Pr(e_1 < 0.5 - bu_0) = \int_{-\infty}^{0.5-bu_0} \phi_{e_{\delta 0}}(\epsilon)d\epsilon.$$

We now define $\tilde{e}_1$ as the error in estimating the term e_1 after y_1 arrives, with pdf $\phi_{\tilde{e}}$, so that

$$\tilde{e}_1 = \begin{cases} w_0 - \bar{w}_0 & \delta_0 = 1 \\ ax_0 + w_0 - \bar{e}_{\delta 0} & \delta_0 = 0 \end{cases}, \tag{A.5}$$

$$\text{and,} \quad \phi_{\tilde{e}}(\epsilon) = \begin{cases} \phi_{\tilde{w}}(\epsilon + \bar{w}_0 | w_0 < 0.5 - ax_0 - bu_0) & \delta_0 = 1 \\ \phi_{e_{\delta 0}}(\epsilon + \bar{e}_{\delta 0} | e_1 < 0.5 - bu_0) & \delta_0 = 0 \end{cases}.$$

Now, we can define the estimation error variance $P_{1|1}$ by

$$P_{1|1} = \begin{cases} 0 & \delta_1 = 1 \\ R_{e_1} & \delta_1 = 0 \end{cases} , \tag{A.6}$$

where $R_{e_1} = \mathbb{E}[\tilde{e}_1^2|\delta_1 = 0]$ is given by

$$R_{e_1} = \begin{cases} \int_{-\infty}^{0.5-ax_0-bu_0-\bar{w}_0} w^2 \frac{\phi_{w_0}(w+\bar{w}_0)}{Pr(w_0<0.5-ax_0-bu_0)} dw & \delta_0 = 1 \\ \int_{-\infty}^{0.5-bu_0-\bar{e}_{\delta 0}} \epsilon^2 \frac{\phi_{\delta 0}(\epsilon+\bar{e}_{\delta 0})}{Pr(e_1<0.5-bu_0)} d\epsilon & \delta_0 = 0 \end{cases} .$$

Note that increasing u_0 will decrease R_{e_1}.

Derivation of V_1 and V_0

We use dynamic programming to find the Bellman equations V_1 and V_0, which must be minimized to get u_1 and u_0. Using (3.14), we write

$$\begin{aligned} V_1 &= \min_{u_1} \ \mathbb{E}[x_1^2 Q_1 + u_1^2 Q_2 + x_2^2 Q_0 | \mathbb{I}_1^c] \\ &= \min_{u_1} \ \mathbb{E}[x_1^2 (Q_1 + a^2 Q_0) | \mathbb{I}_1^c] + \mathsf{tr}\{Q_0 R_w\} + u_1^2 (Q_2 + b^2 Q_0) + 2abQ_0 \hat{x}_{1|1} u_1 . \end{aligned}$$

Minimizing the above expression with respect to u_1, we get (3.34). Substituting for u_1 in the above expression for V_1, we get

$$V_1 = \mathbb{E}[x_1^2 S_1 + \mathsf{tr}\{\frac{a^2 Q_0^2 b^2}{Q_2 + b^2 Q_0} P_{1|1}\} | \mathbb{I}_1^c] + \mathsf{tr}\{Q_0 R_w\} ,$$

where $S_1 = Q_1 + a^2 Q_0 - \frac{a^2 Q_0^2 b^2}{Q_2 + b^2 Q_0}$. To derive V_0, we need to find the expected value $\mathbb{E}[P_{1|1}|\mathbb{I}_0^c]$. From the definition of $P_{1|1}$, we find that

$$\mathbb{E}[P_{1|1}|\mathbb{I}_0^c] = Pr(\delta_1 = 0|\mathbb{I}_0^c)\, \mathbb{E}[R_{e_1}|\mathbb{I}_0^c] .$$

Then, we can find the u_0 that minimizes V_0. We have

$$\begin{aligned} V_0 &= \min_{u_0} \ \mathbb{E}[x_0^2 Q_1 + u_0^2 Q_2 + V_1 | \mathbb{I}_0^c] \\ &= \min_{u_0} \ \mathbb{E}[x_0^2 (Q_1 + a^2 S_1) | \mathbb{I}_0^c] + \mathsf{tr}\{S_1 R_w\} + \mathsf{tr}\{Q_0 R_w\} \\ &\qquad + u_0^2 (Q_2 + b^2 S_1) + 2\hat{x}_{0|0} abS_1 u_0 + \frac{a^2 Q_0^2 b^2}{Q_2 + b^2 Q_0} \mathbb{E}[P_{1|1}|\mathbb{I}_0^c] . \end{aligned}$$

Chapter B

Appendix to Chapter 5

We now present some lemmas and proofs used in deriving the stability results in Chapter 5.

Properties of the Majorization Operator

We first need the following result on neat and even PDFs.

Lemma B.1. *If the PDFs ϕ_a and ϕ_b on $\mathbb{R}$ are neat and even, then $\phi_a * \phi_b$ is also neat and even.*

Proof. The PDF ϕ_b is a convex combination of indicator functions, $\chi_n(x) = 1$ for $x \in [-n, n]$ and zero otherwise. For any n, note that $\phi_a * \chi_n$ is symmetric and non-increasing. Convex combinations of neat distributions are neat, and hence, the result follows. □

We now present a series of results that use the majorization operator. The proofs presented here are adapted from (Hajek et al., 2008). These results are used in the proofs presented in Section 5.3.3.

Lemma B.2. *If the PDFs ϕ_a and ϕ_b on $\mathbb{R}^n$ are such that $\phi_a \succ \phi_b$, then $\int \phi_a^\sigma(x)h(x)dx \geq \int \phi_b^\sigma(x)h(x)dx$ for any symmetric non-increasing function h.*

Proof. The function h is a convex combination of indicator functions of balls centered at the origin. For any such indicator function, the above result is obvious from the definition of majorization. Hence, the result follows. □

Lemma B.3. *If the PDFs ϕ_a, ϕ_b and ψ on $\mathbb{R}^n$ are such that $\phi_a \succ \phi_b$, and if ϕ_a and ψ are symmetric non-increasing, then $\phi_a * \psi \succ \phi_b * \psi$.*

This proof uses Riesz's rearrangement inequality and is given as proof of Lemma 6.7 in (Hajek et al., 2008).

Lemma B.4. *If the PDFs ϕ_a and ϕ_b on $\mathbb{R}$ are such that $\phi_a \succ \phi_b$, and $\phi^+(x) \triangleq \frac{1}{|a|}\phi(\frac{x}{a})$, then $\phi_a^+ \succ \phi_b^+$.*

Proof. Using the definition of majorization, and the definitions of ϕ_a^+ and ϕ_b^+, the result can be shown to hold directly. □

Lemma B.5. *If the symmetric non-increasing PDFs ϕ_a and ϕ_b on $\mathbb{R}^n$ are such that $\phi_a \succ \phi_b$, and if h be a symmetric non-decreasing positive function, then $\int \phi_a^\sigma(x)h(x)dx \leq \int \phi_b^\sigma(x)h(x)dx$.*

Proof. Note that the function h is symmetric and quasi-concave, thus making it Schur-concave. It is known that $\mathbb{E}_{\phi_a}[h] \leq \mathbb{E}_{\phi_b}[h]$, for Schur-concave functions. Thus, the desired result follows. □

Other Lemmas

Lemma B.6. *For a general n^{th}-order plant in the network setup given by (5.1)–(5.6), the posterior variance of the innovations is less than its a priori value, i.e.,* $\mathrm{var}(\phi^e_{(I,d+1)}) \leq \mathrm{var}(\phi_{(I,d)})$.

Proof. We can find expressions for the a priori variance, denoted $\sigma^2_{(I,d)} \triangleq \mathrm{var}(\phi_{(I,d)})$, and the posterior variance, denoted $(\sigma^e_{(I,d+1)})^2 \triangleq \mathrm{var}(\phi^e_{(I,d+1)})$, as

$$\begin{aligned}\sigma^2_{(I,d)} &= \sigma^2_{\Delta^-,d} + \sigma^2_{\Delta^+,d}\,,\\ (\sigma^e_{(I,d+1)})^2 &= \sigma^2_{\Delta^-,d}\frac{1}{1-p_{\gamma,d}p_\alpha q} + \sigma^2_{\Delta^+,d}\frac{(1-p_\alpha q)}{1-p_{\gamma,d}p_\alpha q}\,,\end{aligned}$$

where $\sigma^2_{\Delta^-,d} = \int_{|\tilde{x}|\leq\Delta_d} |\tilde{x}|^2\psi(\tilde{x})d\tilde{x}$ and $\sigma^2_{\Delta^+,d} = \int_{|\tilde{x}|>\Delta_d} |\tilde{x}|^2\psi(\tilde{x})d\tilde{x}$. Now, the variance of the posterior distribution can be rewritten as

$$\begin{aligned}(\sigma^e_{(I,d+1)})^2 &= \sigma^2_{(I,d)} + \sigma^2_{\Delta^-,d}\left(\frac{1}{1-p_{\gamma,d}p_\alpha q} - 1\right) + \sigma^2_{\Delta^+,d}\left(\frac{(1-p_\alpha q)}{1-p_{\gamma,d}p_\alpha q} - 1\right)\\ &\leq \sigma^2_{(I,d)} + \max\left(\sigma^2_{\Delta^-,d}\left(\frac{1}{1-p_{\gamma,d}p_\alpha q} - 1\right) + \sigma^2_{\Delta^+,d}\left(\frac{(1-p_\alpha q)}{1-p_{\gamma,d}p_\alpha q} - 1\right)\right)\,.\end{aligned}$$

The maximum value of the first term can be found by evaluating the integral at the upper boundary to obtain $\max \sigma^2_{\Delta^-,d} = \Delta_d^2 q_{\gamma,d}$. However, the second term is negative as $\frac{(1-p_\alpha q)}{1-p_{\gamma,d}p_\alpha q} < 1$. The maximum value of this term is found by evaluating the integral at the lower boundary. Doing so, we obtain

$$\begin{aligned}(\sigma^e_{(I,d+1)})^2 &\leq \sigma^2_{(I,d)} + \Delta_d^2\left(q_{\gamma,d}\left(\frac{1}{1-p_{\gamma,d}p_\alpha q} - 1\right) - p_{\gamma,d}\left(\frac{(1-p_\alpha q)}{1-p_{\gamma,d}p_\alpha q} - 1\right)\right)\\ &= \sigma^2_{(I,d)} + 0\,,\end{aligned}$$

where it is easy to check that the terms in the inner bracket sum to zero. □

Chapter C

Appendix to Chapter 6

We now present some lemmas and proofs used in deriving the results in Chapter 6.

Proof of Lemma 6.5

Simplification of the Proof: For the purpose of this proof, we consider only the first term in (6.19). The contribution of the second term is negligible due to the required number of collisions ($\sum_{s=n-N_{\mathrm{T}}+2}^{n} C_s^n p_{\mathrm{C},s}(\alpha)$). However, the proof can easily be extended to show that the above result holds even when the second term is taken into account by requiring the probability of the attention factor to fall sufficiently fast, which is satisfied for PMFs obtained from multivariate Gamma-type distributions or Chi-squared distributions, such as the one in (6.15).

Proof. We show this result using induction on the number of tournaments. We begin with $N_{\mathrm{T}} = 1$. Then, $\mathrm{P}(W_{N_{\mathrm{T}},M-1}|\alpha_k) = p_{\mathrm{LE}}^{M-1}(\alpha)$. Due to the inequality $p_{\mathrm{LE}}(\alpha) \geq p_{\mathrm{LE}}(\alpha-1)$, the desired result $\mathrm{P}(W_{1,M-1}|\alpha) \geq \mathrm{P}(W_{1,M-1}|\alpha-1)$ follows, for $\alpha > 0$.

Let this be true for $N_{\mathrm{T}} = n$. Then, we have

$$\sum_{i=0}^{n-1} C_i^{M-1}(1-p_{\mathrm{LE}}(\alpha))^i p_{\mathrm{LE}}^{M-1-i}(\alpha) \geq \sum_{i=0}^{n-1} C_i^{M-1}(1-p_{\mathrm{LE}}(\alpha-1))^i p_{\mathrm{LE}}^{M-1-i}(\alpha-1)\ .$$

This implies that the derivative of the above expression with respect to p_{LE} must be positive. We write this as

$$(M-1)\sum_{i=0}^{n-1} C_i^{M-1}(1-p_{\mathrm{LE}})^i p_{\mathrm{LE}}^{M-2-i} - \sum_{i=0}^{n-1}((i+1)C_{i+1}^{M-1}+iC_i^{M-1})(1-p_{\mathrm{LE}})^i p_{\mathrm{LE}}^{M-2-i} \geq 0\ . \tag{C.1}$$

Now, we prove that the same holds when $N_{\mathrm{T}} = n+1$. To see this, note that the derivative of the expression $\sum_{i=0}^{n} C_i^{M-1}(1-p_{\mathrm{LE}}(\alpha))^i p_{\mathrm{LE}}^{M-1-i}(\alpha)$ has the following

terms in addition to those in (C.1). These extra terms are given by

$$\begin{aligned}&(M-1)C_n^{M-1}(1-p_{\mathrm{LE}})^n p_{\mathrm{LE}}^{M-2-n} - ((n+1)C_{n+1}^{M-1} + nC_n^{M-1})(1-p_{\mathrm{LE}})^n p_{\mathrm{LE}}^{M-2-n}\\ =&((M-1-n)C_n^{M-1} - (n+1)C_{n+1}^{M-1})(1-p_{\mathrm{LE}})^n p_{\mathrm{LE}}^{M-2-n}\ .\end{aligned}$$

Now, note that $((M-1-n)C_n^{M-1} - (n+1)C_{n+1}^{M-1})$ can be simplified as

$$\begin{aligned}(M-1-n)C_n^{M-1} - (n+1)C_{n+1}^{M-1} &= MC_n^{M-1} - (n+1)(C_{n+1}^{M-1} + C_n^{M-1})\\ &= MC_n^{M-1} - (n+1)C_{n+1}^{M}\\ &= \frac{M\cdot(M-1)!}{n!(M-1-n)!} - \frac{(n+1)\cdot M!}{(n+1)!(M-n-1)!}\\ &= 0\ .\end{aligned}$$

Thus, the derivative of the expression for $N_{\mathrm{T}} = n+1$ retains the same property as in (C.1), and is positive. This proves that for $N_{\mathrm{T}} < M-1$, the conditional probability of winning a tournament is always a non-decreasing function of α. □

Proof of Lemma 6.7

Proof. The a priori variance of $\bar{e}_k$ is given by $\sigma^2_{\bar{e},k} = \mathsf{tr}\{AK_{f,k}R_{e,k}K_{f,k}^{\top}A^{\top}\}$, where $R_{e,k}$ is given in (6.4). We denote the variance of the posterior distribution in (6.23), as $\sigma^2_{DPU,k}$. We can find expressions for both the variances as

$$\begin{aligned}\sigma^2_{\bar{e},k} &= \sigma^2_{\bar{e},k,1} + \cdots + \sigma^2_{\bar{e},k,A_{\max}}\ ,\\ \sigma^2_{DPU,k} &= \sigma^2_{\bar{e},k,1}\frac{(1-\mathrm{P}(T_{N_{\mathrm{T}},M-1}|\alpha_k=0))}{1-p_{\mathfrak{T}}} + \dots\\ &\quad + \sigma^2_{\bar{e},k,A_{\max}}\frac{(1-\mathrm{P}(T_{N_{\mathrm{T}},M-1}|\alpha_k=A_{\max}-1))}{1-p_{\mathfrak{T}}}\ ,\end{aligned}$$

where $\sigma^2_{\bar{e},k,a} = \int_{\Delta_{a-1}\le|\bar{e}|<\Delta_a}|\bar{e}|^2\psi(\bar{e})d\bar{e}$, for $1\le a\le A_{\max}$. The thresholds of the symmetric scheduling policy are denoted $\{\Delta\}_0^{A_{\max}}$, where $\Delta_0 = 0$ and $\Delta_{A_{\max}} = \infty$. Now, the variance of the posterior distribution can be rewritten as

$$\begin{aligned}\sigma^2_{DPU,k} &= \sigma^2_{\bar{e},k} + \sigma^2_{\bar{e},k,1}\rho(0) + \cdots + \sigma^2_{\bar{e},k,A_{\max}}\rho(A_{\max}-1)\\ &\le \sigma^2_{e,k} + \max\left(\sigma^2_{e,k,1}\rho(0) + \cdots + \sigma^2_{e,k,A_{\max}}\rho(A_{\max}-1)\right)\ ,\end{aligned}$$

where $\rho(a) = \frac{(1-\mathrm{P}(T_{N_{\mathrm{T}},M-1}|\alpha_k=a))-(1-p_{\mathfrak{T}})}{1-p_{\mathfrak{T}}}$. The maximum value of the latter terms can be found by evaluating the integrals at their upper boundaries, such as $\max\sigma^2_{\bar{e},k,a} = \Delta_a^2\,\mathrm{P}(\alpha_k = a-1)$. However, not all of these terms are positive. This can be seen from Lemma 6.6, as $(1-\mathrm{P}(T_{N_{\mathrm{T}},M-1}|\alpha_k=a-1)) > (1-p_{\mathfrak{T}})$ for small attention factors and vice versa. Thus, there is a value $\bar{a}\in\{0,\dots,A_{\max}-1\}$, such that $\rho(a)$ for

$a \leq \bar{a}$ are positive or zero and $\rho(a)$ for $a > \bar{a}$ are negative. The maximum value of the negative terms are found by evaluating the integrals at their lower boundaries. Doing so, we obtain

$$\sigma^2_{DPU,k} \leq \sigma^2_{\bar{e},k} + \Big(\Delta_1^2 \, \mathrm{P}(\alpha_k = 0)\rho(0) + \cdots + \Delta_{\bar{a}}^2 \, \mathrm{P}(\alpha_k = \bar{a} - 1)\rho(\bar{a} - 1) \\ - \Delta_{\bar{a}}^2 \, \mathrm{P}(\alpha_k = \bar{a})\rho(\bar{a}) - \cdots - \Delta^2_{A_{\max}-1} \, \mathrm{P}(\alpha_k = A_{\max} - 1)\rho(A_{\max} - 1) \Big) \, .$$

Now, the increasing order of the thresholds $\Delta_1 \leq \cdots \leq \Delta_{A_{\max}-1}$ implies that we can upper bound the terms in parenthesis in the above expression as

$$\begin{aligned} \sigma^2_{DPU,k} &\leq \sigma^2_{\bar{e},k} + \Delta_{\bar{a}}^2 \Big(\mathrm{P}(\alpha_k = 0)\rho(0) + \cdots + \mathrm{P}(\alpha_k = \bar{a} - 1)\rho(\bar{a} - 1) \\ &\qquad - \mathrm{P}(\alpha_k = \bar{a})\rho(\bar{a}) - \cdots - \mathrm{P}(\alpha_k = A_{\max} - 1)\rho(A_{\max} - 1) \Big) \\ &= \sigma^2_{\bar{e},k} + 0 \, , \end{aligned}$$

where it is easy to check that the terms in the inner bracket sum to zero. From this inequality, it is simple to deduce that the posterior variance of the innovations is also less than its a priori value $\sigma^2_{\bar{e},k} = \mathsf{tr}\{R_{e,k}\}$. Thus, the non-decreasing probability of transmission leads to a lower variance than would have resulted otherwise. □

Chapter D

Appendix to Chapter 7

Evaluating the Cost-to-go V_0 in Example 7.3

In the following notes, we find an expression for $\mathbb{E}\left[\hat{x}_{1|1}^2 \middle| z_0\right]$. The estimate $\hat{x}_{1|1}$ is evaluated using the knowledge that x_0 lies in the cell (θ_{l-1}, θ_l), for $z_0 = l$, and x_1 lies in the cell (δ_{j-1}, δ_j), for $z_1 = j$, respectively. The estimate can then be found as

$$\begin{aligned}\mathbb{E}\left[x_1 \middle| \{z_i\}_0^1\right] &= \int x_1 \mathbb{P}\left(x_1 \mid x_0 \in (\theta_{l-1}, \theta_l), x_1 \in (\delta_{j-1}, \delta_j)\right) dx_1 \\ &= \int_{\delta_{j-1}}^{\delta_j} \int_{\theta_{l-1}}^{\theta_l} x_1 \frac{\mathbb{P}\left(x_1, x_0\right)}{\mathbb{P}\left(x_0 \in (\theta_{l-1}, \theta_l), x_1 \in (\delta_{j-1}, \delta_j)\right)} dx_0 dx_1 \ .\end{aligned}$$

Then, the desired quantity $\mathbb{E}\left[\hat{x}_{1|1}^2 \middle| z_0\right]$ can be written as

$$\begin{aligned}\mathbb{E}\left[\hat{x}_{1|1}^2 \middle| z_0\right] &= \sum_{j=1}^{N} \mathbb{P}\left(x_1 \in (\delta_{j-1}, \delta_j) \mid x_0 \in (\theta_{l-1}, \theta_l)\right) \cdot \left(\mathbb{E}\left[x_1 \middle| z_0 = l, z_1 = j\right]\right)^2 \\ &= \frac{1}{\mathbb{P}\left(x_0 \in (\theta_{l-1}, \theta_l)\right)} \sum_{j=1}^{N} \frac{\left(\int_{\delta_{j-1}}^{\delta_j} \int_{\theta_{l-1}}^{\theta_l} x_1 \mathbb{P}\left(x_1, x_0\right) dx_0 dx_1\right)^2}{\mathbb{P}\left(x_0 \in (\theta_{l-1}, \theta_l), x_1 \in (\delta_{j-1}, \delta_j)\right)} \ .\end{aligned}$$

In the above expression, the joint probability of x_0 and x_1 is given by $\mathbb{P}\left(x_1, x_0\right) = \frac{1}{\sigma_x}\phi(\frac{x_0}{\sigma_x}) \cdot \frac{1}{\sigma_w}\phi(\frac{x_1 - a x_0 - u_0}{\sigma_w})$, where $\phi(n)$ is the probability density function of the standard normal distribution, i.e., $\phi(n) = \frac{1}{\sqrt{2\pi}} e^{-n^2/2}$. Using this, and the table of normal integrals (Owen, 1980), we evaluate the integral in the numerator as

$$\begin{aligned}\int_{\delta_{j-1}}^{\delta_j} \int_{\theta_{l-1}}^{\theta_l} x_1 \mathbb{P}\left(x_1, x_0\right) dx_0 dx_1 =& u_0 \mathbb{P}(x_0 \in (\theta_{l-1}, \theta_l), x_1 \in (\delta_{j-1}, \delta_j)) \\ &+ \vartheta\left(\frac{\delta_{j-1} - u_0}{\sigma_2}, \frac{\delta_j - u_0}{\sigma_2}\right) \ ,\end{aligned}$$

where $\vartheta(\underline{r}, \bar{r})$ is given in (7.14).

Thus, the desired quantity $\mathbb{E}\left[\hat{x}_{1|1}^2 \middle| z_0\right]$ can be written as

$$\mathbb{E}\left[\hat{x}_{1|1}^2 \mid z_0\right] = u_0^2 + 2au_0\sigma_x \frac{\phi\left(\frac{\theta_{l-1}}{\sigma_x}\right) - \phi\left(\frac{\theta_l}{\sigma_x}\right)}{\mathbb{P}\left(x_0 \in (\theta_{l-1}, \theta_l)\right)} + \frac{\sum_{j=1}^{N} \vartheta^2\left(\frac{\delta_{j-1}-u_0}{\sigma_2}, \frac{\delta_j - u_0}{\sigma_2}\right)}{\mathbb{P}\left(x_0 \in (\theta_{l-1}, \theta_l)\right)} .$$

Now, note that $\hat{x}_{0|0} = \sigma_x \left(\phi\left(\frac{\theta_{l-1}}{\sigma_x}\right) - \phi\left(\frac{\theta_l}{\sigma_x}\right)\right) / \mathbb{P}\left(x_0 \in (\theta_{l-1}, \theta_l)\right)$. Thus, the cost-to-go to be minimized can be rewritten as in (7.13).

Index

Bibliography

N. Abramson. The ALOHA system: another alternative for computer communications. In *Proceedings of the November 17-19, 1970, fall joint computer conference*, AFIPS '70 (Fall), pages 281–285, New York, NY, USA, November 1970. ACM.

J. Åkerberg, M. Gidlund, and M. Björkman. Future research challenges in wireless sensor and actuator networks targeting industrial automation. In *Proceedings of the 9th IEEE International Conference on Industrial Informatics (INDIN)*, pages 410–415, July 2011.

I. Akyildiz, J. McNair, L. Martorell, R. Puigjaner, and Y. Yesha. Medium access control protocols for multimedia traffic in wireless networks. *IEEE Network*, 13 (4):39–47, 1999.

I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, Aug. 2002.

B. Andrievsky, A. Matveev, and A. Fradkov. Control and estimation under information constraints: Toward a unified theory of control, computation and communications. *Automation and Remote Control*, 71(4):572–633, 2010.

A. Anta and P. Tabuada. On the Benefits of Relaxing the Periodicity Assumption for Networked Control Systems over CAN. In *Proceedings of the 30th IEEE Real-Time Systems Symposium*, pages 3–12. IEEE, Dec. 2009.

A. Anta and P. Tabuada. To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9):2030–2042, Sept. 2010.

D. Antunes, J. Hespanha, and C. Silvestre. Volterra integral approach to impulsive renewal systems: Application to networked control. *IEEE Transactions on Automatic Control*, 57(3):607–619, March 2011.

J. Araujo, M. Mazo, A. Anta, P. Tabuada, and K. Johansson. System architectures, protocols and algorithms for aperiodic wireless control systems. *IEEE Transactions on Industrial Informatics*, 10(1):175–184, February 2014.

K. J. Åström. *Introduction to Stochastic Control Theory*. Academic Press, 1970. Republished by Dover Publications, 2006.

K. J. Åström. Event based control. In A. Astolfi and L. Marconi, editors, *Analysis and Design of Nonlinear Control Systems*, pages 127–147. Springer Berlin Heidelberg, 2008.

K. J. Åström and B. Bernhardsson. Comparison of periodic and event based sampling for first order stochastic systems. In *Proceedings of the 14th IFAC World Congress*, volume 11, pages 301–306, 1999.

K. J. Åström and B. M. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 2, pages 2011–2016, Dec. 2002.

K. J. Åström and B. Wittenmark. *Adaptive Control*. Addison-Wesley, 1995.

L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.

J. Baillieul and P. Antsaklis. Control and communication challenges in networked real-time systems. *Proceedings of the IEEE*, 95(1):9–28, 2007.

R. Bansal and T. Başar. Simultaneous design of measurement and control strategies for stochastic systems with feedback. *Automatica*, 25(5):679–694, 1989.

L. Bao, M. Skoglund, and K. Johansson. Iterative encoder-controller design for feedback control over noisy channels. *IEEE Transactions on Automatic Control*, 56(2):265–278, 2011.

Y. Bar-Shalom and E. Tse. Dual effect, certainty equivalence, and separation in stochastic control. *IEEE Transactions on Automatic Control*, 19:494–500, October 1974.

J. S. Baras and A. Bensoussan. Optimal Sensor Scheduling in Nonlinear Filtering of Diffusion Processes. *SIAM Journal on Control and Optimization*, 27(4):786–813, 1989.

G. Battistelli, A. Benavoli, and L. Chisci. Data-driven communication for state estimation with sensor networks. *Automatica*, 48(5):926–935, May 2012.

R. E. Bellman. The theory of dynamic programming. Technical report, The RAND Corporation, July 1954.

B. Bernhardsson. Dual control of a first-order system with two possible gains. *International Journal of Adaptive Control and Signal Processing*, 3(1):15–22, 1989.

D. P. Bertsekas and S. E. Shreve. *Stochastic optimal control: The discrete time case*, volume 139 of *Mathematics in Science and Engineering*. Academic Press Inc., New York, 1978. ISBN 0-12-093260-1.

G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, Mar. 2000.

G. Bianchi, I. Tinnirello, and L. Scalia. Understanding 802.11e contention-based prioritization mechanisms and their coexistence with legacy 802.11 stations. *IEEE Network*, 19(4):28–34, July 2005.

R. Blind and F. Allgöwer. Analysis of networked event-based control with a shared communication medium: Part I - Pure ALOHA. In *Proceedings of the IFAC World Congress*, pages 10092–10097, 2011a.

R. Blind and F. Allgöwer. Analysis of networked event-based control with a shared communication medium: Part II - Slotted ALOHA. In *Proceedings of the IFAC World Congress*, pages 8830–8835, 2011b.

C. Bordenave, D. McDonald, and A. Proutiere. A particle system in interaction with a rapidly varying environment: Mean field limits and applications. *Networks and Heterogeneous Media*, 5(1):31–62, March 2010.

V. Borkar and S. K. Mitter. LQG control with communication constraints. In A. Paulraj, V. Roychowdhury, and C. D. Schaper, editors, *Communications, Computation, Control, and Signal Processing*, pages 365–373. Springer US, 1997.

V. Borkar, S. K. Mitter, and S. Tatikonda. Markov control problems under communication constraints. *Communications in Information and Systems*, 1(1):15–32, 2001a.

V. S. Borkar, S. K. Mitter, and S. Tatikonda. Optimal Sequential Vector Quantization of Markov Sources. *SIAM Journal on Control and Optimization*, 40(1): 135–148 (electronic), Nov. 2001b.

P. Breun and W. Utschick. On transmitter design in power constrained LQG control. In *American Control Conference*, pages 4979–4984, 2008.

G. C. Buttazzo. *Hard Real-time Computing Systems: Predictable Scheduling Algorithms And Applications (Real-Time Systems Series)*. Springer-Verlag TELOS, Santa Clara, CA, USA, 2004. ISBN 0387231374.

N. Cardoso de Castro, C. Canudas de Wit, and F. Garin. Energy-aware wireless networked control using radio-mode management. In *Proceedings of the American Control Conference*, pages 2836–2841, June 2012a.

N. Cardoso de Castro, D. Quevedo, F. Garin, and C. Canudas de Wit. Smart energy-aware sensors for event-based control. In *IEEE 51st Annual Conference on Decision and Control*, pages 7224–7229, Dec 2012b.

A. Cervin and T. Henningsson. Scheduling of event-triggered controllers on a shared network. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 3601–3606, Dec. 2008.

H. Chernoff. Backward induction in dynamic programming. Unpublished, 1963.

C. Y. Chong and S. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, Aug. 2003.

D. Christmann, R. Gotzhein, S. Siegmund, and F. Wirth. Realization of try-once-discard in wireless multi-hop networks. *IEEE Transactions on Industrial Informatics*, 10(1):17–26, feb 2014.

R. E. Curry. *Estimation and Control with Quantized Measurements*. MIT Press, 1970.

B. Demirel, V. Gupta, and M. Johansson. On the trade-off between control performance and communication cost for event-triggered control over lossy networks. In *Proceedings of the 12th European Control Conference*, pages 1168–1174, 2013.

M. C. F. Donkers and W. P. M. H. Heemels. Output-based event-triggered control with guaranteed L_∞-gain and improved event-triggering. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 3246–3251, 2010.

M. C. F. Donkers and W. P. M. H. Heemels. Output-based event-triggered control with guaranteed L_∞-gain and improved and decentralised event-triggering. *IEEE Transactions on Automatic Control*, 57(6):1362–1376, 2012.

E. Dynkin. Controlled random sequences. *Theory of Probability and its Applications*, 10(1):1–14, 1965.

F. Farokhi. *Decentralized Control of Networked Systems: Information Asymmetries and Limitations*. PhD thesis, Department of Automatic Control, KTH Royal Institute of Technology, Sweden, March 2014.

F. Farokhi and K. Johansson. Stochastic sensor scheduling for networked control systems. *IEEE Transactions on Automatic Control*, 2014. ISSN 0018-9286. doi: 10.1109/TAC.2014.2298733. To Appear.

A. A. Feldbaum. Dual-control theory. I. *Automation and Remote Control*, 21: 874–880, 1960.

A. A. Feldbaum. Dual-control theory. IV. *Automation and Remote Control*, 22: 109–121, 1961.

X. Feng and K. Loparo. Active probing for information in control systems with quantized state measurements: a minimum entropy approach. *IEEE Trans. Automatic Control*, 42(2):216–238, 1997.

N. M. Filatov and H. Unbehauen. Survey of adaptive dual control methods. *IEE Proceedings - Control Theory and Applications*, 147(1):118–128, Jan. 2000.

T. Fischer. Optimal quantized control. *IEEE Transaction on Automatic Control*, 27(4):996–998, 1982.

J. Freudenberg, R. Middleton, and J. Braslavsky. Minimum variance control over a gaussian communication channel. *IEEE Transactions on Automatic Control*, 56 (8):1751–1765, 2011.

M. Fu. Lack of separation principle for quantized linear quadratic gaussian control. *IEEE Transactions on Automatic Control*, 57(9):2385–2390, 2012.

A. Goldsmith and S. Wicker. Design challenges for energy-constrained ad hoc wireless networks. *IEEE Wireless Communications*, 9(4):8–27, August 2002.

G. Goodwin, E. Silva, and D. Quevedo. A brief introduction to the analysis and design of networked control systems. In *Proceedings of the IEEE Control and Decision Conference*, pages 1–13, 2008.

R. Gotzhein and T. Kuhn. Decentralized Tick Synchronization for Multi-hop Medium Slotting in Wireless Ad Hoc Networks using Black Bursts. In *Proceedings of the 5th Annual IEEE ComSoc Conference on Sensor, Mesh, and Ad Hoc Communications and Networks*, pages 422–431, 2008.

A. C. V. Gummalla and J. O. Limb. Wireless medium access control protocols. *IEEE Communications Surveys Tutorials*, 3(2):2–15, 2000.

V. Gupta and N. Martins. On stability in the presence of analog erasure channel between the controller and the actuator. *IEEE Transactions on Automatic Control*, 55(1):175–179, Jan. 2010.

V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42(2):251–260, Feb. 2006.

V. Gupta, B. Hassibi, and R. M. Murray. Optimal LQG control across packet-dropping links. *Systems & Control Letters*, 56(6):439–446, 2007.

B. Hajek. Jointly optimal paging and registration for a symmetric random walk. In *Proceedings of the 2002 IEEE Information Theory Workshop*, pages 20–23, oct 2002.

B. Hajek, K. Mitzel, and S. Yang. Paging and registration in cellular networks: Jointly optimal policies and an iterative algorithm. *IEEE Transactions on Information Theory*, 54(2):608–622, Feb. 2008.

Y. Halevi and A. Ray. Integrated communication and control systems. I - analysis. *ASME, Transactions, Journal of Dynamic Systems, Measurement and Control.*, 110:367–373, December 1988a.

Y. Halevi and A. Ray. Integrated communication and control systems. II - design considerations. *ASME, Transactions, Journal of Dynamic Systems, Measurement and Control.*, 110:367–373, December 1988b.

D. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, Jan 1997.

HART Communication Foundation. *WirelessHART Data Sheet*, 2007. Datasheet.

W. Heemels, K. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *Proceedings of the IEEE 51st Annual Conference on Decision and Control*, pages 3270–3285, Dec 2012.

W. P. M. H. Heemels, J. H. Sandee, and P. P. J. Van Den Bosch. Analysis of event-driven controllers for linear systems. *International Journal of Control*, 81 (4):571–590, Apr. 2008.

T. Henningsson. Sporadic event-based control using path constraints and moments. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 4723 – 4728, 2011.

T. Henningsson. *Stochastic Event-Based Control and Estimation.* PhD thesis, Department of Automatic Control, Lund University, Sweden, Dec. 2012.

T. Henningsson and A. Cervin. A simple model for the interference between event-based control loops using a shared medium. In *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.

T. Henningsson, E. Johannesson, and A. Cervin. Sporadic event-based control of first-order linear stochastic systems. *Automatica*, 44(11):2890–2895, Nov. 2008.

E. Henriksson, D. E. Quevedo, H. Sandberg, and K. H. Johansson. Self-Triggered Model Predictive Control for Network Scheduling and Control. In *Proceedings of the 8th IFAC International Symposium on Advanced Control of Chemical Processes*, volume 8, pages 432–438, 2012.

A. Hernandez. Modification of the IEEE 802.15.4 implementation extended gts implementation. Technical report, KTH Royal Institute of Technology, July 2011. URL `http://tinyos.cvs.sourceforge.net/viewvc/tinyos/tinyos-2.x-contrib/kth/index.html`.

K. Herring and J. Melsa. Optimum Measurements for Estimation. *IEEE Transactions on Automatic Control*, 19(3):264–266, 1974.

J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. In *Proceedings of the IEEE*, volume 95, pages 138–162, Jan. 2007.

Y.-C. Ho. Team decision theory and information structures. *Proceedings of the IEEE*, 68(6):644–654, June 1980.

Y.-C. Ho and K.-C. Chu. Team Decision Theory and Information Structures in Optimal Control Problems - Part I. *IEEE Transactions on Automatic Control*, 17(1):15–22, February 1972.

S. H. Hong. Scheduling algorithm of data sampling times in the integrated communication and control systems. *IEEE Transactions on Control Systems Technology*, 3(2):225 –230, June 1995.

W. A. Horn. Some simple scheduling algorithms. *Naval Research Logistics Quarterly*, 21(1):177–185, 1974.

IEEE. *IEEE 802.11 standard: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999. URL http://www.ieee802.org/11.

IEEE. *IEEE 802.15.4 standard: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, 2006. URL http://www.ieee802.org/15/pub/TG4.html.

International Society of Automation. ISA-SP100 wireless systems for automation website. http://www.isa.org/isa100, 2010.

K. H. Johansson, M. Törngren, and L. Nielsen. Vehicle Applications of Controller Area Network. In D. Hristu-Varsakelis and W. Levine, editors, *Handbook of Networked and Embedded Control Systems*, Control Engineering, pages 741–765. Birkhäuser Boston, 2005.

D. P. Joseph and T. J. Tou. On linear control theory. *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, 80(4):193–196, 1961.

T. Kailath, A. Sayed, and B. Hassibi. *Linear estimation*. Prentice-Hall information and system sciences series. Prentice Hall, 2000. ISBN 9780130224644.

S. Kar, B. Sinopoli, and J. Moura. Kalman Filtering With Intermittent Observations: Weak Convergence to a Stationary Distribution. *IEEE Transactions on Automatic Control*, 57(2):405–420, February 2012.

J. Kay and P. Lauder. A fair share scheduler. *Communications of the ACM*, 31: 44–55, 1988.

L. Kleinrock and F. Tobagi. Packet switching in radio channels: Part I–carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23(12):1400–1416, Dec. 1975.

L. Kleinrock, Fouad, and A. Tobagi. Carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23: 1400–1416, 1983.

F. Kozin. A survey of stability of stochastic systems. *Automatica*, 5(1):95–112, 1969.

A. S. Krishnamoorthy and M. Parthasarathy. A multivariate gamma-type distribution. *The Annals of Mathematical Statistics*, 22:549–557, 1951.

KTH wireless NCS code repository, October 2013. URL `http://code.google.com/p/kth-wsn/`.

R. Larson. Optimum quantization in dynamic systems. *IEEE Transactions on Automatic Control*, 12(2):162–168, 1967.

E. Lee. Cyber Physical Systems: Design Challenges. In *Proceedings of the 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing*, pages 363–369, May 2008.

M. Lemmon and X. S. Hu. Almost sure stability of networked control systems under exponentially bounded bursts of dropouts. In *Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control*, Apr. 2011.

M. Lemmon, T. Chantem, X. S. Hu, and M. Zyskowski. On self triggered full-information H-infinity controllers. In *Proceedings of the 10th International Conference on Hybrid Systems: Computation and Control*, pages 371–384, 2007.

P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, et al. TinyOS: An operating system for wireless sensor networks. *Ambient Intelligence*, 2004.

F. L. Lian, J. R. Moyne, and D. M. Tilbury. Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet. *IEEE Control Systems Magazine*, 21(1):66–83, Feb. 2001.

G. M. Lipsa and N. C. Martins. Remote State Estimation With Communication Costs for First-Order LTI Systems. *IEEE Transactions on Automatic Control*, 56(9):2013–2025, Sept. 2011.

C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal ACM*, 20(1):46–61, Jan. 1973.

X. Liu and A. Goldsmith. Wireless medium access control in networked control systems. In *Proceedings of the American Control Conference*, volume 4, pages 3605–3610, June 2004.

R. Marleau and J. Negro. Comments on "optimum quantization in dynamic systems". *IEEE Transactions on Automatic Control*, 17(2):273–274, 1972.

A. S. Matveev and A. V. Savkin. The problem of state estimation via asynchronous communication channels with irregular transmission times. *IEEE Transactions on Automatic Control*, 48(4):670–676, April 2003.

A. S. Matveev and A. V. Savkin. The problem of LQG optimal control via a limited capacity communication channel. *Systems & Control Letters*, 53(1):51–64, 2004.

X. Meng and T. Chen. Optimal Sampling and Performance Comparison of Periodic and Event Based Impulse Control. *IEEE Transactions on Automatic Control*, 57 (12):3252–3259, 2012.

S. K. Mitter. Control with limited information. *European Journal of Control*, 7 (2-3):122–131, 2001.

A. Molin and S. Hirche. On LQG joint optimal scheduling and control under communication constraints. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 5832–5838, Dec. 2009.

A. Molin and S. Hirche. Structural characterization of optimal event-based controllers for linear stochastic systems. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 3227–3233, 2010.

A. Molin and S. Hirche. Adaptive event-triggered control over a shared network. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 6591–6596, Dec. 2012.

A. Molin and S. Hirche. On the Optimality of Certainty Equivalence for Event-Triggered Control Systems. *IEEE Transactions on Automatic Control*, 58(2): 470–474, 2013.

R. M. Murray et al. *Control in an Information Rich World: Report of the Panel on Future Directions in Control, Dynamics, and Systems*. Society for Industrial and Applied Mathematics, 2003. URL http://www.cds.caltech.edu/~murray/cdspanel.

G. N. Nair and R. J. Evans. Stabilizability of stochastic linear systems with finite feedback data rates. *SIAM Journal on Control and Optimization*, 43(2):413–436, Feb. 2004.

G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans. Feedback control under data rate constraints: An overview. *Proceedings of the IEEE*, 95(1):108–137, Jan. 2007.

A. Nayyar. *Sequential Decision Making in Decentralized Systems*. PhD thesis, The University of Michigan, 2011.

A. Nayyar, T. Başar, D. Teneketzis, and V. Veeravalli. Optimal strategies for communication and remote estimation with an energy harvesting sensor. *IEEE Transactions Automatic Control*, 58(9):2246–2260, 2013.

A. Nayyar, A. Gupta, C. Langbort, and T. Başar. Common information based markov perfect equilibria for stochastic games with asymmetric information: Finite games. *IEEE Transactions on Automatic Control*, 59(3):555–570, March 2014.

J. Nilsson. *Real-Time Control Systems with Delays*. PhD thesis, Lund Institute of Technology, Department of Automatic Control, Sweden, January 1998.

P. G. Otanez, J. R. Moyne, and D. M. Tilbury. Using deadbands to reduce communication in networked control systems. In *Proceedings of the American Control Conference*, volume 4, pages 3015–3020, 2002.

D. B. Owen. A table of normal integrals. *Communications in Statistics B—Simulation and Computation*, 9(4):389–419, 1980. ISSN 0361-0918.

N. Pereira, B. Andersson, and E. Tovar. WiDom: A dominance protocol for wireless medium access. *IEEE Transactions on Industrial Informatics*, 3(2):120–130, May 2007.

J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks*, pages 364–369, Apr 2005.

S. Pollin, M. Ergen, S. Ergen, B. Bougard, L. Der Perre, I. Moerman, A. Bahai, P. Varaiya, and F. Catthoor. Performance analysis of slotted carrier sense IEEE 802.15.4 medium access layer. *IEEE Transactions on Wireless Communications*, 7(9):3359–3371, September 2008.

M. Rabi. *Packet based Inference and Control*. PhD thesis, Institute for Systems Research, University of Maryland, 2006. URL http://hdl.handle.net/1903/6612.

M. Rabi and K. H. Johansson. Scheduling packets for event-triggered control. In *Proceedings of 10th European Control Conference*, pages 3779–3784, 2009a.

M. Rabi and K. H. Johansson. Optimal stopping for updating controls. In *Proccedings of the second International workshop on sequential methods*, UTT, Troyes, France, June 2009b.

M. Rabi, L. Stabellini, A. Proutiere, and M. Johansson. Networked estimation under contention-based medium access. *International Journal of Robust and Nonlinear Control*, 20(2):140–155, 2010.

M. Rabi, G. Moustakides, and J. Baras. Adaptive sampling for linear state estimation. *SIAM Journal on Control and Optimization*, 50(2):672–702, 2012.

R. Rajkumar, I. Lee, L. Sha, and J. Stankovic. Cyber-physical systems: The next computing revolution. In *Proceedings of the 47th ACM/IEEE Design Automation Conference*, pages 731–736, June 2010.

I. Ramachandran, A. K. Das, and S. Roy. Analysis of the contention access period of ieee 802.15.4 mac. *ACM Transactions on Sensor Networks*, 3(1), Mar. 2007.

R. Ramaswami and K. Parhi. Distributed scheduling of broadcasts in a radio network. In *Proceedings of the Eighth Annual Joint Conference of the IEEE Computer and Communications Societies. Technology: Emerging or Converging*, volume 2, pages 497–504, April 1989.

C. Ramesh. *Contention-based Multiple Access Architectures for Networked Control Systems*. Licentiate thesis, KTH Royal Institute of Technology, 2011. URL http://kth.diva-portal.org/smash/record.jsf?pid=diva2:397755.

A. Ray. Performance evaluation of medium access control protocols for distributed digital avionics. *ASME, Transactions, Journal of Dynamic Systems, Measurement and Control.*, 109:370–377, December 1987.

Robert Bosch GmbH. *Bosch CAN Specification, ver. 2*. Stuttgart, 1991. URL http://www.semiconductors.bosch.de/pdf/can2spec.pdf.

R. Rom and M. Sidi. *Multiple access protocols: performance and analysis*. Springer-Verlag New York, Inc., New York, NY, USA, 1990. ISBN 0-387-97253-6.

J. G. Root. Optimum control of non-Gaussian linear stochastic systems with inaccessible state variables. *SIAM Journal on Control and Optimization*, 7:317–323, 1969.

W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, Inc., third edition, 1976.

G. Sauer and J. Melsa. Stochastic control with continuously variable observation costs for a class of discrete nonlinear systems. *IEEE Transactions on Automatic Control*, 19(3):234–239, Jun 1974.

A. Savkin, R. Evans, and E. Skafidas. The problem of optimal robust sensor scheduling. In *Proceedings of the 39th IEEE Conference on Decision and Control*, volume 4, pages 3791–3796, 2000.

L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry. Foundations of control and estimation over lossy networks. *Proceedings of the IEEE*, 95(1): 163–187, January 2007.

A. N. Shiryaev. On the theory of decision functions and control by the process of observation on partial data [in russian]. In *Transactions of the Third Prague Conference on Information Theory, Statistical Decision Functions, and Random Processes*, pages 657–672, Prague, 1964.

S. C. Smith and P. Seiler. Estimation with lossy measurements: jump estimators for jump systems. *IEEE Transactions on Automatic Control*, 48(12):2163–2171, Dec. 2003.

C. Striebel. Sufficient statistics in the optimum control of stochastic systems. *Journal of Mathematical Analysis and Applications*, 12(3):576–592, Dec. 1965.

M. Tabbara and D. Nesić. Input-Output Stability of Networked Control Systems with Stochastic Protocols and Channels. *IEEE Transactions on Automatic Control*, 53(5):1160–1175, 2008.

P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, September 2007. ISSN 0018-9286. doi: 10.1109/TAC.2007.904277.

A. Tanenbaum. *Computer Networks*. Prentice Hall Professional Technical Reference, 4th edition, 2002. ISBN 0130661023.

S. Tatikonda and S. Mitter. Control under communication constraints. *IEEE Transactions on Automatic Control*, 49(7):1056–1068, 2004a.

S. Tatikonda and S. Mitter. Control over noisy channels. *IEEE Transactions on Automatic Control*, 49:1196–1201, 2004b.

S. Tatikonda, A. Sahai, and S. Mitter. Stochastic linear control over a communication channel. *IEEE Transactions on Automatic Control*, 49(9):1549–1561, Sept. 2004.

R. Tomovic and G. Bekey. Adaptive sampling based on amplitude sensitivity. *IEEE Transactions on Automatic Control*, 11(2):282–284, Apr. 1966.

H. van de Water and J. Willems. The Certainty Equivalence property in Stochastic control theory. *IEEE Transactions on Automatic Control*, 26(5):1080–1087, 1981.

P. Varaiya and J. Walrand. On delayed sharing patterns. *IEEE Transactions on Automatic Control*, 23(3):443–445, 1978.

J. Walrand and P. Varaiya. Optimal causal coding - decoding problems. *IEEE Transactions on Information Theory*, 29(6):814–820, 1983.

G. Walsh and H. Ye. Scheduling of networked control systems. *IEEE Control Systems Magazine*, 21(1):57–65, 2001.

G. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. *IEEE Transactions on Control Systems Technology*, pages 2876–2880, 1999.

X. Wang and M. D. Lemmon. Event design in event-triggered feedback control systems. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 2105–2110, 2008.

X. Wang and M. D. Lemmon. Event-Triggering in Distributed Networked Control Systems. *IEEE Transactions on Automatic Control*, 56(3):586–601, Mar. 2011.

J. Weimer, J. Araujo, and K. H. Johansson. Distributed event-triggered estimation in networked systems. In *Proceedings of the 4th IFAC conference on Analysis and Design of Hybrid Systems*, pages 178–185, June 2012.

A. Willig. Recent and emerging topics in wireless industrial communications: A selection. *IEEE Transactions on Industrial Informatics*, 4(2):102–124, may 2008.

A. Willig, K. Matheus, and A. Wolisz. Wireless technology in industrial networks. *Proceedings of the IEEE*, 93(6):1130–1151, june 2005.

H. R. Wilson. *Spikes, Decisions, and Actions: The Dynamical Foundations of Neuroscience*. Oxford University Press, 1999.

H. S. Witsenhausen. A counterexample in stochastic optimum control. *SIAM Journal on Control*, 6(1):131–147, 1968.

H. S. Witsenhausen. Separation of estimation and control for discrete time systems. *Proceedings of the IEEE*, 59(11):1557–1566, November 1971.

W. Wonham. On the separation theorem of stochastic control. *SIAM Journal on Control*, 6(2):312–326, 1968.

W. Wu and A. Arapostathis. Optimal control of stochastic systems with costly observations - the general markovian model and the LQG problem. In *Proceedings of the American Control Conference*, volume 1, pages 294–299, 2005.

M. Xia, V. Gupta, and P. J. Antsaklis. Networked State Estimation over a Shared Communication Medium. In *Proceedings of the American Control Conference*, pages 4128–4133, June 2013.

Y. Xu and J. P. Hespanha. Optimal communication logics in networked control systems. In *Proceedings of the 43rd Conference on Decision and Control*, volume 4, pages 3527–3532, Dec. 2004a.

Y. Xu and J. P. Hespanha. Communication logics for networked control systems. In *Proceedings of the American Control Conference*, pages 572–577, 2004b.

Y. Xu, J. Winter, and W.-C. Lee. Dual prediction-based reporting for object tracking sensor networks. In *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 154–163, Aug. 2004.

J. K. Yook, D. M. Tilbury, and N. R. Soparkar. Trading computation for bandwidth: reducing communication in distributed control systems using state estimators. *IEEE Transactions on Control Systems Technology*, 10(4):503–518, July 2002.

H. Yu and P. J. Antsaklis. Event-Triggered Output Feedback Control for Networked Control Systems using Passivity: Triggering Condition and Limitations. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 199–204, 2011.

S. Yüksel. Jointly optimal LQG quantization and control policies for multi-dimensional linear gaussian sources. In *Proceedings of the 50th Annual Allerton Conference onCommunication, Control, and Computing*, pages 466–473, 2012.

S. Yüksel. On optimal causal coding of partially observed markov sources in single and multiterminal settings. *IEEE Transactions on Information Theory*, 59(1): 424–437, 2013.

W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84–99, Feb. 2001.

ZigBee Alliance. *ZigBee Specification*, 2005. URL `http://www.caba.org/standard/zigbee.html`.