On LQG-Optimal Event-Based Sampling

Thelander Andrén, Marcus

2020

Document Version:
Publisher's PDF, also known as Version of record

On LQG-Optimal Event-Based Sampling

Marcus Thelander Andrén

LUND
UNIVERSITY

Department of Automatic Control

Ph.D. Thesis TFRT-1128
ISBN 978-91-7895-657-9 (print)
ISBN 978-91-7895-656-2 (web)
ISSN 0280–5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Media-Tryck.
Lund 2020

Till Sara och Signe

Abstract

Event-based control is a promising concept for the design of resource-efficient modern feedback systems based on wireless communication. In this concept, events such as sampling, actuation, and data transmissions are triggered reactively based on monitored control performance rather than a periodic timer. This avoids redundant use of limited system resources such as battery lifetime, network bandwidth, and CPU utilization. In this thesis, we investigate how sampling and communication events should be triggered to fully exploit the potential of event-based control, and take a step towards a complete event-based design scheme based on the classic linear–quadratic–Gaussian (LQG) framework.

The design of the event trigger is formulated as a trade-off between a quadratic cost on control performance and the average event rate. This problem, dubbed the *optimal sampling problem*, is equivalent to an impulse control problem. The optimal even trigger is well-known for first-order systems, where it corresponds to a scalar symmetric threshold on the monitored control performance. In this thesis, we consider systems of higher order, where the shape of the optimal threshold is generally unknown. For two new system classes with previously unknown solutions, we prove that the optimal threshold is ellipsoidal for all system orders. Additionally, we propose two numerical methods for finding the optimal threshold shape for general systems. Both methods are numerically validated and show benefits in terms of accuracy and solver time compared to a reference method based on dynamic programming. We also extend a well-cited performance comparison between event-based and periodic triggering for a first-order integrator system to the higher-order case. The factor three improvement observed for event-based triggering in the first-order case is shown to constitute an upper bound for higher-order systems.

Using one of the proposed numerical methods, we study the shape of the optimal threshold for different classes of second-order systems. Findings suggest that ellipsoidal thresholds are good approximations for most system classes, and can give near-optimal performance even in exceptional cases.

Therefore, we argue that ellipsoids constitute a good general alternative to the exact optimal threshold, especially for higher-order systems where an optimal solution is impractical to obtain numerically. To find suitable ellipse parameters for any system order, we propose and validate a previously presented Monte Carlo approach applied to the optimal sampling problem.

Suboptimal but simpler designs in the form of event-based proportional–integral–derivative (PID) control are also considered in this thesis. Inspired by results from LQG-optimal sampled-data control, we derive an "ideal" (in the LQG sense) sampled-data PID implementation, from which a range of design options of varying complexity for event-based PID control is proposed. Based on numerical evaluations, we conclude that event triggering based on both the proportional and derivative part of the control signal is desirable from a performance perspective.

Finally, this thesis also consider stochastic triggering, where events are triggered according to a certain probability. This type of event trigger is considered for a prototypical event-based state estimation problem, where the sensor intermittently triggers measurement transmissions to a remote estimator. With stochastic triggering, the LQG-optimal estimator takes the form of a Kalman filter with event-dependent measurement updates, where the quality of the estimates largely depends on the choice of trigger signal in the sensor. In this thesis, we propose two policies for stochastic triggering, both featuring predictions in the sensor for improved estimator performance. In addition to comparing well to other policies previously proposed in the literature, one of the policies also gives simplified performance analysis, with expressions for average transmission rate and estimator performance that are easily computed without resorting to Monte Carlo simulations.

Acknowledgments

Now that I stand here on the finish line, I realize that I have a lot of people to thank for making this journey such a fun, rewarding, and instructive experience. First, I would like to give special thanks to my supervisor Anton Cervin, who introduced me to the exciting field of event-based control and has guided me from start to finish. I could not have wished for a better supervisor. Thank you for always hearing me out when I ramble about optimal stopping problems and for the support you have shown me in times of self-doubt.

I am also very grateful to my co-supervisors Bo Bernhardsson and Kristian Soltesz. With his knack for finding intuitive explanations to even the most intricate mathematical problems, Bo has helped me over several hurdles. Kristian, thank you for being such an upbeat and friendly partner for research discussions, and for always reminding me about the practical aspects in an otherwise theory-heavy field.

To all my colleagues at the department—both current and former—I would like to say thank you for making it such a warm and friendly workplace. I am especially grateful to all the people I have had the pleasure of sharing an office with. Josefin Berner, Olof Sörnmo, and Andreas Stolt; thank you for taking me under your wing as a fresh Ph.D. student and for showing me the ropes (sometimes the hard way, as I quickly learned after forgetting to lock my computer). Morten Hagdrup, Marcus Greiff, Claudio Mandrioli, and Richard Pates; thank you for all the great discussions, both research-related and otherwise, and for taking me seriously when I talk about "lemon-shaped" trigger policies.

Also at the department, I would like to give special thanks to Mattias Fält and Fredrik Bagge Carlson. Without your enthusiastic help with Julia programming and speeding up my code, I would have spent a lot more time staring at completion bars slowly filling up. Johan Ruuskanen, thank you for all the interesting discussions concerning event-based control, statistics, and beer brewing. I thank Tommi Berner for being my guide during my brief foray into the field of cloud control. Gautham Nayak Seetanadi, thank you

for taking the initiative on many fun activities at the department, including after works, game nights, and floorball tournaments. Anders Robertsson, thank you for making the teaching experience at Beihang university such a fun time, and for inspiring me to further hone my teaching skills. Tore Hägglund, thank you for your support in times of doubt, and for giving me an excellent first introduction to the world of control theory. Mika Nishimura, and Kaoru Yamamoto, thank you for indulging me to practice my poor Japanese with you during coffee breaks and e-mail correspondence. Anders Nilsson and Anders Blomdell, thank you for helping me with all things computer-related over the years. Also, a big thank you to Eva Westin for being so supportive since my very first day at the department.

This thesis would look a lot less pretty without the LaTeXwizardry provided by Leif Andersson. For proof-reading, I would like to thank Anton Cervin, Bo Bernhardsson, Kristian Soltesz, Olof Troeng, Leif Andersson, and Johan Ruuskanen—you have the gratitude of both me and the readers of this thesis for your efforts.

Finally, I would like to thank my friends and family for all their support. My parents, Jerry and Lillevi, and my sister, Amanda—thank you for all your loving support, and for always believing in me. A special thank you to my grandfather, Gunnar, who keeps me inspired, and with whom I always enjoy discussing the big scientific mysteries. Lastly, but most importantly; thank you Sara and Signe, my own little family—you are the greatest joy of my life.

Financial Support

This research was supported by the Swedish Research Council through the project 2017-04491. The author is a member of the ELLIIT Strategic Research Area at Lund University.

Contents

Nomenclature

Nomenclature	**Description**
$\mathbb{N}_0$	Natural numbers, including zero
$\mathbb{R}$	Real numbers
$\mathbb{R}^n$	Real-valued column vectors of length n
$\mathbb{R}^{n\times m}$	Real-valued matrices of dimension $n \times m$
I_n	Identity matrix of size $n \times n$
$\mathrm{tr}(\cdot)$	Matrix trace
$\det(\cdot)$	Matrix determinant
$(\cdot)^\intercal$	Matrix transpose
$(\cdot)_{ij}$	Matrix element in the i^{th} row and j^{th} column
$\odot$	Hadamard (element-wise) product
$X \succ 0$	Positive definite matrix X
$X \succeq 0$	Positive semidefinite matrix X
$\exp(\cdot)$	Exponential function
$\delta(\cdot)$	Dirac delta function
$\{\cdot\}$	Sequence
$\lvert\cdot\rvert$	Absolute value
$\lVert\cdot\rVert_1$	1-norm
$\lVert\cdot\rVert$	2-norm
$\lVert\cdot\rVert_X$	Weighted 2-norm, with weight X
$\lVert\cdot\rVert_{\mathrm{Frob}}$	Frobenius norm
$x \propto y$	x is proportional to y
$\Pr(\cdot)$	Probability
$p(\cdot)$	Probability density function
$\mathbb{E}[\cdot]$	Expected value
$\mathcal{N}(\mu,\, \Sigma)$	Normal distribution with mean μ and covariance Σ
$\mathcal{A}(\cdot)$	Infinitesimal generator
∇	Gradient
∇^2	Hessian
$\mathcal{C}^n$	Continuous differentiability up to order n
$\mathcal{O}(\cdot)$	Function order

Throughout this thesis, solid and dotted lines in block diagrams signify continuous-time and discrete-time signals respectively.

Abbreviation	Description
ARE	Algebraic Riccati equation
A/D	Analog–to–digital
BTCS	Backward time, centered space
CL	Closed loop
CLMB	Closed loop with measurement buffer
CPU	Central processing unit
DC	Direct current
D/A	Digital–to–analog
FIR	Finite impulse response
FOPDT	First-order process with deadtime
HJB	Hamilton–Jacobi–Bellman
IaR	Integrate–and–reset
i.i.d.	Independent and identically distributed
KBF	Kalman–Bucy filter
KF	Kalman filter
LEQG	Linear–exponential–quadratic–Gaussian
LQG	Linear–quadratic–Gaussian
LQR	Linear–quadratric regulator
LTI	Linear time-invariant
MAE	Maximum absolute error
MMSE	Minimum mean square error
MSE	Mean square error
OL	Open loop
OSP	Optimal sampling problem
PDE	Partial differential equation
pdf	Probability density function
PID	Proportional–integral–derivative
QP	Quadratic program
RBF	Radial basis function
RDD	Row diagonally dominant
SI	Single input
SO	Single output
SSOD	Stochastic send–on–delta
SSODP	Stochastic send–on–delta with a simple prediction
ZOH	Zero-order hold

1

Introduction

Automatic control is the hidden technology that enables many of the comforts we take for granted, such as cruise control and automatic brake systems in our cars, air conditioning, autofocus in our cameras, and automatic adjustment of signal strength in the communication between our smartphones and base stations. The key to automatically regulating dynamical systems in the presence of uncertainty is the concept of *feedback*, illustrated in Figure 1.1. The process we wish to control is then regulated by a controller via the input signal u, based on observations of the output signal y. Modern control systems are almost exclusively implemented using computers, where the controller operates in discrete time. The measured output y is then sampled by a sensor and fed to the controller as a discrete-time signal $\bar{y}$, which then produces the discrete-time instructions $\bar{u}$ for the actuator that produces the control signal u. For such *sampled-data* feedback systems, wireless communication is often used to transfer data between sensor, controller, and actuator.

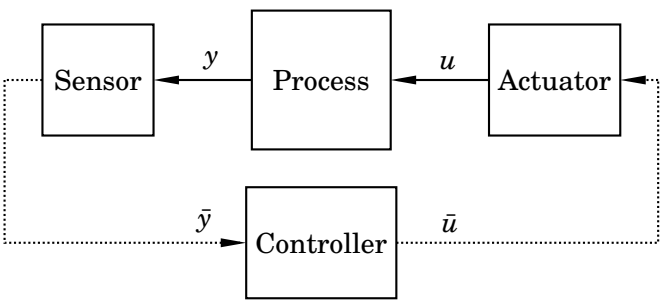

Figure 1.1 Illustration of a sampled-data feedback system. The measured output y of the controlled process is sampled by a sensor, which then feeds the sampled signal $\bar{y}$ to a discrete-time controller. The controller computes instructions $\bar{u}$, which the actuator uses to generate the control signal u.

Traditionally, feedback actions such as sampling, actuation, and data transmissions are triggered periodically based on a set timer—regardless if adjustments to the current control action are required or not. This results in redundant computations and communication, which can cause unnecessary loads on CPU utilization and network bandwidth in the control system. It also needlessly drains batteries in wireless sensors, where communication often constitutes a major part of the total power consumption. To avoid this redundancy, it is natural to instead trigger feedback actions only when a significant change in system behavior is detected—an *event*. This is the concept behind *event-based control*. By introducing decisions on *when* to apply feedback, the event-based approach has the potential for more resource-efficient designs, saving e.g., energy, network bandwidth, and computations in the system. In this thesis, we investigate how the event trigger should be designed to fully exploit this potential.

A great challenge when implementing event-based control systems is the lack of a mature design framework. One promising avenue is the classic linear-quadratic-Gaussian (LQG) framework, which we consider in this thesis. In this framework, the aim is to achieve an optimal trade-off between a quadratic cost on the control objective and the average rate of triggered events. Finding an optimal combination of event trigger and inter-event control actions is generally a very complex task. However, the LQG framework offers a great benefit in the form of a separation principle, which enables us to consider the optimization of the event trigger in isolation. This thesis describes the background on these results, and provide tools for finding the optimal event trigger. Suggestions for near-optimal but simpler event triggers are also given based on findings using these tools. In the same LQG framework, this thesis also proposes simple yet well-performing alternatives for event-based proportional–integral–derivative (PID) control and state estimation.

1.1 Event-Based Control

We begin with a general description of the event-based control concept, followed by a brief overview of the research field's history, and a small, but influential, motivating example. We also give an overview of different structures for event-based control and specify what type of structure we consider in this thesis.

General Concept

Control based on sampled measurements traditionally uses periodic sensing, computations, and actuation, with a constant control signal generated via zero-order hold between each measurement update [Åström and Witten-

mark, 1997]. However, when some of these periodic actions are associated with a cost, it becomes natural to avoid redundant actions and make sure they are only triggered when required. This is the concept of event-based control, where the cost of such actions (i.e., *events*) are taken into account in the design formulation.

Two general aspects to consider in event-based control design are

(i) the policy for event triggering, and

(ii) the policy for generating a control signal between events.

The policy for event triggering (i) can take many forms, ranging from a simple threshold on the measured process output, commonly referred to as *send–on–delta* [Miskowicz, 2006] or a *deadband* [Otanez et al., 2002], to more complex, model-dependent rules. An illustration that compares event-based send–on–delta to periodic triggering is shown in Figure 1.2. In this thesis, we consider only *reactive* event triggering, meaning policies that trigger events based on the behavior of constantly monitored system variables. Policies that schedule events based on predictions are commonly referred to as *self-triggered* [Anta and Tabuada, 2010; Heemels et al., 2012], and are outside the scope of this thesis.

Just as for event triggering, there is a range of choices for the control signal generator (ii)—from a standard zero-order hold [Åström and Wittenmark, 1997, Section 7.3] with stepwise constant control signals, to more complex *generalized hold circuits* based on model information [Kabamba, 1987]. In general, it is a hard task to find an optimal combination of both (i) and (ii), and much research has therefore been focused on only one aspect while keeping the other fixed. However, the problem is separable in the classic design framework of LQG [Goldenshluger and Mirkin, 2017], which promotes its usefulness for event-based control design. Background on the LQG design framework, including the useful separation property, is presented in Chapter 2.

A general notion in event-based control is the trade-off between event rate and the corresponding penalty to the control objective. This trade-off is illustrated in Figure 1.3, where it is assumed that periodic (red) and event-based (blue) triggering is used for two otherwise identical controllers. As the average time between events grows smaller (increasing event rate), the performance of both controllers approaches that of an analog (continuous-time) controller. Since events occur frequently, their exact timing becomes less important, and the performance difference between periodic and event-based triggering is small. However, as events occur more infrequently (decreasing event rate), the performance impact of the exact timing is greater. Consequently, the performance difference between periodic and event-based triggering becomes more pronounced. The illustration in Figure 1.3 shows

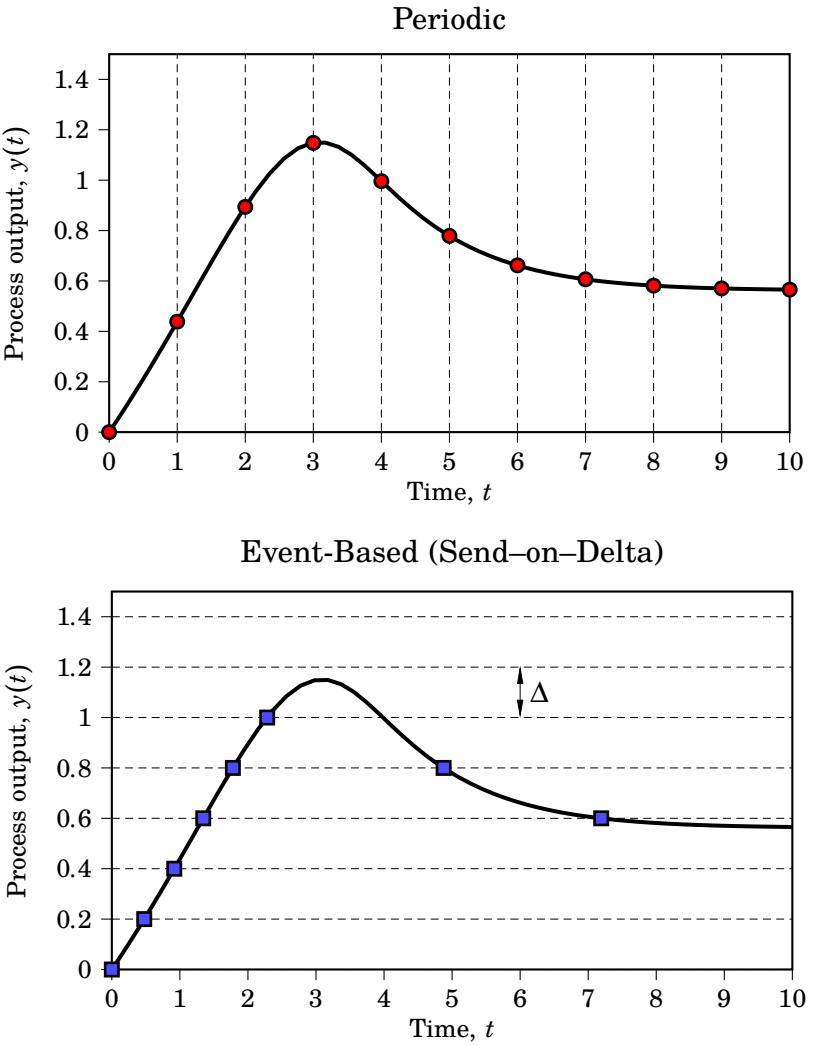

Figure 1.2 Comparison of periodic (upper) and send–on–delta-triggered (lower) sampling. In periodic triggering, each sampling instant (red circles) occur at fixed time intervals, whereas in the send–on–delta case, each sampling instant (blue squares) occur when the process output $y(t)$ differs by Δ from the previous sampled value. Note for the send–on–delta case how sampling instants are triggered more frequently during the initial rise, and less frequently as the process output $y(t)$ settles.

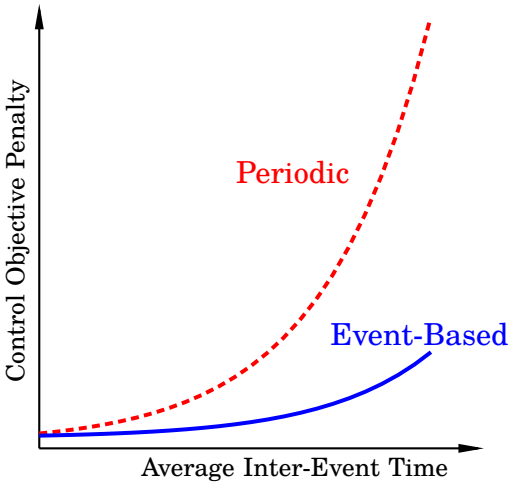

Figure 1.3 Illustration of typical trade-off behavior between average inter-event time and corresponding penalty to the control objective under periodic (red, dashed) and event-based (blue, solid) triggering for two otherwise identical controllers.

an ideal situation, where event-based triggering gives a consistently better trade-off (i.e., a consistently smaller penalty for all inter-event times) than periodic triggering. An event-based controller with this desirable property is commonly referred to as being *consistent* [Antunes and Khashooei, 2016]. Naturally, not every event-based controller is consistent, and poor event-based design can still be outperformed by periodic triggering.

Like all control strategies, event-based control comes with both advantages and disadvantages, and its ultimate usefulness depends on the specific application. Advantages include

- more efficient use of limited or costly system resources, with potential benefits such as reduced wear-and-tear, increased battery lifetime, reduced CPU utilization, less bandwidth allocation, et.c.,
- more intuitive treatment of systems components with discrete-event behavior, such as relays, rotary encoders, queuing networks, et.c.

However, these advantages should be weighed against

- more challenging theoretical analysis and design problem compared to periodic triggering, and
- unpredictable timings of events, resulting in difficult scheduling of shared computational resources.

Historical Overview

Event-based control has a history reaching back to at least the late 1950s, when the idea of sampling new measurement data based on level crossings (i.e., send–on–delta) was proposed for a computer-controlled system by Phillip H. Ellis at the Sperry Gyroscope Company. Ellis then made the following observation:

> The most suitable sampling is by transmission of only significant data, as the new value obtained when the data are changed by a given increment. In certain cases, transmission of data by this means can be used to increase channel capacity.
>
> [Ellis, 1959]

This idea of triggering sampling based on monitored system variables later became popularized in the 1960-70s under the term *adaptive sampling* [Dorf et al., 1962; Hsia, 1972]. During the same period, much work was also focused on developing accelerometers and gyroscopes with event-based pulse feedback [Åström, 2008].

In the 1980s, the concept of *neuromorphic engineering* was developed by Carver Mead's research group at Caltech, after inspiration from biological systems such as neurons. They proposed to integrate biologically inspired electronic sensors with analog circuits, where asynchronous event-based communications protocols would be used. Today, this work has resulted in a range of biologically inspired event-based sensors, such as the *dynamic vision sensor*—a camera inspired by the human eye. In the dynamic vision sensor, each pixel uses event-based triggering to update its values based on incremental changes in light intensity [Liu et al., 2014b].

The current wave of research interest in event-based control began in the late 1990s, motivated by increasing interest in wireless networked control systems and the potential benefit event-based policies can provide in terms of energy, computations, and network bandwidth. This potential was demonstrated in the seminal works [Åström and Bernhardsson, 1999] and [Årzén, 1999], and the research field has since experienced exponential growth in the yearly number of publications [Aranda-Escolástico et al., 2020].

Due to the impact made by the work in [Åström and Bernhardsson, 1999], and its particular relevance for the problems considered in this thesis, we proceed by giving a brief review of one of its main results.

Example 1.1—Periodic versus Event-Based Impulse Feedback

In this example from [Åström and Bernhardsson, 1999], we compare performance between periodic and event-based impulse feedback. The process model is a simple first-order integrator, represented by the following scalar

stochastic differential equation

$$dx = udt + dw.$$

Here, the disturbance w is a Wiener process with unit incremental variance, and u is the control signal. The objective in this example is to keep the state x as close to the origin as possible using control actions in the form of impulses. With x available for feedback, the optimal impulse applied at time τ is

$$u(t) = -x(\tau)\delta(t-\tau),$$

which exactly resets x to the origin. We note that $x(t) = 0$ could be achieved for all t by allowing an infinite impulse rate. However, when the impulse rate is restricted to being finite, there is time between each impulse where x drifts randomly due to the disturbance w. In this case, we face the design problem of *when* each impulse should be triggered.

The standard option is periodic triggering, with an impulse triggered after every h time units. Since the state x evolves according to a Wiener process between each impulse, the average variance of x is given by

$$J_{\text{per}} := \frac{1}{h}\mathbb{E}\Big[\int_0^h x^2(t)dt\Big] = \frac{1}{h}\int_0^h \mathbb{E}\big[w^2(t)\big]dt = \frac{1}{h}\int_0^h tdt = \frac{h}{2}.$$

Now, consider an event-based send–on–delta policy (referred to as *Lebesgue sampling* in [Åström and Bernhardsson, 1999]), where impulses are triggered whenever x satisfies

$$|x(t)| = \Delta. \tag{1.1}$$

A comparison of the time responses for send–on–delta and periodic triggering is shown in Figure 1.4. Let τ denote the first time the state x satisfies (1.1) after starting in the origin. Since the process $t - x^2(t)$ is a Martingale, we have

$$\mathbb{E}[\tau - x^2(\tau)] = 0.$$

This means that the expected time between two impulses is given by

$$\mathbb{E}[\tau] = \mathbb{E}[x^2(\tau)] = \Delta^2.$$

The average variance under the event-based policy can be computed from the stationary probability distribution of x. It is given by

$$p(x) = \begin{cases} \dfrac{\Delta - |x|}{\Delta^2}, & |x| \le \Delta, \\ 0 & \text{otherwise,} \end{cases}$$

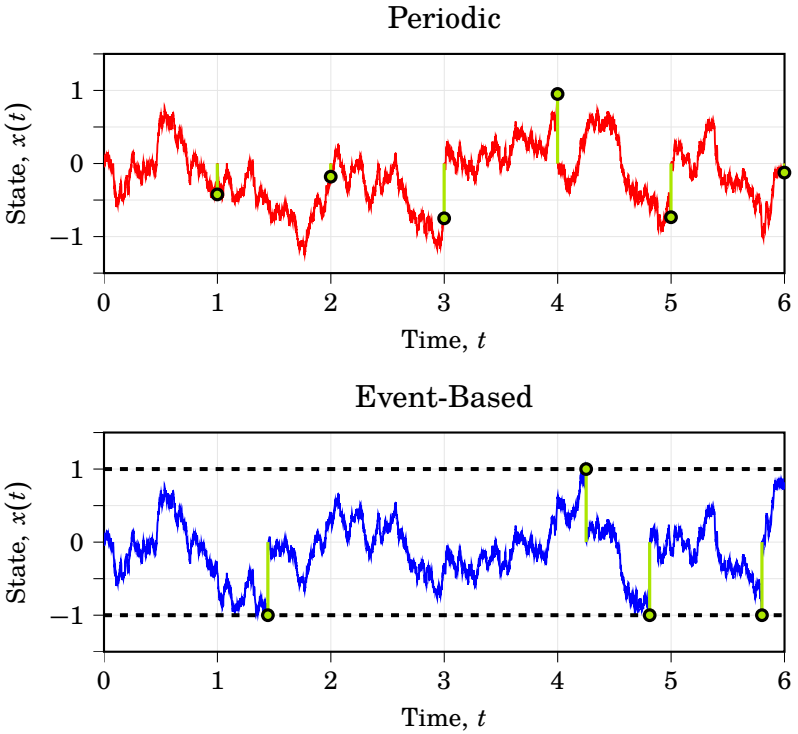

Figure 1.4 Time responses for periodic (upper) and event-based send–on–delta (lower) impulse feedback (green stems) based on the example from [Åström and Bernhardsson, 1999]. In this example, we have $h = \Delta = 1$, corresponding to the same average sampling interval in both cases.

where we refer to [Åström and Bernhardsson, 1999] for its derivation. The average variance under the event-based policy is then computed as

$$J_{\text{eb}} \coloneqq \int_{-\infty}^{\infty} x^2 p(x) dx = \int_{-\Delta}^{\Delta} x^2 \frac{\Delta - |x|}{\Delta^2} dx = 2 \int_{0}^{\Delta} x^2 \frac{\Delta - x}{\Delta^2} dx = \frac{\Delta^2}{6}.$$

Finally, we compare the average variances obtained under periodic and event-based triggering respectively. For the same average impulse rate, with $\mathbb{E}[\tau] = \Delta^2 = h$, the ratio of variances is given by

$$\frac{J_{\text{per}}}{J_{\text{eb}}} = \frac{h}{2} \Big/ \frac{h}{6} = 3.$$

With this, we conclude that event-based triggering always results in three times smaller variance under the same impulse rate—an elegant result that demonstrates the efficiency of event-based control.

While this small example may seem trivial, it turns out that a more general version of it has a fundamental role in LQG-optimal sampled-data control, where the impulses directly translate to sampling actions in a feedback system. This connection is described in Chapter 2. While the send–on–delta trigger in (1.1) is an optimal sampling policy for first-order systems (see [Henningsson et al., 2008]), its counterpart for higher-order systems is mostly unknown. Similarly, it is unknown how the factor three improvement in (1.1) generalizes for higher-order systems. In this thesis, we investigate these unknowns beyond the first-order case—an important step towards a complete design framework for event-based control.

Event-Based Control Structures

The exact meaning of an *event* in event-based control differs based on the considered problem structure. Here follows an overview of some common structures.

Inherent versus By Design. Event-based control problems generally fall into one of the following two broad categories:

- problems with inherently event-based system components, and
- problems that are event-based by design.

The former category includes systems with components such as rotary encoders, relays, pulse width modulators, queuing networks, et.c [Åström, 2008]. In all these components, the internal state changes based on events rather than time. For instance, a rotary encoder triggers pulses only at fixed angular increments, which is an inherent form of send–on–delta.

The latter category is arguably more common in the literature, and the problems considered in this thesis are of this type. Event-based logic is then deliberately introduced in the control system to avoid redundant usage of limited or costly system resources. Different resource limitations, and corresponding event actions, give rise to different design problems.

Design Based on Resource Limitations. As mentioned in the historical overview, a motivating application for event-based strategies is control over data networks, commonly referred to as *networked control systems* [Hespanha et al., 2007; Zhang et al., 2017]. These networks are often shared by many nodes that communicate wirelessly, making network access and battery lifetime limited resources. Common examples of event-based control with multiple feedback loops are

- systems of several independent single loop feedback systems, all sharing a common network bus, see e.g., [Cervin and Henningsson, 2008; Trimpe and D'Andrea, 2014], and

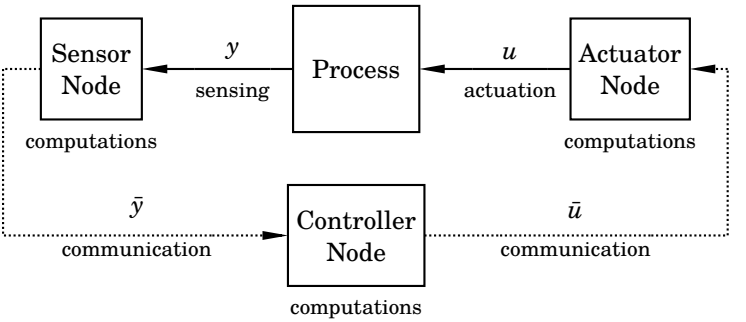

Figure 1.5 Feedback loop illustrating different nodes and associated actions that may be relevant for an event-based design. Adapted from [Henningsson, 2012, Figure 1.1]

- multi-agent networks with a consensus objective, see e.g., [Dimarogonas et al., 2011; Seyboth et al., 2013].

While multiple feedback loops are arguably of more practical interest, the single loop case is useful for theoretical research on event triggers and inter-event control actions, since they are then considered in isolation. For this reason, we focus on the single loop case in this thesis. Ultimately, we expect that insights gained from the single loop case can be useful also in the case of multiple feedback loops.

Many single-loop event-based control problems are described by a feedback loop of the form in Figure 1.5. Similar to Figure 1.1, it illustrates a closed-loop system with a controller distributed over three nodes; a sensor node, a controller node, and an actuator node. The sensor node monitors the output y of the controlled process and communicates a sampled measurement $\bar{y}$ to the controller node. The controller node updates its internal state based on the received measurement and then communicates new instructions $\bar{u}$ to the actuator node, which generates the control signal u. As listed in Figure 1.5, each node and link are associated with some action that may draw from limited resources. These are:

- communication, which may be costly in terms of network bandwidth and/or power consumption from radio transmissions [Hespanha et al., 2007; Zhang et al., 2017; Rault et al., 2014];
- actuation, where changes may be costly due to upsets in production or increased wear-and-tear of equipment [Åström, 2008];
- sensing, for which high-resolution hardware might be expensive while sensors with more coarsely quantized measurements can provide a cheaper option; and

- computations, which are performed on hardware with limited memory and/or processing power, that should possibly also be shared by other tasks.

In this thesis, we focus on reducing communication between the sensor, controller, and actuator nodes. Examples of networked control systems where communication is limited include unmanned aerial vehicles with stealth requirements, vehicles with tight power-budgets such as planetary rovers, long-endurance sensor networks with limited energy supply, and underwater vehicles [Hespanha et al., 2007]. While we do not explicitly consider computational limitations in this thesis, we also consider some suboptimal solutions that trade some performance for simpler, less computationally demanding implementations.

1.2 Related Work

Topics related to the work in this thesis can be broadly categorized into

- stochastic event-based control,
- event-based state estimation, and
- event-based PID control.

For a comprehensive survey on the literature of event-based control over the last twenty years, see [Aranda-Escolástico et al., 2020].

Stochastic Event-Based Control

The stochastic approach to event-based control has been much inspired by the initial work in [Åström and Bernhardsson, 1999], from which we presented an example in Section 1.1. Since then, different research directions have included optimal control, suboptimal control with performance guarantees, and, in later years, methods inspired by machine learning.

Optimal Control. Optimality results in stochastic event-based control have mainly been obtained within the framework of LQG. In [Meng and Chen, 2012], the results in [Åström and Bernhardsson, 1999] are extended to a class of second-order systems, where expressions for the LQG-optimal cost and average event rate are derived under an ellipsoidal threshold policy. The Lebesgue policy in [Åström and Bernhardsson, 1999] is formally shown to be LQG-optimal for first-order systems in [Henningsson et al., 2008], where also a minimum inter-event time is introduced under the term *sporadic* event-based control. A general framework based on path integrals is proposed in [Henningsson, 2011], which is then used in [Henningsson,

2012, Paper II] to extend the results in [Henningsson et al., 2008] for the case of multivariate integrator systems.

In [Molin and Hirche, 2009], a principle of separation is shown to hold between the inter-event control actions and the event trigger in a discrete-time LQG setup with state feedback. This result is extended to the continuous-time case in [Molin and Hirche, 2010a], and the discrete-time case with output feedback in [Molin and Hirche, 2010b]. The same separation is formally shown for the continuous-time output feedback case in [Goldenshluger and Mirkin, 2017], which in turn is based on similar results for the case of $\mathcal{H}_2$-optimization in [Mirkin, 2017]. In [Braksmayer and Mirkin, 2017], the $\mathcal{H}_2$-optimal sampled-data controller from [Mirkin, 2017] is used in a stochastic setting, where some heuristically motivated event-based trigger policies are evaluated.

In this thesis, we also focus on optimal control within the LQG framework, with the results in [Goldenshluger and Mirkin, 2017] being an important foundation. With several strong theoretical results in the literature, we believe that this approach is a good candidate for obtaining a relatively simple design framework for event-based control.

Suboptimal Control with Performance Guarantees. To enable simpler analysis and implementation, there has also been a large focus on suboptimal event-based control. By also providing performance guarantees, a suboptimal solution can also prove very useful.

In [Cogill et al., 2007], a simple ellipsoidal threshold policy is proposed for a discrete-time impulse control problem with quadratic cost, which is shown to give a cost within a factor six of the optimal. The results are extended to the case of zero-order hold actuation in [Cogill, 2009], where an ellipsoidal threshold policy is jointly optimized with a linear feedback control law to minimize an upper bound on the cost.

In [Antunes et al., 2012; Antunes and Heemels, 2014], the authors considers the co-design of event trigger and inter-event control actions for a discounted LQG setup with state feedback, within the framework of *periodic event-triggered control* introduced in [Heemels et al., 2013]. The co-design problem is tackled using a roll-out dynamic programming technique, which guarantees a lower cost compared to periodic triggering with the same average event rate. A similar method is also used in [Gatsis et al., 2014], where the event policy is a smooth probability function for successful data transmissions rather than a deterministic threshold. Performance guarantees are subsequently established in [Gatsis et al., 2016].

The work in [Antunes, 2013] generalizes upon the standard disturbance assumption of white Gaussian noise by considering an LQ setup with random impulse disturbances arriving at a Poisson rate. For this setup, an event-based controller is derived that is guaranteed to outperform its pe-

riodic counterpart. Similar results are obtained in the subsequent work in [Antunes and Khashooei, 2016], where the notion of *consistency* for event-based control is introduced. In [Khashooei et al., 2017], a consistent event-based controller is derived for a discrete-time LQG setup with output feedback.

Machine Learning. A recent trend in stochastic event-based control is the adoption of methods inspired by machine learning. In [Baumann et al., 2018], deep reinforcement learning is used to simultaneously learn policies for event triggering and inter-event control actions for nonlinear discrete-time systems subject to additive Gaussian noise. The policies are obtained via a value function parameterized by a deep neural network, whose parameters are learned from data.

In [Solowjow et al., 2018], the concept of *event-triggered learning* is proposed. Based on an LQG setup, an identification experiment is triggered to update model parameters whenever the measured event rate differs significantly from the one expected from the model. The exact value for the expected event rate is given by the solution to an elliptic partial differential equation, which becomes intractable to solve online for higher-order systems. Instead, the authors estimate the expected event rate online using Monte Carlo simulations. Event-triggered learning is also applied for a continuous-time linear impulse control problem in [Baumann et al., 2019]. The considered setup shares similarities to those considered in [Åström and Bernhardsson, 1999; Henningsson et al., 2008; Henningsson, 2011; Henningsson, 2012], but differs in that impulse magnitudes are model-dependent.

Event-Based State Estimation

Remote state estimation is another common problem in the event-based literature. In this problem class, events correspond to decisions in a sensor for transmitting measurements to a remote state estimator. Typically, the remote estimator is aware of the policy used for event triggering in the sensor, which can be exploited to gain implicit measurement information even in the absence of transmissions. While the inclusion of implicit measurement information can result in better estimation performance, it also significantly raises the complexity of the design of the remote estimator. Consequently, there is a wide range of proposals in the literature for coping with this added complexity.

In [Li et al., 2010], an LQG setup is considered, where the sensor includes a Kalman filter and a local copy of the remote estimator. The event trigger is based on the error between the optimal estimates obtained from the Kalman filter and those of the remote estimator (available via the local copy), with the Kalman filter estimates being transmitted at events. The

remote estimator employs an open-loop prediction in the absence of events but does not take any implicit information into account. Optimization of the event trigger is considered over a finite horizon based on a quadratic over-approximation of the value function and shows similarities to the problem considered in [Cogill et al., 2007].

A similar setup with a local estimator at the sensor is considered in [Trimpe and D'Andrea, 2014]. In this work, several sensors are sharing a common network bus, each having a local copy of the remote estimator. Global information is ensured in the setup by the assumption that every sensor node listens to all transmissions over the network bus. The trigger function in each sensor is based on the variance of its measurement prediction, but no measurement updates are taken by the estimators in the absence of events.

A framework for event-based state estimation in an LQG setting under general trigger conditions is proposed in [Sijs and Lazar, 2012]. In this framework, the state estimates are updated with implicit information in the absence of events by using a sum of Gaussians approximation for the likelihood function. With this approximation, the estimator becomes available on a recursive form, similar to the standard Kalman filter.

Particle filtering has also been considered as an alternative for approximating the nonlinear measurement update in the absence of events. In [Sid and Chitraganti, 2016], an event-based bootstrap particle filter is proposed, with a likelihood function obtained via numerical integration. Another bootstrap implementation is considered in [Davar and Mohammadi, 2017], with an estimator tailored for an LQG setting with send–on–delta event triggering. Improved performance over the bootstrap implementations is noted in [Ruuskanen and Cervin, 2019], where an event-based auxiliary particle filter is derived using the same likelihood approximation as in [Sijs and Lazar, 2012].

A method of stochastic triggering for the LQG setting was proposed in [Han et al., 2015], with extended analysis in [Shi et al., 2016]. With a suitable choice of trigger probability, it is shown that the Gaussian distribution of the process state is preserved also in the absence of events, resulting in a Kalman filter-like estimator with event-dependent measurement update. Improved estimator performance is noted in [Schmitt et al., 2019], where stochastic triggering is combined with a measurement prediction based on a finite impulse response filter in the sensor.

Event-Based PID Control

Research on event-based PID control started with the seminal work in [Årzén, 1999], where a simple event-based extension of a standard discrete-time PID implementation was proposed. In this work, the potential for

computational savings is demonstrated through simulations on a double tank process, with the estimated CPU utilization of the event-based PID being less than half of that of its periodic counterpart.

In [Vasyutynskyy and Kabitzsch, 2006], the problem of *sticking* is observed for Årzén's implementation. This phenomenon is caused by too small deviations in the control error at the peak of overshoots, resulting in the absence of new events and getting temporarily "stuck" in an equilibrium away from the setpoint until a time-out is reached. A simple sticking detection algorithm is proposed to alleviate this problem. Another scheme for canceling sticking is introduced in [Tiberi et al., 2012]. In this work, the event trigger is based on the PI part of the controller, with the latter being implemented on the automatic reset form used in the PIDPLUS commercial controller [Song et al., 2006]. The inclusion of the integral part results in an implicit time-out since it continues to grow and eventually triggers an event also for stationary errors.

Improvements to the integral action of Årzén's implementation are proposed in [Durand and Marchand, 2009a]. To reduce overshoots in setpoint changes, the authors propose the removal of the time-out and limiting the growth of the integral action based on three alternatives; saturation of the integral approximation, adding an exponential forgetting factor, and a hybrid of the two. Further improvements are noted in [Durand and Marchand, 2009b], where an asynchronous event trigger is considered.

To further improve transient behavior, the inclusion of model information has also been considered. In [Vasyutynskyy and Kabitzsch, 2009], a first-order observer is introduced in the event trigger. The observer is designed based on a first-order approximation of a possibly higher-order linear model. The work in [Sánchez et al., 2011] considers the design of an event-based PI controller with two degrees of freedom, where the feedforward controller is based on a first-order process model with deadtime (FOPDT). The feedforward controller is active during setpoint changes, where a self-triggered policy based on the FOPDT model is used to reach the setpoint.

1.3 Contributions

Central to this thesis is a stochastic control problem concerned with designing an LQG-optimal event trigger, formulated in Chapter 3 as the *optimal sampling problem* (OSP). Based on the OSP, the main contributions of this thesis are:

- Analytic results, in the form of exact solutions to the OSP for two new system classes and tight bounds on optimal performance compared to periodic triggering for multivariate integrator systems. These contributions are presented in Chapter 3.

- Numerical results, in the form of two methods for solving the OSP approximately; a simulation-based method and a method based on linear complementarity. These methods are presented in Chapter 4.
- New insights regarding the optimal event trigger based on numerical studies on second-order systems. Findings suggest that ellipsoidal threshold policies offer a simple yet well-performing alternative to the optimal trigger policy, with optimal parameters obtainable also for higher-order systems via a Monte Carlo approach. These contributions are presented in Chapter 4.

Additionally, the following contributions are made for event-based PID control and state estimation:

- New proposals for implementation of event-based PID control based on results in LQG-optimal sampled-data control, offering improved trade-offs between performance in terms of quadratic cost and event rate. These proposals are presented in Chapter 5.
- Two new simple yet well-performing stochastic trigger policies for event-based state estimation, presented in Chapter 6.

1.4 Publications

The work presented in this thesis is based on the following publications:

Thelander Andrén, M., B. Bernhardsson, A. Cervin, and K. Soltesz (2017). "On event-based sampling for LQG-optimal control". In: *Proceedings of the 56th IEEE Conference on Decision and Control*. Melbourne, Australia, pp. 5438–5444.

This article considers sampled-data LQG control and introduces the OSP covered in Chapter 3. It also proposes a simulation-based numerical method for solving the optimization problem, presented in Chapter 4.

M. Thelander Andrén made the initial observation that previous results on impulse control could be used as a framework for solving the OSP. B. Bernhardsson contributed with the core idea of the simulation-based numerical method, while M. Thelander Andrén was responsible for its implementation and related simulation results. The performance bounds for multivariate integrator systems were derived jointly by M. Thelander Andrén and K. Soltesz. Most of the manuscript was written by M. Thelander Andrén, with A. Cervin contributing ideas on its structure. All authors contributed to the editing of the manuscript.

Thelander Andrén, M. (2019). "Using radial basis functions to approximate the LQG-optimal event-based sampling policy". In: *Proceedings of the 18th European Control Conference*. Naples, Italy, pp. 2832–2838.

This article proposes an improved numerical method based on linear complementarity for solving the OSP. The method is presented in Chapter 4 and approximates the value function with a basis expansion, for which weights are found by solving a quadratic program.

Thelander Andrén, M. and A. Cervin (2020). "Optimal event-based sampling in LQG control: Exact solutions and numerical methods". *IEEE Transactions on Automatic Control*. Submitted.

This article serves as a foundation for Chapter 3, and summarizes results on the OSP for sampled-data LQG control. Additionally, it discusses the properties of the value function, introduces exact solutions for two special cases, and compares the simulation-based and linear complementarity methods of the previous articles in a numerical study. M. Thelander Andrén was the main contributor to this work, with editing suggestions provided by A. Cervin.

Cervin, A. and M. Thelander Andrén (2020). "LQG-optimal versus simple event-based PID controllers". In: *Proceedings of the 2020 American Control Conference*. Denver, CO, USA, pp. 3678–3684.

This article serves as a foundation for Chapter 5, and considers a continuous-time LQG benchmark problem for which the solution is an ideal PID controller. Based on previous results, the optimal sampled-data controller and sampling policy can then be expressed in terms of PID control. With the optimal solution as a starting point, some proposals for suboptimal but simpler event-based PID controllers are discussed and evaluated.

The core idea of applying the results on sampled-data LQG on the benchmark problem was by A. Cervin, while the interpretation of the optimal sampled-data controller was by M. Thelander Andrén. A. Cervin performed simulations of the sub-optimal controllers, while M. Thelander Andrén computed the optimal sampling policies and performed the simulations for the optimal controller. Both authors contributed equally to writing the article.

Thelander Andrén, M. and A. Cervin (2016). "Event-based state-estimation using an improved stochastic send–on–delta scheme". In: *Proceedings of the 2nd International Conference on Event-based Control, Communication and Signal Processing*. Kraków, Poland, pp. 1–8.

This article serves as a foundation for Chapter 6, and considers the design of an event-based state estimator using a stochastic transmission policy. The stochastic policy enables an analytic derivation of the minimum mean square error (MMSE) estimator, which has a recursive formula similar to that of the standard Kalman filter. This work improves upon related work by adding a simple predictor in the sensor that does not require any feedback from the estimator, resulting in improved estimation. M. Thelander Andrén was the main contributor to this work, with editing provided by A. Cervin.

Thelander Andrén, M. (2020). "Remote state estimation using stochastic triggering with buffered measurements". Manuscript in preparation.

This unpublished work is presented in Chapter 6, and considers the same event-based state estimation problem as in the previous publication. Here, a new stochastic transmission policy is proposed for the case of bidirectional communication between the estimator and the sensor. In addition to showing improved estimation performance compared to other policies in related works, we also derive exact expressions for the average transmission rate and mean-square error under the proposed policy.

The author has also contributed to the following publications that are not included in the thesis:

Hübel, M., S. Meinke, M. Thelander Andrén, C. Wedding, J. Nocke, C. Gierow, E. Hassel, and J. Funkquist (2017). "Modelling and simulation of a coal-fired power plant for start-up optimisation". *Applied Energy* **208**, pp. 319–331.

Nylander, T., M. Thelander Andrén, K.-E. Årzén, and M. Maggio (2018). "Cloud application predictability through integrated load-balancing and service time control". In: *Proceedings of the 2018 IEEE International Conference on Autonomic Computing*. Receiver of the Best Paper Award. Trento, Italy, pp. 51–60.

1.5 Outline

This thesis is divided into seven chapters. In Chapter 2, we present relevant background on sampled-data LQG control, including a review of the optimal controller structure. For this controller structure, we then pose the problem of finding the corresponding LQG-optimal sampling policy in Chapter 3, along with exact solutions to problems with certain structures. For the general problem, we propose two numerical methods in Chapter 4, which are validated and compared against a standard value iteration method. In Chapter 4, we also discuss ellipsoidal threshold policies, and how optimal ellipse parameters can be obtained via a Monte Carlo approach. The results

on LQG-optimal sampled-data control are then used as a starting point for the design of event-based PID control in Chapter 5. In Chapter 6, we consider a probabilistic type of event-based trigger policy, known as stochastic triggering, for a prototypical remote estimation problem. Finally, a summary and proposals for future research directions are given in Chapter 7.

2

Sampled-Data LQG Control

This chapter presents relevant background on sampled-data linear–quadratic–Gaussian (LQG) control. We first review the analog (continuous-time) case, where no sampled-data structure is imposed on the controller. The optimal analog controller acts as a performance baseline for all sampled-data implementations. We then proceed by imposing a sampled-data structure on the controller, consisting of an analog–to–digital (A/D) converter (sampler), a microprocessor (discrete-time controller), and a digital–to–analog (D/A) converter (hold).

Sampled-data control design is typically done under assumptions of periodic sampling and standard choices for the sampler and hold, leaving only the discrete-time controller free in the design. While these assumptions limit the attainable performance, they simplify the problem since the design can then be considered entirely in discrete-time. As a reference for future performance comparisons, the LQG-optimal controller for such a standard setup is also presented in this chapter.

The chapter is concluded by reviewing results for the optimal co-design of the sampler, hold, discrete-time controller, and sampling policy. By considering the complete co-design, performance can be pushed closer to the analog case. A key result is that the sampler, hold and discrete-time controller can be optimized separately from the sampling policy. This enables us to consider the optimization of the sampling policy in isolation, which is the main topic of this thesis.

2.1 Plant Model and Objective

Throughout this thesis, we consider the design of feedback controllers for a continuous-time generalized plant G of the form shown in Figure 2.1. The plant outputs are the measured signal y, used for feedback, and the regulated output z, used to measure control performance. The inputs to the plant are the control signal u and the disturbance signal w in the form of

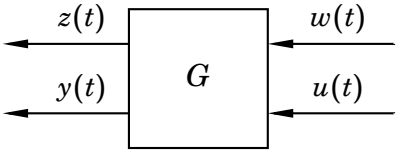

Figure 2.1 Generalized plant G, with the control signal u and disturbance w as inputs, and regulated output z and measured signal y as outputs.

a Gaussian white process with unit intensity. The purpose of the stochastic signal w is to act as a source of uncertainty in the model, corresponding to, e.g., unknown external disturbances and model errors.

The LQG design framework assumes a linear plant model, and we specifically consider a linear, time-invariant (LTI) plant G with a state-space realization of the form

$$G : \begin{cases} \dot{x}(t) = Ax(t) + B_w w(t) + B_u u(t), \\ z(t) = C_z x(t) + D_{zu} u(t), \\ y(t) = C_y x(t) + D_{yw} w(t), \end{cases} \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the plant state vector and the coefficients in (2.1) are matrices of appropriate dimensions. The matrices C_z and D_{zu} are design parameters that add different weights to x and u in the regulated output z. The other matrices in (2.1) are part of the plant model and are considered fixed in the design.

To ensure a well-posed design problem, we assume;

A1: (A, B_u) is stabilizable, and (C_y, A) is detectable.

A2: D_{zu} and D_{yw} have full column and row rank respectively.

A3: The matrices

$$\begin{bmatrix} A - i\omega I_n & B_u \\ C_z & D_{zu} \end{bmatrix}, \quad \begin{bmatrix} A - i\omega I_n & B_w \\ C_y & D_{yw} \end{bmatrix}$$

have full column and row rank respectively, for all ω.

These assumptions are standard in LQG and $\mathcal{H}_2$ literature, see, e.g., [Chen and Francis, 1994, Chapter 2]. **A1** is required for existence of an internally stabilizing controller, while **A2** ensures that all components of u are weighted in z and that all components of y are corrupted by w respectively. **A3** ensures there are no zeros on the imaginary axis in the cross-systems from u to z and from w to y.

Now consider Figure 2.2, where the feedback loop from y to u has been closed with a controller K. For this feedback system, the LQG design

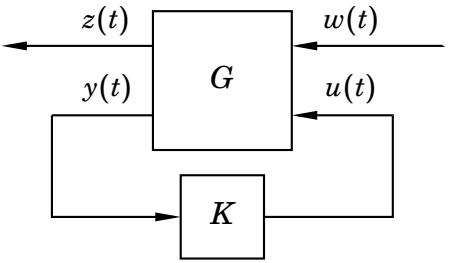

Figure 2.2 Feedback system with controller K and generalized plant G.

objective is to find an internally stabilizing controller K that minimizes the cost

$$J \coloneqq \lim_{T\to\infty} \mathbb{E}\Big[\frac{1}{T}\int_0^T \|z(t)\|^2 dt\Big], \tag{2.2}$$

i.e., the asymptotic mean variance of z.

2.2 Analog LQG Control

We first consider the case when the controller K in Figure 2.2 is analog (i.e., operating entirely in continuous-time). The key to solving the analog LQG design problem is the *separation principle*. It states that the design problem can be separated into optimization of state feedback and state estimation respectively, and it holds due to the whiteness of w and the LTI dynamics of G [Potter, 1964]. The two optimization problems are dual to each other, where both have an LTI solution obtained by solving an algebraic Riccati equation (ARE).

Under **A2**, the optimal state estimator is an LTI system of the form

$$\dot{\hat{x}}(t) = A\hat{x}(t) + B_u u(t) - L\big(y(t) - C_y\hat{x}(t)\big), \tag{2.3}$$

where $\hat{x} \in \mathbb{R}^n$ is the state estimate. The estimator gain L is given by

$$L = -(YC_y^\mathsf{T} + B_w D_{yw}^\mathsf{T})(D_{yw}D_{yw}^\mathsf{T})^{-1}, \tag{2.4}$$

where Y solves the ARE

$$AY + YA^\mathsf{T} + B_w B_w^\mathsf{T} - R = 0, \quad R \coloneqq LD_{yw}D_{yw}^\mathsf{T}L^\mathsf{T}. \tag{2.5}$$

Under **A1** and **A3**, $Y \succeq 0$ is unique, and the optimal state estimator has asymptotically stable estimation error dynamics [Zhou et al., 1996, Chapters 13–14]. In the context of stochastic control, the optimal state estimator (2.3)

is the minimum mean square error (MMSE) estimator, generally known as the Kalman–Bucy filter (KBF) [Åström, 1970].

Under **A2**, the optimal state feedback is a linear static gain of the form

$$u(t) = Fx(t). \tag{2.6}$$

The feedback gain F is given by

$$F = -(D_{zu}^{\mathsf{T}} D_{zu})^{-1}(B_u^{\mathsf{T}} X + D_{zu}^{\mathsf{T}} C_z). \tag{2.7}$$

where X solves the ARE

$$A^{\mathsf{T}} X + XA + C_z^{\mathsf{T}} C_z - Q = 0, \quad Q \coloneqq F^{\mathsf{T}} D_{zu}^{\mathsf{T}} D_{zu} F. \tag{2.8}$$

Under **A1** and **A3**, $X \succeq 0$ is unique, and the optimal state feedback gives an asymptotically stable closed-loop system [Zhou et al., 1996, Chapters 13–14]. The optimal state feedback (2.6) is known as the linear–quadratic regulator (LQR).

By combining the KBF with the LQR we get the optimal controller

$$K : \begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + B_u u(t) - L\big(y(t) - C_y \hat{x}(t)\big), \\ u(t) = F\hat{x}(t), \end{cases} \tag{2.9}$$

which we refer to as the *analog LQG controller*. The minimum cost achieved by (2.9) is given by [Zhou et al., 1996, Theorem 14.7]

$$\min_K J \coloneqq \gamma_0 = \operatorname{tr}(B_w^{\mathsf{T}} X B_w) + \operatorname{tr}(C_z Y C_z^{\mathsf{T}}) + 2\operatorname{tr}(XAY). \tag{2.10}$$

The cost γ_0 is a fundamental performance bound for all controllers—no sampled-data design can achieve a lower cost. Intuitively, this holds since imposing any sampled-data structure on K corresponds to adding constraints to the optimization problem, which can only result in an equal or higher optimal cost. Therefore, γ_0 is a useful baseline when determining how much performance impact a given sampled-data structure has.

2.3 Sampled-Data Control

Now, consider the case when the controller K has the sampled-data structure in Figure 2.3. The controller now consists of a (generalized) sampler S, a discrete-time controller $\bar{K}$ and a (generalized) hold H. This structure can be interpreted as a mathematical abstraction of a digital controller implementation, where S and H corresponds to A/D and D/A converters respectively, and $\bar{K}$ represents a microprocessor [Chen and Francis, 1994, Chapter 1].

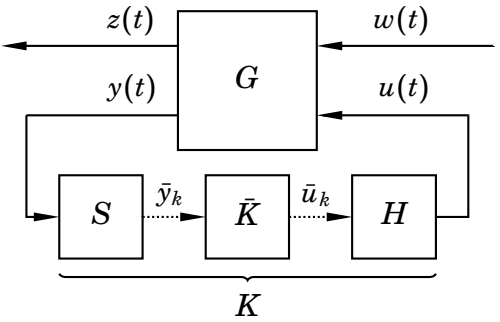

Figure 2.3 Sampled-data feedback system with continuous-time generalized plant G and a sampled-data controller K consisting of a sampler S, a discrete-time controller $\bar{K}$ and a hold H.

Let $\{\tau_k\}_{k\in\mathbb{N}_0}$ denote a sequence of sampling times where $h_k \coloneqq \tau_k - \tau_{k-1}$ are the corresponding sampling intervals. The sampler $S : y \mapsto \bar{y}$ is an operator converting the continuous-time measurement signal y into a discrete-time signal $\bar{y}$ at sampling times according to

$$S: \quad \bar{y}_k \coloneqq \int_0^{\tau_k} \psi(\tau_k, v) y(v) dv.$$

The sampler is characterized by its *sampling function* $\psi(t, v)$, for which it holds that $\psi(t, v) = 0$ when $v > t$ [Braksmayer and Mirkin, 2017]. The sampler transmits $\bar{y}$ to the controller $\bar{K} : \bar{y} \mapsto \bar{u}$, which computes the discrete-time control signal $\bar{u}$ and transmits it to the hold $H : \bar{u} \mapsto u$, which generates a continuous-time control signal u based on $\bar{u}$. This is done according to

$$H: \quad u(t) = \sum_{k=0}^{k_t} \phi(t, \tau_k) \bar{u}_k,$$

where $k_t \coloneqq \max_{\tau_k \le t} k$. The hold is characterized by its *hold function* $\phi(t, v)$, with the property $\phi(t, v) = 0$ when $v > t$ [Braksmayer and Mirkin, 2017].

We consider the following assumptions for the sampled-data controller:

A4: Each component performs its operations instantly, i.e. computational time is assumed to be negligible.

A5: Both $\bar{y}$ and $\bar{u}$ are updated and transmitted at sampling times.

A6: There is no quantization of $\bar{y}$ and $\bar{u}$.

These are common assumptions in the sampled-data literature used to simplify the mathematical analysis [Chen and Francis, 1994, Chapter 1].

From the presented setup, we see that there are three main aspects to consider in the sampled-data design:

(i) the inter-sample behavior, defined by S and H,

(ii) the controller update at sampling times, defined by $\bar{K}$,

(iii) the policy that generates the sequence of sampling times $\{\tau_k\}$.

Typically in control literature, standard choices for (i) (ideal sampling and zero-order hold) and (iii) (periodic sampling) that simplify the analysis are assumed, leaving only $\bar{K}$ available in the design [Åström and Wittenmark, 1997]. However, by also including (i) and (iii) in the optimization we can obtain a controller that achieves similar performance to the standard case for considerably lower sampling rates. This is appealing for applications where sampling is associated with allocation of system resources, such as networked control systems where the different components of the sampled-data structure are not necessarily collocated in a single node [Heemels et al., 2012; Liu et al., 2014a]. Instead, A/D and D/A converters are typically collocated with sensors and actuators respectively, while the microprocessor receives and transmits data from a remote location. In these instances, sampling inherently comes with a cost due to, e.g., allocation of network bandwidth, computational resources, and energy usage in wireless data transmission.

2.4 A Baseline Sampled-Data LQG Controller

To provide a baseline for future designs, we present a standard sampled-data setup and the corresponding LQG-optimal controller. In the setup, we assume that

- S is an ideal sampler combined with a pre-filter,
- H is a zero-order hold circuit, and
- a periodic sampling policy, with constant intervals $h_k \coloneqq h$ for all k.

Ideal Sampling and Pre-filtering

The ideal sampler is an operator which extracts the exact value of a continuous-time signal at sampling times. Consequently, its sampling function is represented by a Dirac delta function, i.e., $\psi(t, v) = \delta(t-v)$. However, the presence of continuous-time white noise in y prohibits us from applying the ideal sampler directly. The reason is that continuous-time white noise is a generalized process with infinite variance, and thus can not be evaluated point-wise in a meaningful way. In contrast, its integral (a Wiener

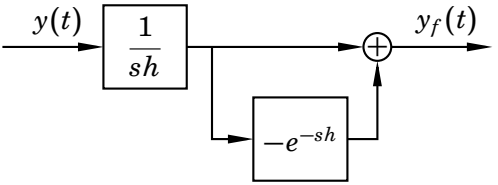

Figure 2.4 Block diagram representation of the integrate–and–reset (IaR) filter.

process) is well-defined for point-wise evaluation, and for this reason we need to integrate (i.e., filter) y before sampling. Naturally, this is not only a theoretical issue—an analog pre-filter (i.e., an anti-aliasing filter) should always be featured in practical implementations to reduce the impact of measurement noise and prevent aliasing [Åström and Wittenmark, 1997].

Common options for pre-filters include Butterworth and Bessel filters of varying orders [Åström and Wittenmark, 1997], and the integrate–and–reset (IaR) filter (also known as an averaging filter) [Goodwin et al., 2010]. Owing to its simplicity, we use the IaR filter in the baseline setup.

The IaR filter produces a filtered version y_f of y by averaging it over each sampling period according to

$$y_f(t) = \frac{1}{h}\int_{t-h}^{t} y(v)dv.$$

A block diagram representation of the IaR filter is shown in Figure 2.4. Applying the ideal sampler to the filtered signal y_f gives

$$\bar{y}_k = \int_0^{\tau_k} \delta(\tau_k - v) y_f(v)dv = y_f(\tau_k). \tag{2.11}$$

Zero-Order Hold

The zero-order hold circuit generates a continuous-time control signal by outputting a constant over each interval. Its hold function is given by an indicator function, $\phi(t, \tau_k) = \mathbf{1}_{[0,h_{k+1}]}(t - \tau_k)$, resulting in

$$u(t) = \sum_{k=0}^{k_t} \mathbf{1}_{[0,h_{k+1}]}(t - \tau_k)\bar{u}_k = \bar{u}_{k_t}.$$

Equivalent Discrete-Time Formulation

With the assumption of zero-order hold, it is possible to derive a discrete-time equivalent of the continuous-time plant that captures system signals exactly at sampling times. This is known as a *stroboscopic model* [Åström

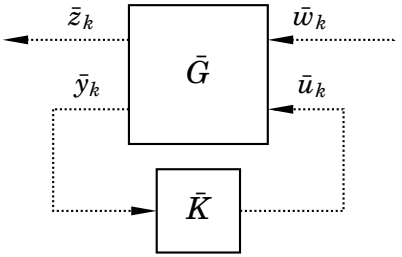

Figure 2.5 Under assumptions of ideal sampling and zero-order hold, the sampled-data feedback system in Figure 2.3 can be equivalently represented by a discrete-time feedback system as above, with a discrete-time controller $\bar{K}$ and generalized discrete-time plant $\bar{G}$.

and Wittenmark, 1997]. The design of the controller $\bar{K}$ can then be considered for a discrete-time version $\bar{G}$ of the plant G including the pre-filter, see Figure 2.5. The resulting discrete-time generalized plant can be written on the following form

$$\bar{G}: \begin{cases} \bar{x}_{k+1} = A_{\bar{x}}\bar{x}_k + B_{\bar{w}}\bar{w}_k + B_{\bar{u}}\bar{u}_k, \\ \bar{z}_k = C_{\bar{z}}\bar{x}_k + D_{\bar{z}\bar{u}}\bar{u}_k, \\ \bar{y}_k = C_{\bar{y}}\bar{x}_k, \end{cases} \tag{2.12}$$

where $\bar{x} \coloneqq \begin{bmatrix} x^\mathsf{T} & y_f^\mathsf{T} \end{bmatrix}^\mathsf{T}$ is the state vector of the extended plant, consisting of the plant state x and the IaR filter output y_f, and $\{\bar{w}_k\}_{k\in\mathbb{N}_0}$ is an i.i.d. sequence of vector-valued standard normal random variables. Note that there is no measurement noise acting on $\bar{y}$ in (2.12). Instead, the measurement noise of the continuous-time model appears as process noise acting on the state y_f. The relations between the given parameters for G in (2.1) and the corresponding ones for $\bar{G}$ in (2.12) are given in Appendix A. The cost (2.2) has an equivalent discrete-time representation based on the regulated output $\bar{z}$, given by [Chen and Francis, 1994, Chapter 12]

$$J = \gamma_w + \lim_{N\to\infty} \mathbb{E}\left[\frac{1}{Nh}\sum_{k=0}^{N-1} \left\|\bar{z}[k]\right\|^2\right], \tag{2.13}$$

where γ_w (see (A.6)) is the accumulated cost over each period due to the disturbance w. With (2.12) and (2.13), the original sampled-data LQG design problem for the baseline setup is reformulated as a discrete-time one.

The Baseline LQG Controller

Similar to the analog case, the discrete-time LQG design problem is also solved via the separation principle. The optimal controller will differ depending on if measurements up until $\bar{y}_k$ (the *filter case*) or $\bar{y}_{k-1}$ (the *predictor*

case) are available when computing the control signal $\bar{u}_k$. Here we opt for the filter case, as it results in the smallest LQG cost of the two cases. The optimal controller is then given by [Åström and Wittenmark, 1997, Chapter 11]

$$\bar{K}: \begin{cases} \hat{x}_{k+1} = A_{\bar{x}}\hat{x}_k + B_{\bar{u}}\bar{u}_k - L\big(\bar{y}_k - C_{\bar{y}}\hat{x}_k\big), \\ \bar{u}_k = \big(F - MC_{\bar{y}}\big)\hat{x}_k + M\bar{y}_k, \end{cases} \tag{2.14}$$

where F, L and M are given by

$$\begin{aligned} F &= -\big(B_{\bar{u}}^\intercal X B_{\bar{u}} + D_{\bar{z}\bar{u}}^\intercal D_{\bar{z}\bar{u}}\big)^{-1}\big(B_{\bar{u}}^\intercal X A_{\bar{x}} + D_{\bar{z}\bar{u}}^\intercal C_{\bar{z}}\big), \\ L &= -A_{\bar{x}} Y C_{\bar{y}}^\intercal \big(C_{\bar{y}} Y C_{\bar{y}}^\intercal\big)^{-1}, \\ M &= FYC_{\bar{y}}^\intercal\big(C_{\bar{y}} Y C_{\bar{y}}^\intercal\big)^{-1}, \end{aligned}$$

and X and Y solve the discrete AREs

$$\begin{aligned} X &= A_{\bar{x}}^\intercal X A_{\bar{x}} + C_{\bar{z}}^\intercal C_{\bar{z}} - F^\intercal (B_{\bar{u}}^\intercal X B_{\bar{u}} + D_{\bar{z}\bar{u}}^\intercal D_{\bar{z}\bar{u}})F, \\ Y &= A_{\bar{x}} Y A_{\bar{x}}^\intercal + B_{\bar{w}} B_{\bar{w}}^\intercal - LC_{\bar{y}} Y C_{\bar{y}}^\intercal L^\intercal. \end{aligned}$$

We refer to the baseline sampled-data setup with the corresponding optimal controller $\bar{K}$ given by (2.14) as the *baseline LQG controller*. The minimum cost achieved by this controller is given by [Åström and Wittenmark, 1997, Chapter 11]

$$\begin{aligned} \min_{\bar{K}} J = \gamma_w &+ \frac{1}{h}\mathrm{tr}\big(B_{\bar{w}}^\intercal X B_{\bar{w}}\big) \\ &+ \frac{1}{h}\mathrm{tr}\Big(\big(FYF^\intercal - MC_{\bar{y}} Y C_{\bar{y}}^\intercal M^\intercal\big)\big(B_{\bar{u}}^\intercal X B_{\bar{u}} + D_{\bar{z}\bar{u}}^\intercal D_{\bar{z}\bar{u}}\big)\Big). \end{aligned} \tag{2.15}$$

2.5 Optimal Sampled-Data LQG Control

In the previous section we presented the LQG-optimal sampled-data controller based on standard choices of the sampler S and hold H in Figure 2.3. We now consider the case when also S and H are available in the design.

Derivation

The optimal sampled-data controller was first derived in terms of $\mathcal{H}_2$-optimization in [Mirkin, 2017], and later extended to the stochastic case with event-based sampling in [Goldenshluger and Mirkin, 2017]. Here, we give a brief outline of the derivation, closely following [Goldenshluger and Mirkin, 2017].

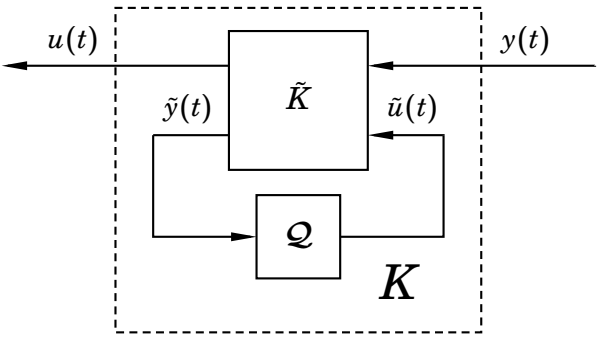

Figure 2.6 The $\mathcal{Q}$-parametrization (2.16) of the controller K.

The first step is to characterize the set of all (including nonlinear and time-varying) internally stabilizing controllers for the plant G. This is done through the $\mathcal{Q}$-parametrization [Goldenshluger and Mirkin, 2017, (12)]

$$K = \tilde{K}_{uy} + \tilde{K}_{u\tilde{u}}\mathcal{Q}(I - \tilde{K}_{\tilde{y}\tilde{u}}\mathcal{Q})^{-1}\tilde{K}_{\tilde{y}y}, \tag{2.16}$$

where $\mathcal{Q} : \tilde{y} \mapsto \tilde{u}$ is any causal stable system (including nonlinar and time-varying) and $\tilde{K}_{ij}$ denote the subsystems from input j to output i of the generalized LTI system

$$\tilde{K} : \begin{cases} \dot{\hat{x}}(t) = \big(A + B_u F + LC_y\big)\hat{x}(t) - Ly(t) + B_u\tilde{u}(t), \\ u(t) = F\hat{x}(t) + \tilde{u}(t), \\ \tilde{y}(t) = -C_y\hat{x}(t) + y(t). \end{cases} \tag{2.17}$$

A block diagram illustrating the $\mathcal{Q}$-parametrization is shown in Figure 2.6. The $\mathcal{Q}$-parametrization given by (2.16) and (2.17) is valid for any choice of F and L that renders the matrices $A + B_u F$ and $A + LC_y$ Hurwitz. However, when F and L are chosen as the LQR and KBF gains (2.7) and (2.4) respectively, the cost satisfies [Goldenshluger and Mirkin, 2017, Lemma 1]

$$J = \gamma_0 + \lim_{T\to\infty} \mathbb{E}\Big[\frac{1}{T}\int_0^T \big\|D_{zu}\tilde{u}(t)\big\|^2 dt\Big]. \tag{2.18}$$

If no further constraints are imposed on K, we immediately see in (2.18) that the cost is minimized by the choice $\mathcal{Q} = 0$, which results in $\tilde{u}(t) = 0$ at all times t. The minimum cost is then $\min_K J = \min_{\mathcal{Q}} J = \gamma_0$ and $K = \tilde{K}_{yu}$ is the analog LQG controller, as expected.

The next step is to impose constraints on $\mathcal{Q}$ such that the resulting K is a sampled-data controller. Necessary and sufficient constraints were first identified in [Mirkin, 2017] by considering the closed-loop system in the

lifted domain [Yamamoto, 1994]. Translated back to the continuous-time domain, the constraints state that K will be a causal sampled-data controller for an arbitrary sampling sequence $\{\tau_k\}_{k\in\mathbb{N}_0}$ iff $\mathcal{Q}$ can be decomposed as [Goldenshluger and Mirkin, 2017, Lemma 2]

$$\mathcal{Q} = \mathcal{Q}_{\text{stat}} + \mathcal{Q}_{\text{sd}}, \tag{2.19}$$

where $\mathcal{Q}_{\text{stat}} : \tilde{y} \mapsto \tilde{u}_{\text{stat}}$ is a static system in the lifted domain, with the continuous-time realization

$$\mathcal{Q}_{\text{stat}} : \begin{cases} \dot{\tilde{x}}(t) = A\tilde{x}(t) + L\tilde{y}(t), & \tilde{x}(\tau_k) = 0, \\ \tilde{u}_{\text{stat}}(t) = F\tilde{x}(t), \end{cases} \tag{2.20}$$

and $\mathcal{Q}_{\text{sd}} : \tilde{y} \mapsto \tilde{u}_{\text{sd}}$ is any stable, causal sampled-data system for the same sampling sequence $\{\tau_k\}$. For a $\mathcal{Q}$ of the form in (2.19) it holds that [Goldenshluger and Mirkin, 2017, Lemma 3]

$$\begin{aligned} &\lim_{T\to\infty} \mathbb{E}\Big[\frac{1}{T}\int_0^T \big\|D_{zu}\tilde{u}(t)\big\|^2 dt\Big] \\ &\qquad = \lim_{T\to\infty} \mathbb{E}\Big[\frac{1}{T}\int_0^T \big(\big\|D_{zu}\tilde{u}_{\text{stat}}(t)\big\|^2 + \big\|D_{zu}\tilde{u}_{\text{sd}}(t)\big\|^2\big) dt\Big]. \end{aligned} \tag{2.21}$$

So, by imposing sampled-data constraints on K, we are left with the free parameter $\mathcal{Q}_{\text{sd}}$ in the optimization. However, from (2.21) we see that $\mathcal{Q}_{\text{sd}} = 0$ results in an optimal sampled-data controller with $\mathcal{Q} = \mathcal{Q}_{\text{stat}}$, and thus $\tilde{u} = \tilde{u}_{\text{stat}}$. From (2.17) and (2.20) we then get the following realization of K

$$K : \begin{cases} \begin{bmatrix} \dot{\hat{x}}(t) \\ \dot{\tilde{x}}(t) \end{bmatrix} = \begin{bmatrix} \big(A + B_u F + LC_y\big) & B_u F \\ -LC_y & A \end{bmatrix} \begin{bmatrix} \hat{x}(t) \\ \tilde{x}(t) \end{bmatrix} + \begin{bmatrix} -L \\ L \end{bmatrix} y(t), \\ \tilde{x}(\tau_k) = 0, \\ u(t) = \begin{bmatrix} F & F \end{bmatrix} \begin{bmatrix} \hat{x}(t) \\ \tilde{x}(t) \end{bmatrix}, \end{cases} \tag{2.22}$$

which achieves the minimal cost

$$\min_K J = \gamma_0 + \underbrace{\lim_{T\to\infty} \mathbb{E}\Big[\frac{1}{T}\int_0^T \|\tilde{x}(t)\|_Q^2 dt\Big]}_{:=J_\mu}, \tag{2.23}$$

where the weight matrix Q is defined in (2.8). The relation (2.23) holds for any sampling sequence—including those generated by event-based policies—and in the case of periodic sampling it can be simplified further

as [Braksmayer and Mirkin, 2017, Remark 4]

$$J_\mu = \frac{1}{h}\int_0^h \int_0^t \|D_{zu} F e^{vA} L D_{yw}\|_{\text{Frob}}^2 dv dt$$
$$= \frac{1}{h}\int_0^h \int_0^t \text{tr}(e^{vA} R e^{vA^\intercal} Q) dv dt, \quad (2.24)$$

where the matrix R is defined in (2.5).

The two terms on the right-hand side of (2.23) are the cost γ_0 from the corresponding analog LQG controller, and an additional cost J_μ due to the sampled-data constraint. Since the value of J_μ only depends on the choice of sampling policy μ, its optimization can be performed separately. Thus, there holds a principle of separation between the controller K and the sampling policy μ for the sampled-data LQG problem. The key to this property is the nested pattern of information in the feedback loop, where the information available at the actuator side is a subset of the information available to the sampling policy implemented at the sensor side. To the author's knowledge, this property was first noted in [Molin and Hirche, 2010a].

While not immediately apparent from (2.22), the optimal controller K has a sampled-data structure and can be translated to the form in Figure 2.3 with a sampler S, a hold H, and a discrete-time controller $\bar{K}$. However, before presenting this version, we first consider an equivalent realization of the form in (2.22), which provides more insight into the controller's operation.

An Equivalent Realization

As described in [Mirkin, 2017], we can introduce the state $x_a := \hat{x} + \tilde{x}$ and formulate an equivalent realization of (2.22) as

$$K : \begin{cases} \begin{bmatrix} \dot{\hat{x}}(t) \\ \dot{x}_a(t) \end{bmatrix} = \begin{bmatrix} A + LC_y & B_u F \\ 0 & A + B_u F \end{bmatrix} \begin{bmatrix} \hat{x}(t) \\ x_a(t) \end{bmatrix} + \begin{bmatrix} -L \\ 0 \end{bmatrix} y(t), \\ x_a(\tau_k) = \hat{x}(\tau_k), \\ u(t) = \begin{bmatrix} 0 & F \end{bmatrix} \begin{bmatrix} \hat{x}(t) \\ x_a(t) \end{bmatrix}. \end{cases} \quad (2.25)$$

We see in (2.25) that the control signal is solely based on the state vector x_a, which operates in open-loop between sampling times. Feedback from the measured signal y is achieved only at sampling times through the sampled-data operation $x_a(\tau_k) = \hat{x}(\tau_k)$. Comparing (2.25) to the analog LQG controller (2.9), we see that the optimal sampled-data controller operates like a KBF on the sensor side (cf. (2.3)), which, at sampling times, transmits its estimate $\hat{x}$ to an LQR (cf. (2.6)) based on the simulated plant state x_a

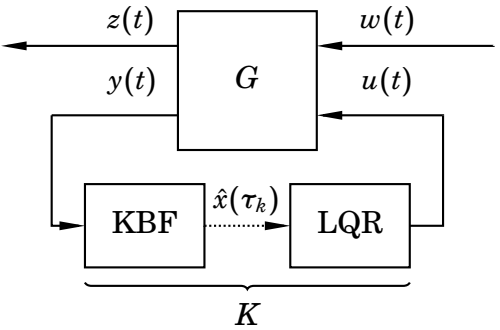

Figure 2.7 Equivalent realization of the optimal sampled-data controller K, with a Kalman–Bucy filter (KBF) on the sensor side and a simulated linear–quadratic regulator (LQR) on the actuator side of the plant G.

on the actuator side, see Figure 2.7. Naturally, a copy of the simulated LQR has to be featured on the sensor side to provide the control signal to the KBF. In principle, this poses no problem as the operation of the simulated LQR is completely deterministic from the KBF's point of view.

Sampled-Data Structure

We now present the optimal controller in (2.25) decomposed into the form considered in Figure 2.3, with a sampler S, a hold H and a discrete-time controller $\bar{K}$. The translation is performed by integrating the system of differential equations in (2.25) using Van Loan's formula [Van Loan, 1978, Theorem 1] over one sampling interval and identifying the corresponding sampling and hold functions ψ and ϕ, respectively. For a detailed proof, see [Mirkin, 2017, Corollary 4.3].

The sampling function ψ is given by

$$\psi(\tau_k, v) = e^{(\tau_k - v)(A + LC_y)} L \mathbf{1}_{[0,h_k]}(\tau_k - v), \quad \forall v \in [\tau_{k-1},\ \tau_k], \tag{2.26}$$

while the hold function ϕ is given by

$$\phi(t, \tau_k) = F e^{(t-\tau_k)(A+B_uF)} \mathbf{1}_{[0,h_{k+1}]}(t - \tau_k), \quad \forall t \in [\tau_k,\ \tau_{k+1}]. \tag{2.27}$$

The discrete-time controller $\bar{K}$ is given by

$$\bar{u}_k = \Psi(h_k)\bar{u}_{k-1} + \bar{y}_k, \tag{2.28}$$

where

$$\Psi(h_k) \coloneqq e^{h_k(A+LC_y)}\Big(I + \int_0^{h_k} e^{-v(A+LC_y)} B_u F e^{v(A+B_uF)} dv\Big). \tag{2.29}$$

Note that the discrete-time controller (2.28) is time-varying in general, but constant for periodic sampling when $h_k = h$ for all k.

We refer to the LQG-optimal sampled-data controller given by the two equivalent forms (2.25) and (2.26)–(2.28) as *Mirkin's LQG controller*.

Example 2.1
We conclude this chapter by considering an example where we compare the performance between the presented LQG controllers. Consider a simple LTI model of a DC motor, with parameters [Glad and Ljung, 2000, (4.15)]

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, \quad B_u = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_y = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{2.30}$$

Here, y represent the measured position of the DC motor, while u represents the input voltage.

Let us consider the LQG design problem for the DC motor with the following (quite arbitrary) noise and cost parameters

$$B_w = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad D_{yw} = \begin{bmatrix} 0 & 0.1 \end{bmatrix}, \quad C_z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad D_{zu} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{2.31}$$

Solving the continuous-time AREs (2.5) and (2.8), we get

$$F = \begin{bmatrix} -1 & -1 \end{bmatrix}, \quad L = \begin{bmatrix} -3.58 & -6.42 \end{bmatrix}^{\mathsf{T}},$$

and the cost $\gamma_0 = 1.46$ via (2.10) for the analog LQG controller. We now consider the cost J for three sampled-data controllers:

(i) the baseline LQG controller,

(ii) Mirkin's LQG controller with periodic sampling, and

(iii) Mirkin's LQG controller with optimal event-based sampling.

Optimal event-based sampling will be formally introduced in Chapter 3, but we include it in this example as a reference for achievable performance.

The performance of the three controllers is shown in Figure 2.8, where the cost J normalized by γ_0 is plotted against the average sampling interval $\bar{h}$ (defined in (3.18)). By the rule of thumb given in [Åström and Wittenmark, 1997, Chapter 4], the sampling period of the baseline LQG controller should be smaller than roughly $h = 0.2$ for the design in this example. At that point, the resulting cost is approximately 8% higher than that of the analog LQG. By switching to Mirkin's LQG controller with periodic sampling, we can use a slightly longer period, $h = 0.22$, for the same cost. However, by instead considering event-based sampling for Mirkin's LQG controller, we

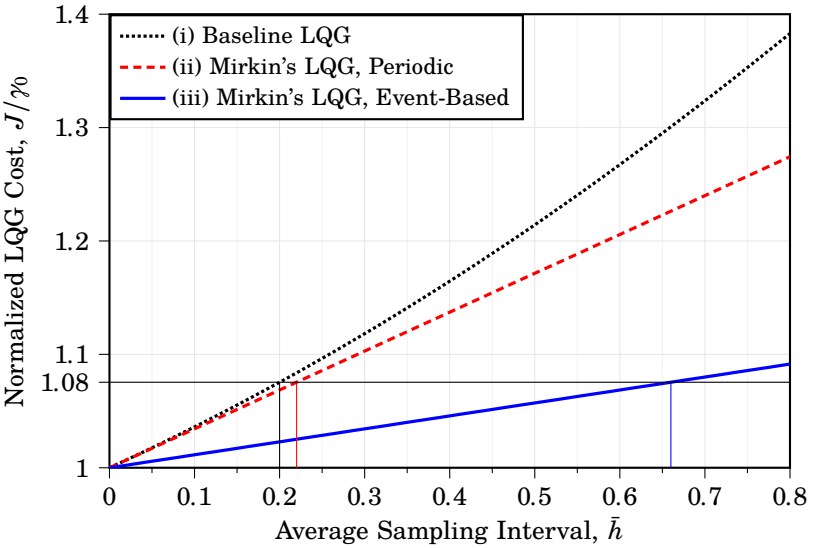

Figure 2.8 Normalized LQG cost J/γ_0 versus average sampling interval $\bar{h}$ for the DC motor system in Example 2.1. Trade-off curves are shown for the baseline LQG controller (black, dotted) and Mirkin's LQG controller with periodic (red, dashed) and optimal event-based sampling (blue, solid). The cost level $J/\gamma_0 = 1.08$ has been highlighted for reference, corresponding to $\bar{h} = 0.2$, 0.22 and 0.66 for the three controllers respectively.

can achieve $\bar{h} = 0.66$ for the same cost—three times larger compared to periodic sampling. Thus, the results in Figure 2.8 (and similar examples like it, see [Braksmayer and Mirkin, 2017; Thelander Andrén et al., 2017]) indicate that the main performance contributor is the choice of sampling policy, rather than the sampled-data structure of the controller.

It can be shown that the factor three improvement obtained from event-based sampling in fact holds for any cost level we might consider in Figure 2.8 (c.f Section 3.4). This is the same observation made in the example in Section 1.1 based on the seminal work in [Åström and Bernhardsson, 1999], and there is a clear link—optimizing the sampling policy is essentially an impulse control problem. This link is further explained in Chapter 3.

3

The Optimal Sampling Problem

In the previous chapter, we concluded that the co-design problem in sampled-data LQG control is separable, with a known solution for the sampler, hold, and discrete-time controller. This chapter focuses on the design of the final component—the sampling policy.

The objective is to design a policy that achieves an optimal trade-off between control performance and resource utilization. This trade-off is expressed in terms of LQG cost and average sampling rate, and we begin this chapter with a formal definition of this objective—denoted as the *optimal sampling problem* (OSP). To characterize the solution of the OSP, we proceed by deriving the corresponding Hamilton–Jacobi–Bellman (HJB) equation for the value function of the optimization problem. This HJB equation has the form of a free boundary problem—a special kind of partial-differential equation common in impulse control problems. While free boundary problems generally require numerical methods to solve, there are a few systems with special structures where the considered HJB equation admits an exact solution. The chapter is concluded with a presentation of four such special structures—two known results from the literature, and two new contributions.

3.1 Problem Formulation

From Chapter 2 we recall that the LQG cost J for the optimal sampled-data controller (Mirkin's LQG controller) is given by

$$J = \gamma_0 + J_\mu,$$

where γ_0 is the cost for the corresponding analog LQG controller (cf. (2.10)) and

$$J_\mu = \lim_{T\to\infty} \mathbb{E}\Big[\frac{1}{T}\int_0^T \big\|\tilde{x}(t)\big\|_Q^2 dt\Big], \tag{3.1}$$

is the cost due to the sampled-data operation (cf. (2.23)), with $Q \coloneqq F^\intercal D_{zu}^\intercal D_{zu} F$ (cf. (2.8)). The cost J_μ depends on the trajectory of the state $\tilde{x} \in \mathbb{R}^n$, which is governed by the dynamics

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + L\tilde{y}(t), \qquad \tilde{x}(\tau_k) = 0, \tag{3.2}$$

i.e., $\tilde{x}$ is reset to the origin at each sampling time τ_k. We recall from Section 2.5 that $\tilde{x} \coloneqq x_a - \hat{x}$ represents the error between the estimate $\hat{x}$ of a KBF on the sensor side of the plant, and the state x_a of an open-loop simulation of the plant on the actuator side (see Figure 2.7). The input signal $\tilde{y}$ in (3.2) represents the innovations signal of the KBF, i.e.

$$\tilde{y}(t) = y(t) - C_y\hat{x}(t),$$

where y is the measured plant output. The innovations signal is a continuous-time white Gaussian process with the same intensity as the measurement noise [Glad and Ljung, 2000, Theorem 5.5], and (3.2) can therefore equivalently be represented as

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + v(t), \qquad \tilde{x}(\tau_k) = 0, \tag{3.3}$$

where v is a continuous-time white Gaussian process with intensity $R \coloneqq LD_{yw}D_{yw}^\intercal L^\intercal$ (cf. (2.5)).

The only available design variable to control the trajectory of $\tilde{x}$—and ultimately the value of J_μ—is the policy μ for generating the sequence of sampling times $\{\tau_k\}_{k\in\mathbb{N}_0}$. We restrict the investigation of μ to *admissible* sampling policies, defined as follows:

Definition 3.1—Admissible Sampling Policy
A sampling policy μ is said to be *admissible* if it generates a sequence of sampling times $0 = \tau_0 < \tau_1 < \tau_2 \ldots$ such that

(i) the process $\tilde{x}$ exists and is unique, and

(ii) $\lim_{k\to\infty} \tau_k = \infty$. ♢

The first item of Definition 3.1 ensures that the trajectory of $\tilde{x}$ is well-defined, while the second item excludes policies that lead to Zeno behaviour, i.e., where an infinite number of sampling times occur over a finite time interval [Miskowicz, 2015, Chapter 1].

The objective is to find an admissible sampling policy μ that achieves an efficient trade-off between sampling rate (incurring costs in, e.g., energy and network bandwidth) and closed-loop performance. To this end, we define the average sampling rate as

$$f \coloneqq \limsup_{T\to\infty} \frac{\mathbb{E}[k_T]}{T}, \tag{3.4}$$

where $k_T = \max_{\tau_k \le T} k$ counts the number of sampling actions up until time T. With an added penalty $\rho > 0$ per sampling action, we arrive at the optimal sampling problem (OSP).

Definition 3.2—The Optimal Sampling Problem
Consider the cost J_μ in (3.1), where the dynamics of $\tilde{x}$ are given by (3.3). For a given penalty $\rho > 0$, the optimal sampling problem is

$$\begin{aligned} \underset{\mu}{\text{minimize}} \quad & J_{\text{osp}} \coloneqq J_\mu + \rho f, \\ \text{subject to} \quad & \mu \text{ admissible.} \end{aligned} \tag{3.5}$$

$\diamondsuit$

Note that (3.3) can also be viewed as a system with impulse feedback of the form

$$\begin{cases} \dot{\tilde{x}}(t) = A\tilde{x}(t) + v(t) + u(t), \\ u(t) = -\sum_{k=0}^{\infty} \tilde{x}(t)\delta(t-\tau_k). \end{cases}$$

The OSP is thus equivalent to an impulse control problem, here with long-term average cost. Optimal policies for this problem class are typically Markovian, and trigger impulses (sampling) based only on the current state value [Robin, 1983]. Specifically, an optimal policy typically has the form

$$\tau_k = \min\left\{ t > \tau_{k-1} : g\big(\tilde{x}(t)\big) = 0 \right\}, \quad \tau_0 = 0, \tag{3.6}$$

for some *trigger function* $g : \mathbb{R}^n \mapsto \mathbb{R}$. The policy (3.6) triggers an impulse (sampling) whenever the trajectory of $\tilde{x}$ reaches the level set

$$\partial\Omega \coloneqq \left\{ \tilde{x} : g(\tilde{x}) = 0 \right\}.$$

This means that optimal sampling policies for this problem class are inherently event-based rather than time-triggered, and that the OSP can be viewed as a search for a threshold set $\partial\Omega$ in the state space, see Figure 3.1. Finding $\partial\Omega$ is an optimal control problem with two options at each point in time; (i) do nothing, and accumulate an uncertain quadratic cost, or (ii) reset $\tilde{x}$ and pay a fixed penalty ρ.

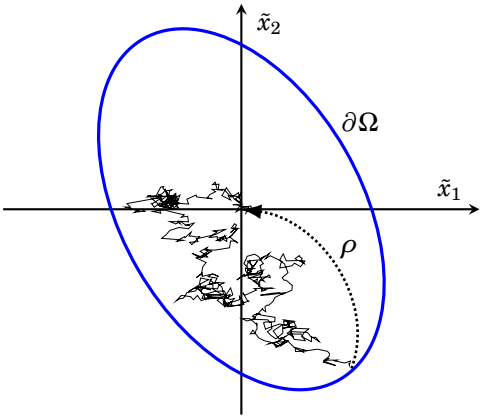

Figure 3.1 The optimal sampling problem (here in two dimensions) can be viewed as the search for a threshold $\partial\Omega$ (blue) on the trajectory of $\tilde{x}$ (black), from where it should be reset to the origin and incur the penalty ρ (dotted arrow).

3.2 Hamilton–Jacobi–Bellman Formulation

We now take the first step towards solving the OSP using the framework of dynamic programming. Central in continuous-time dynamic programming is the *Hamilton–Jacobi–Bellman* (HJB) equation, which is a partial-differential equation (PDE) in the value function of the optimization problem (see e.g. [Bertsekas, 2012]). By obtaining a solution to the HJB equation, we implicitly also retrieve the optimal policy.

Derivation

The HJB equation for the OSP was first derived in [Henningsson, 2012]. Here, we give an outline of the derivation and refer to [Henningsson, 2012, Paper II] for a detailed proof.

First, consider a bounded function $V : \mathbb{R}^n \mapsto \mathbb{R}$, and let

$$I_\mu(T) \coloneqq \Big\{ \tau_k : k = 0 \dots k_T \Big\},$$

denote the sequence of sampling times generated by an admissible policy μ up until time T. Furthermore, define $\mathcal{T} \coloneqq [0, T] \setminus I_\mu(T)$ as the set of intervals between sampling times up until time T. By Dynkin's formula (see, e.g., [Øksendal, 2000, Theorem 7.4.1]), the expected change in V

$$\Delta_T V \coloneqq V\big(\tilde{x}(T)\big) - V\big(\tilde{x}(0)\big),$$

is given by

$$\mathbb{E}\big[\Delta_T V\big] = \mathbb{E}\Big[\int_{\mathcal{T}} \mathcal{A}V\big(\tilde{x}(t)\big)dt + \sum_{\tau_k \in I_\mu(T)} \Big(V(0) - V\big(\tilde{x}(\tau_k^-)\big)\Big)\Big], \tag{3.7}$$

where $\tilde{x}(\tau_k^-)$ denotes the limit value of $\tilde{x}$ just before time τ_k. The infinitesimal generator $\mathcal{A}(\cdot)$ corresponding to (3.3) between sampling times is given by [Øksendal, 2000, Theorem 7.3.3]

$$\mathcal{A}V(\tilde{x}) = \frac{1}{2}\mathrm{tr}\big(R\nabla_{\tilde{x}}^2 V(\tilde{x})\big) + \tilde{x}^\intercal A^\intercal \nabla_{\tilde{x}} V(\tilde{x}). \tag{3.8}$$

Let $v(T)$ denote the non-averaged cost over the interval $[0,\ T]$,

$$v(T) \coloneqq \mathbb{E}\Big[\int_{\mathcal{T}} \big\|\tilde{x}(t)\big\|_Q^2 dt + \rho k_T\Big].$$

Adding $v(T)$ to (3.7) then yields

$$\begin{aligned} &v(T) + \mathbb{E}\big[\Delta_T V\big] \\ &= \mathbb{E}\Big[\int_{\mathcal{T}} (\mathcal{A}V\big(\tilde{x}(t)\big) + \big\|\tilde{x}(t)\big\|_Q^2)dt + \sum_{\tau_k \in I(T)} \Big(\rho + V(0) - V\big(\tilde{x}(\tau_j^-)\big)\Big)\Big]. \end{aligned} \tag{3.9}$$

Now, assume that V satisfies the following *complementarity conditions* for some constant $\lambda > 0$

$$\forall \tilde{x} \in \mathbb{R}^n : \begin{cases} 0 = \Big(\mathcal{A}V(\tilde{x}) + \big\|\tilde{x}\big\|_Q^2 - \lambda\Big) \cdot \Big(\rho + V(0) - V(\tilde{x})\Big), \\ 0 \leq \mathcal{A}V(\tilde{x}) + \big\|\tilde{x}\big\|_Q^2 - \lambda, \\ 0 \leq \rho + V(0) - V(\tilde{x}). \end{cases} \tag{3.10}$$

The complementarity conditions imply that equality holds in at least one of the two inequalities in (3.10) for all $\tilde{x}$. With these inequalities we obtain a lower bound on (3.9) as

$$v(T) + \mathbb{E}\big[\Delta_T V\big] \geq \mathbb{E}[\int_{\mathcal{T}} \lambda dt] = \lambda T. \tag{3.11}$$

Now, dividing by T yields

$$\frac{v(T)}{T} + \frac{\mathbb{E}\big[\Delta_T V\big]}{T} \geq \lambda. \tag{3.12}$$

Since

$$\lim_{T\to\infty} \frac{v(T)}{T} = J_{\mathrm{osp}},$$

and boundedness of V implies

$$\lim_{T\to\infty} \frac{\mathbb{E}\big[\Delta_T V\big]}{T} = 0,$$

we obtain a lower bound on the OSP cost by taking the limit $T \to \infty$ in (3.12), resulting in:

$$J_{\text{osp}} \geq \lambda. \tag{3.13}$$

Note that the bound (3.13) holds for any admissible sampling policy. Since the complementarity conditions imply

$$\rho + V(0) - V(\tilde{x}) > 0 \implies \mathcal{A}V(\tilde{x}) + \big\|\tilde{x}\big\|_Q^2 = \lambda,$$

we see from (3.9) that equality is attained in (3.11)—and ultimately (3.13)—if we pick the policy

$$\mu^\star : \tau_k = \min\Big\{t > \tau_{k-1} : \rho + V(0) - V\big(\tilde{x}(t)\big) = 0\Big\}, \quad \tau_0 = 0. \tag{3.14}$$

Thus, if a pair $(V,\, \lambda)$ are found satisfying the complementarity conditions (3.10), then $\min_\mu J_{\text{osp}} = \lambda$, and an optimal admissible sampling policy $\mu^\star$ is to trigger sampling whenever the trajectory of $\tilde{x}$ reaches the level set

$$\partial\Omega = \Big\{\tilde{x} : \rho + V(0) - V(\tilde{x}) = 0\Big\}.$$

The result is summarized in the following theorem.

Theorem 3.1
Assume that a bounded function $V : \mathbb{R}^n \mapsto \mathbb{R}$ and a constant $\lambda > 0$ are found satisfying the complementarity conditions (3.10). Then the minimum cost in the OSP is $\min_\mu J_{\text{osp}} = \lambda$, and an optimal sampling policy is of the form (3.6), with the trigger function

$$g(\tilde{x}) = \rho + V(0) - V(\tilde{x}).$$

Proof. See [Henningsson, 2012, Paper II, Theorem 1]. □

For a given value of λ, the nonlinear PDE (3.10) is a *free-boundary problem* (here presented in complementarity form). The boundary $\partial\Omega$ is said to be free since it is not known beforehand but is implicitly given as a part of the solution. Free boundary PDEs typically arise in optimal stopping problems [Peskir and Shiryaev, 2006], where perhaps the most well-studied application is the pricing of American-style options in finance (see e.g. [Björk, 2019] and references therein). In fact, as will be seen in Section 3.3, the OSP can also be formulated as an optimal stopping problem.

Solution Properties

The function V is known as a *relative value function*. Its interpretation is only meaningful as a difference; for two initial states $\tilde{x}_a$ and $\tilde{x}_b$ the quantity $V(\tilde{x}_a) - V(\tilde{x}_b)$ is the expected difference in cost under the optimal sampling policy over any given time horizon. Consequently, the specific value of $V(0)$ does not matter. This is also apparent in (3.10), where we see that V can only be unique up to a constant. Hence, we may, without loss of generality, restrict our attention to solutions satsifying

$$V(0) \coloneqq -\rho, \tag{3.15}$$

which has the benefit of eliminating the explicit dependence of ρ in (3.10). Having specified (3.15), it can be shown with the results from [Øksendal and Reikvam, 1998] that there exists a unique solution (in the viscosity sense, see [Øksendal and Reikvam, 1998, Definition 2.1]) to (3.10) for any given λ as long as the sampling intervals are finite with probability one.

While results on existence and uniqueness are available, it is more difficult to estimate the regularity of the solution. It is rare to find smooth solutions to problems such as (3.10) due to the free boundary, where non-smoothness may occur. Typically, a solution will be $\mathcal{C}^2$ over the entire domain except over the free boundary, where the solution is only $\mathcal{C}^1$ (known as the *high contact principle*) [Øksendal, 2000, Chapter 10]. Formally, we can obtain regularity estimates under the assumptions that the covariance matrix R is positive definite, and that the domain of $\tilde{x}$ is a bounded set Ξ with sufficiently smooth boundary $\partial\Xi$ (see [Friedman, 1982, Chapter 1, (3.11)]), with the boundary condition

$$V(\tilde{x}) = 0, \quad \forall \tilde{x} \in \partial\Xi.$$

The domain restriction poses no problem for non-pathological cases when the solution of the original problem has a bounded threshold $\partial\Omega$, since we can then simply choose Ξ large enough to encompass $\partial\Omega$. With these assumptions, (3.10) becomes a uniformly elliptic obstacle problem—a well-studied problem in the PDE literature (see, e.g., [Friedman, 1982]). The solution is then unique, with [Friedman, 1982, Chapter 1, Theorem 3.2]

$$V \in W^{2,p}(\Xi), \quad 1 \leq p < \infty, \tag{3.16}$$

where $W^{2,p}(\Xi)$ denotes the Sobolev space of functions on the domain Ξ, with weak derivatives up to order 2 having finite L^p norm. The implications of (3.16) are threefold. First, we know that the solution is twice weakly differentiable on the entire domain Ξ. Second, using the general Sobolev inequalities (see, e.g., [Evans, 1998, Section 5.5, Theorem 6]) we can show that

$$V \in \mathcal{C}^1(\Xi),$$

i.e., the solution is continuously differentiable on the entire domain Ξ. Third, by Meyer–Serrin's theorem (see, e.g., [Evans, 1998, Section 5.3, Theorem 3]) there exists an approximating sequence of smooth functions which converges to the solution. This third implication supports the choice of smooth basis functions in numerical approximations, as in the proposed method described in Section 4.3.

3.3 Optimal Stopping Formulation

As noted in Theorem 3.1, the optimal sampling policy is a static threshold of the form (3.6). By restricting the optimization to static sampling policies, we can equivalently formulate the OSP as an optimal stopping problem. As this formulation is useful for numerical evaluation, we briefly consider it before presenting some special cases where the OSP has an exact solution.

First, let us formally define what is considered a static sampling policy.

Definition 3.3—Static Sampling Policy
A sampling policy μ is said to be *static* if

(i) the k^{th} sampling decision only depends on $\tilde{x}(t)$ for $t \geq \tau_{k-1}$, and

(ii) trigger conditions for sampling are identical between intervals. ♢

The set of static sampling policies includes both periodic sampling and Markovian policies of the form (3.6), but also more general policies that may be time-varying within each interval. In particular, the optimal policy from Theorem 3.1 is still valid if we constrain the OSP to static policies.

A static sampling policy is memoryless between sampling intervals, similar to how the trajectory history of $\tilde{x}$ is reset at each sampling time. Due to this property, a static policy produces a sequence of i.i.d. sampling intervals $\{h_k \coloneqq \tau_k - \tau_{k-1}\}$. Furthermore, the process $k_t = \max_{\tau_k \leq t} k$ that counts the number of sampling actions up until time t will then be a *renewal process*. By the elementary renewal theorem, the average sampling rate as defined in (3.4) then satisfies

$$f = \frac{1}{\mathbb{E}[h_1]}, \tag{3.17}$$

where h_1 is the first sampling interval. For static sampling policies, we define

$$\bar{h} \coloneqq \mathbb{E}[h_1] = \mathbb{E}[\tau_1], \tag{3.18}$$

as the average sampling interval. The relation (3.17) is especially useful from a numerical standpoint, as f is then simple to evaluate via Monte Carlo methods. Similarly, the cost J_μ can also be expressed in terms of an expected value over a single interval [Goldenshluger and Mirkin, 2017, Proposition 1].

THEOREM 3.2
For static sampling policies, the cost J_μ is given by

$$J_\mu = \frac{1}{\bar{h}}\mathbb{E}\Big[\int_0^{\tau_1} \|\tilde{x}(t)\|_Q^2 dt\Big], \quad \tilde{x}(0) = 0.$$

Proof. See [Goldenshluger and Mirkin, 2017, Section III.B]. □

By (3.17) and Theorem 3.2, we conclude that the following relation holds for static sampling policies

$$\min_{\mu} J_{\text{osp}} = \min_{\tau_1} \frac{\mathbb{E}\big[\int_0^{\tau_1} \|\tilde{x}(t)\|_Q^2 dt\big] + \rho}{\bar{h}}, \tag{3.19}$$

where minimization over the stopping time τ_1 is subject to knowledge of $\tilde{x}(t)$ for $t \le \tau_1$ alone (i.e., non-causal policies are excluded). The formulation in (3.19) is an example of an *optimal stopping problem* (see [Peskir and Shiryaev, 2006]).

3.4 Exact Solutions for Some Special Cases

While free boundary problems generally require numerical methods, there are in fact some special cases where an exact solution to the OSP can be obtained. These special cases are

(i) first-order systems,

(ii) multivariate integrator systems,

(iii) spiral systems, and

(iv) single-input (SI) and single-output (SO) systems of certain structure.

Solutions for cases (i) and (ii) are known from the literature, but here we give some additional new results for (ii). Cases (iii) and (iv) are new contributions. For completeness, we present all four cases here (with (3.15), i.e., $V(0) = -\rho$, assumed throughout).

First-Order Systems

For first-order systems, the free boundary problem (3.10) reads

$$\forall \tilde{x} \in \mathbb{R} : \begin{cases} 0 = \Big(\dfrac{r}{2}\dfrac{\partial^2 V(\tilde{x})}{\partial \tilde{x}^2} + a\tilde{x}\dfrac{\partial V(\tilde{x})}{\partial \tilde{x}} + q\tilde{x}^2 - \lambda\Big) \cdot V(\tilde{x}), \\ 0 \le \dfrac{r}{2}\dfrac{\partial^2 V(\tilde{x})}{\partial \tilde{x}^2} + a\tilde{x}\dfrac{\partial V(\tilde{x})}{\partial \tilde{x}} + q\tilde{x}^2 - \lambda, \\ 0 \ge V(\tilde{x}), \end{cases} \tag{3.20}$$

where r, q and a denote scalar versions of R, Q and A, respectively. This case corresponds exactly to the one studied using Lebesgue sampling (i.e., send–on–delta) in [Åström and Bernhardsson, 1999], whose optimality for the OSP was subsequently proven in [Henningsson et al., 2008].

Solution. Based on the results in [Henningsson et al., 2008], the solution to the OSP for first-order systems can be summarized as follows.

THEOREM 3.3
For first-order systems, the optimal admissible sampling policy has the symmetric trigger function

$$g(\tilde{x}, \Delta_\rho) = \Delta_\rho - |\tilde{x}|, \tag{3.21}$$

where

$$\Delta_\rho \coloneqq \arg\min_{\Delta} \Big\{ J_\mu(\Delta) + \rho f(\Delta) \Big\},$$

and $J_\mu(\Delta)$, $f(\Delta)$ are the LQG cost and average sampling rate obtained using the trigger function $g(\tilde{x}, \Delta)$. Expressions for $J_\mu(\Delta)$ and $f(d) = 1/\bar{h}(\Delta)$ are given by

$$J_\mu(\Delta) = \frac{2q}{r\bar{h}(\Delta)} \int_0^\Delta \int_0^\beta \alpha^2 e^{\frac{a}{r}(\alpha^2-\beta^2)} d\alpha d\beta, \tag{3.22}$$

$$\bar{h}(\Delta) = \frac{2}{r} \int_0^\Delta \int_0^\beta e^{\frac{a}{r}(\alpha^2-\beta^2)} d\alpha d\beta. \tag{3.23}$$

The corresponding solution pair (V, λ) to (3.20) is given by

$$\lambda = J_\mu(\Delta_\rho) + \rho f(\Delta_\rho),$$
$$V(\tilde{x}) = \begin{cases} \frac{2}{r} \int_{\Delta_\rho}^{\tilde{x}} \int_0^\beta (\lambda - q\alpha^2) e^{\frac{a}{r}(\alpha^2-\beta^2)} d\alpha d\beta, & |\tilde{x}| \leq \Delta_\rho, \\ 0, \quad \text{otherwise.} \end{cases}$$

Proof. See [Henningsson et al., 2008, Section 3.3]. □

EXAMPLE 3.1
Consider the set of first-order systems defined by

$$\begin{aligned} A = a &\in \{-1,\ 0,\ 1\}, \\ B_u &= 1, \\ C_y &= 1, \\ B_w &= \begin{bmatrix} 1 & 0 \end{bmatrix}, \\ D_{yw} &= \begin{bmatrix} 0 & 1 \end{bmatrix}, \end{aligned}$$

i.e., a stable ($a = -1$), marginally stable ($a = 0$) and unstable system ($a = 1$) respectively. We consider an LQG control design for this set of systems specified by

$$C_z = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad D_{zu} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

resulting in

$$r = q = \begin{cases} 0.17, & a = -1, \\ 1, & a = 0, \\ 5.83, & a = 1. \end{cases}$$

While using Mirkin's LQG controller for this set of systems, we now compare performance of periodic and optimal event-based sampling. To this end, let $J_{\text{eb}}(\bar{h})$ and $J_{\text{per}}(\bar{h})$ denote the cost J_μ obtained with the average sampling interval $\bar{h}$ under optimal event-based and periodic sampling respectively. We then define the cost ratio as

$$J_{\text{ratio}}(\bar{h}) \coloneqq \frac{J_{\text{per}}(\bar{h})}{J_{\text{eb}}(\bar{h})}, \tag{3.24}$$

where optimality of the event-based policy implies $J_{\text{ratio}} \geq 1$. This ratio quantifies the potential benefit of using event-based sampling and corresponds to the ratio studied in the motivating example in Section 1.1. Large values of J_{ratio} indicates a significant improvement, whereas $J_{\text{ratio}} = 1$ corresponds to no difference.

Starting with periodic sampling, it is straightforward to show, using (2.24), that

$$J_{\text{per}}(\bar{h}) = \begin{cases} \dfrac{qr}{2a}\Big(\dfrac{e^{2a\bar{h}} - 1}{2a\bar{h}} - 1\Big), & a \neq 0, \\ \dfrac{qr}{2}\bar{h}, & a = 0. \end{cases} \tag{3.25}$$

For optimal event-based sampling, we can obtain an analytic expression for the marginally stable case from (3.22) and (3.23). The cost is then given by

$$J_{\text{eb}}(\bar{h}) = \frac{qr}{6}\bar{h}, \quad a = 0. \tag{3.26}$$

Thus, for the marginally stable case we have

$$J_{\text{ratio}}(\bar{h}) = 3, \quad a = 0, \tag{3.27}$$

i.e., the well-known factor three improvement from [Åström and Bernhardsson, 1999].

Since there is no analytic expression for J_{eb} when $a \neq 0$, we can not obtain similar expressions for the cases $a = 1$ and $a = -1$. Instead, we

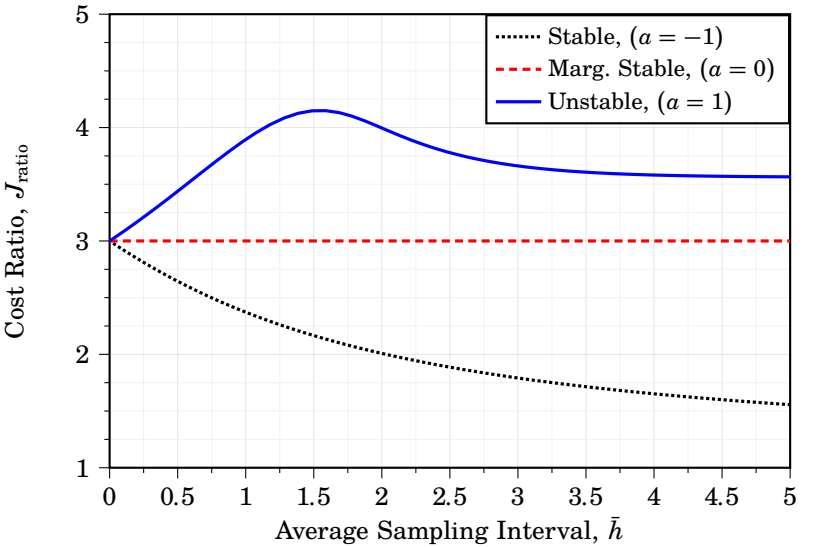

Figure 3.2 Cost ratio $J_{\text{ratio}}(\bar{h}) = J_{\text{per}}(\bar{h})/J_{\text{eb}}(\bar{h})$ between periodic and optimal event-based sampling over varying average sampling intervals $\bar{h}$ for the stable (black, dotted), marginally stable (red, dashed) and unstable (blue, solid) first-order systems in Example 3.1.

evaluate J_{eb} for these cases by numerical integration of (3.22) and (3.23). The results are shown in Figure 3.2. As concluded in (3.27), the cost ratio is constant $J_{\text{ratio}}(\bar{h}) = 3$ in the marginally stable case, and we additionally see that $J_{\text{ratio}}(\bar{h}) \approx 3$ for small sampling intervals in all three cases.

In the stable case ($a = -1$), J_{ratio} is monotonically decreasing. The cost ratio then approaches 1 asymptotically, at which point both strategies attain the same open-loop control cost. The benefit of event-based sampling is thus most prominent for relatively small sampling intervals in the stable case.

In the unstable case ($a = 1$), we initially see the opposite behavior. Here, J_{ratio} is increasing up until $\bar{h} \approx 1.5$, where it has a peak value of $J_{\text{ratio}} \approx 4.15$. The cost ratio is then decreasing with $\bar{h}$, and approaches $J_{\text{ratio}} \approx 3.54$ asymptotically. This curious non-monotonic behavior of J_{ratio} was not observed in [Åström and Bernhardsson, 1999], where the authors considered a smaller range of $\bar{h}$. By only seeing the initial rise of J_{ratio}, one might believe that the benefit of event-based sampling grows monotonically with the average sampling interval, and that J_{ratio} can be made arbitrarily large by considering increasingly larger $\bar{h}$. Clearly, this is not true, and the cost ratio is in fact bounded even in the unstable case.

Multivariate Integrator Systems

We now consider the case of systems with $A \coloneqq 0$, which we refer to as *multivariate integrator systems*. This system class has a closed-form solution for any system order n.

Solution. The OSP solution for multivariate integrator systems was first presented in [Henningsson, 2012, Paper II, Theorem 2]. The result is summarized in the following theorem.

THEOREM 3.4
For multivariate integrator systems ($A = 0$), the optimal admissible sampling policy has an ellipsoidal trigger function given by

$$g(\tilde{x}) = 2\sqrt{\rho} - \left\|\tilde{x}(t)\right\|_P^2. \tag{3.28}$$

The matrix $P \succ 0$ is the unique solution of the Riccati-like equation

$$PRP + \frac{1}{2}\operatorname{tr}(RP)P = Q. \tag{3.29}$$

The LQG cost J_μ and average samplig rate f under the optimal policy are given by

$$J_\mu = \rho f = \frac{1}{2}\sqrt{\rho}\operatorname{tr}(RP). \tag{3.30}$$

The corresponding solution pair (V, λ) to (3.10) is given by

$$\begin{aligned} \lambda &= \sqrt{\rho}\operatorname{tr}(RP), \\ V(\tilde{x}) &= -\frac{1}{4}\Big(\max\big(0,\ g(\tilde{x})\big)\Big)^2. \end{aligned} \tag{3.31}$$

Proof. See [Henningsson, 2012, Paper II, Theorem 2]. □

The solution in Theorem 3.4 is in fact valid for a somewhat extended class of systems [Thelander Andrén and Cervin, 2020, Corollary 1].

COROLLARY 3.1
The solution in Theorem 3.4 holds for all systems where the matrix PA is skew-symmetric, and P is the solution to the Riccati-like equation (3.29).

Proof. With V given by (3.31), the first-order term of the infinitesimal generator $\mathcal{A}(\cdot)$ (cf. (3.8)) becomes

$$\tilde{x}^\mathsf{T} A^\mathsf{T} \nabla_{\tilde{x}} V = (2\sqrt{\rho} - \|\tilde{x}\|_P^2)\tilde{x}^\mathsf{T} A^\mathsf{T} P\tilde{x} = 0,$$

where the last equality holds since PA is skew-symmetric. The PDE (3.10) is then identical to the case with $A = 0$, and both cases therefore share the same solution given by Theorem 3.4. □

The solution in Theorem 3.4 requires us to solve the Riccati-like equation (3.29), which is nonlinear in P and thus cannot be solved directly through semidefinite programming. However, a solution is easy to obtain with the following result [Thelander Andrén et al., 2017, Theorem 4].

THEOREM 3.5
Consider the matrix $S \coloneqq Q^{\frac{1}{2}} R Q^{\frac{1}{2}} \succ 0$, which has an eigendecomposition of the form

$$S = U_S \Lambda_S U_S^\mathsf{T}, \quad U_S U_S^\mathsf{T} = I.$$

Here, Λ_S is a diagonal matrix with the eigenvalues of S along its diagonal. The solution to (3.29) is then given by

$$P = Q^{\frac{1}{2}} U_S \Lambda_P U_S^\mathsf{T} Q^{\frac{1}{2}},$$

where Λ_P is a diagonal matrix with elements $\{\Lambda_{P,ii}\}_{i=1,\dots,n}$ given by

$$\Lambda_{P,ii} = -\frac{\alpha}{4\Lambda_{S,ii}} + \sqrt{\frac{\alpha^2}{16\Lambda_{S,ii}^2} + \frac{1}{\Lambda_{S,ii}}}, \quad i = 1, \dots, n,$$

where $\alpha > 0$ is the unique root of the scalar function

$$l(\alpha) \coloneqq (n+4)\alpha - \sum_{i=1}^{n} \sqrt{\alpha^2 + 16\Lambda_{S,ii}}.$$

Proof. The proof is given in Appendix C. □

From Theorem 3.5 we see that the optimal sampling policy can be obtained via a simple scalar search in the multivariate integrator case.

Comparison to Periodic Sampling. Since the cost J_μ and average sampling interval $\bar{h} = 1/f$ under the optimal policy are available in closed form in the multivariate integrator case, we can compare performance to periodic sampling analytically. Via (3.30) it is straightforward to express J_μ as a function of $\bar{h}$ for optimal event-based sampling as

$$J_{\text{eb}}(\bar{h}) = \frac{\left(\text{tr}(RP)\right)^2}{4}\bar{h}. \tag{3.32}$$

We can obtain the corresponding relation for periodic sampling via (2.24), resulting in

$$J_{\text{per}}(\bar{h}) = \frac{\text{tr}(RQ)}{2}\bar{h}. \tag{3.33}$$

By comparing (3.32) and (3.33) for the same given $\bar{h}$, we arrive at the following result [Thelander Andrén et al., 2017, Theorem 5].

THEOREM 3.6
As defined in (3.24), let J_{ratio} denote the ratio of LQG cost J_μ between periodic and optimal event-based sampling for the same average sampling interval $\bar{h}$. In the multivariate integrator case, J_{ratio} is constant for all $\bar{h}$ and satisfies

$$\begin{cases} J_{\text{ratio}} = 3, & n = 1, \\ J_{\text{ratio}} \in \left[1 + \frac{2}{n},\, 3\right), & n \geq 2. \end{cases} \tag{3.34}$$

Furthermore, the bounds for $n \geq 2$ are tight, with equality in the lower bound when the matrix RQ only has repeated eigenvalues. The upper bound is the limit value of J_{ratio} when all but one eigenvalue of RQ approach zero.

Proof. The proof is given in Appendix C. □

The conclusion from Theorem 3.6 is that optimal event-based sampling always outperform periodic sampling in the multivariate integrator case, and at most with a factor three. However, for the lower bound in Theorem 3.6 we note that

$$1 + \frac{2}{n} \to 1, \quad \text{as } n \to \infty.$$

The performance improvement is thus negligible for those higher-order systems where the eigenvalues of RQ are of similar magnitude.

EXAMPLE 3.2
Consider the following second-order system with parameters

$$\begin{aligned} A &= 0_{2\times 2}, \\ B_u &= I_2, \\ C_y &= I_2, \\ B_w &= \begin{bmatrix} I_2 & 0_{2\times 2} \end{bmatrix}, \\ D_{yw} &= \begin{bmatrix} 0_{2\times 2} & I_2 \end{bmatrix}. \end{aligned}$$

We consider an LQG control design for this system specified by

$$C_z = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\sqrt{\frac{5}{2}} & \sqrt{\frac{5}{2}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad D_{zu} = \begin{bmatrix} 0_{2\times 2} \\ I_2 \end{bmatrix}.$$

Since $A = 0$, we see from the AREs (2.5) and (2.8) that

$$\begin{aligned} R &= B_w B_w^\mathsf{T} = I_2, \\ Q &= C_z^\mathsf{T} C_z = \begin{bmatrix} 3 & -2 \\ -2 & 3 \end{bmatrix}. \end{aligned}$$

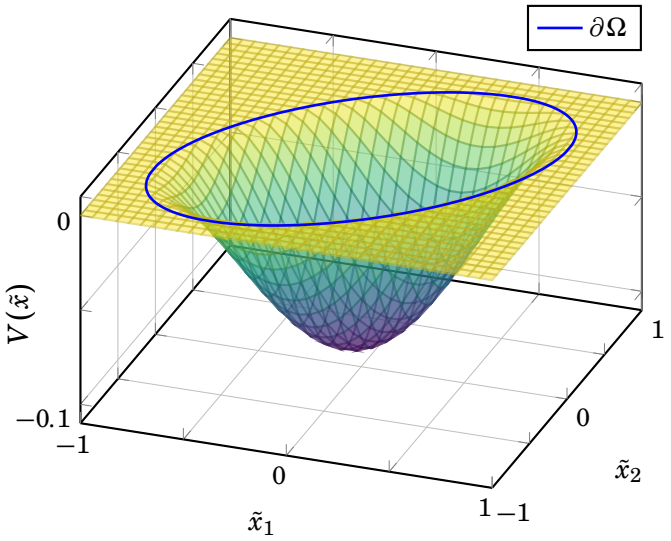

Figure 3.3 Relative value function V (surface) and threshold $\partial\Omega$ (blue line) of the optimal sampling policy for the multivariate integrator system in Example 3.2 with $\rho = 0.1$.

Given this specification, we now compare the trade-off between LQG cost and average sampling interval for the following three sampled-data controllers:

(i) the baseline LQG controller,

(ii) Mirkin's LQG controller with periodic sampling, and

(iii) Mirkin's LQG controller with optimal event-based sampling.

The LQG cost is computed using (2.15), (2.24) and (3.32) for controllers (i), (ii) and (iii) respectively for different average sampling intervals $\bar{h}$. The optimal sampling policy is computed via the scalar search method in Theorem 3.5, where the matrix P for this example evaluates to

$$P = \begin{bmatrix} 1.15 & -0.58 \\ -0.58 & 1.15 \end{bmatrix}.$$

The relative value function and the optimal threshold policy for $\rho = 0.1$ (corresponding to $\bar{h} = 0.27$) are shown in Figure 3.3.

The trade-off results for the three controllers (i)–(iii) are presented in Figure 3.4, which shows the normalized LQG cost J/γ_0 versus the average sampling interval $\bar{h}$ for the three controllers. Similar to the DC motor example in Section 2.5 (cf. Figure 2.8), we note that the performance gain is

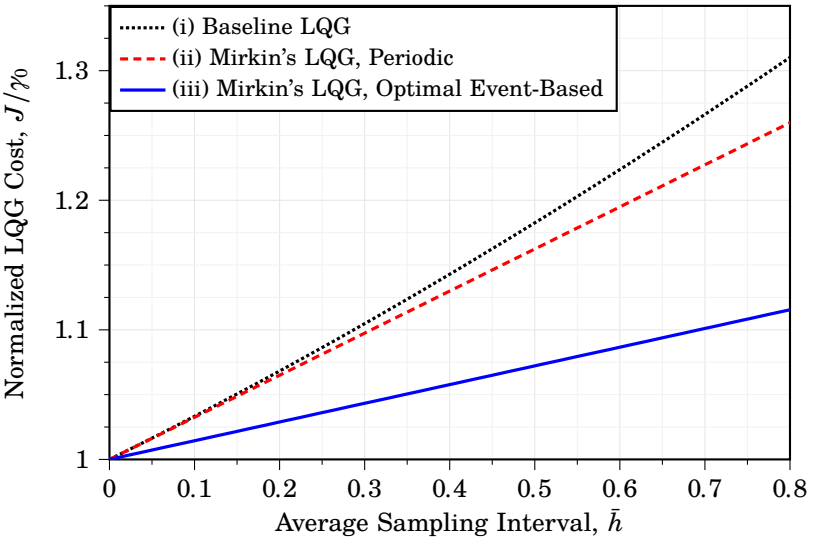

Figure 3.4 Normalized LQG cost J/γ_0 versus average sampling interval $\bar{h}$ for the multivariate integrator system in Example 3.2. Trade-off curves are shown for the baseline LQG controller (black, dotted) and Mirkin's LQG controller with periodic (red, dashed) and optimal event-based sampling (blue, solid).

larger when changing from periodic to optimal sampling (from (ii) to (iii)), compared to changing from suboptimal to optimal sampled-data structure of the controller (from (i) to (ii)). Over the considered interval in Figure 3.4, the cost J_μ for the baseline LQG controller is well-approximated by a linear curve, where a least-squares fit yields

$$J_{\text{base}}(\bar{h}) \approx 3.55\bar{h}.$$

As seen in (3.33) and (3.32), the corresponding curves for Mirkins LQG controller using periodic and optimal event-based sampling respectively are exactly linear. For periodic sampling (ii) we have

$$J_{\text{per}}(\bar{h}) = 3\bar{h}, \tag{3.35}$$

and for optimal event-based sampling we have

$$J_{\text{eb}}(\bar{h}) = \frac{4}{3}\bar{h}. \tag{3.36}$$

Comparing the cost for the same average sampling interval, we see that the cost ratio between the baseline and Mirkin's LQG controller with periodic

sampling is

$$\frac{J_{\text{base}}(\bar{h})}{J_{\text{per}}(\bar{h})} = \frac{3.55}{3} \approx 1.18. \tag{3.37}$$

From Theorem 3.6 we know that the corresponding cost ratio for Mirkin's LQG controller with periodic (ii) and optimal event-based sampling (iii) satisfies

$$1 + \frac{2}{n} = 2 \leq J_{\text{ratio}} < 3.$$

The exact ratio obtained from (3.35) and (3.36) is

$$J_{\text{ratio}} = \frac{9}{4} = 2.25,$$

which is almost twice as large as the ratio in (3.37).

Spiral Systems

We now consider a third class of systems which admits an exact solution to the OSP. We refer to this class as *spiral systems*, since their deterministic counterparts typically have spiral-shaped phase portraits. The considered spiral systems have matrices A, Q and R satisfying

$$A = aI + B, \quad RQ = \alpha I, \quad BR \text{ skew-symmetric}, \tag{3.38}$$

for some $a \in \mathbb{R}$, $\alpha > 0$ and $B \in \mathbb{R}^{n\times n}$. Our analysis of spiral systems was inspired by [Meng and Chen, 2012], where the authors considered the OSP for second-order systems of the form (3.38) with $B = 0$ and $R = Q = I$. As further analysis revealed, an exact solution can be derived for the somewhat more general class of systems satisfying (3.38), and for any system order n.

Solution. The key property of systems satisfying (3.38) is that the corresponding free boundary problem (3.10) can be reduced to a first-order problem. This gives the following result [Thelander Andrén and Cervin, 2020].

Theorem 3.7
For spiral systems defined by (3.38), the optimal sampling policy has the trigger function

$$g(\tilde{x}, \Delta_\rho) = \Delta_\rho - \left\|\tilde{x}\right\|_Q, \tag{3.39}$$

where

$$\Delta_\rho \coloneqq \arg\min_{\Delta} \Big\{ J_\mu(\Delta) + \rho f(\Delta) \Big\},$$

and $J_\mu(\Delta)$, $f(\Delta)$ are the LQG cost and average sampling rate obtained with $g(\tilde{x}, \Delta)$. Expressions for $J_\mu(\Delta)$ and $f(\Delta) = 1/\bar{h}(\Delta)$ are given by

$$J_\mu(\Delta) = \frac{2}{\alpha \bar{h}(\Delta)} \int_0^\Delta \int_0^y z^2 \Big(\frac{z}{y}\Big)^{n-1} e^{\frac{a}{\alpha}(z^2-y^2)} dz dy, \tag{3.40}$$

$$\bar{h}(\Delta) = \frac{2}{\alpha} \int_0^\Delta \int_0^y \Big(\frac{z}{y}\Big)^{n-1} e^{\frac{a}{\alpha}(z^2-y^2)} dz dy. \tag{3.41}$$

The corresponding solution pair (V, λ) to (3.10) is given by

$$\lambda = J_\mu(\Delta_\rho) + \rho f(\Delta_\rho),$$

$$V(\tilde{x}) = \begin{cases} \dfrac{2}{\alpha} \displaystyle\int_{\Delta_\rho}^{\|\tilde{x}\|_Q} \int_0^y (\lambda - z^2) \Big(\frac{z}{y}\Big)^{n-1} e^{\frac{a}{\alpha}(z^2-y^2)} dz dy, & \|\tilde{x}\|_Q \le \Delta_\rho, \\ 0, \quad \text{otherwise.} \end{cases}$$

Proof. The proof is given in Appendix C. □

Although the solution given by Theorem 3.7 relies on the reduction to a first-order problem, we emphasize that the new problem still has a non-trivial dependency on the state dimension n and differs from the problem solved in Theorem 3.3 (cf. (3.40)–(3.41) and (3.22)–(3.23)).

Example 3.3
Consider the following set of third-order systems defined by

$$\begin{aligned} A &= \begin{bmatrix} a & -1 & -2 \\ 1 & a & -3 \\ 2 & 3 & a \end{bmatrix}, \quad a \in \{-1,\ 0,\ 1\}, \\ B_u &= I_3, \\ C_y &= I_3, \\ B_w &= \begin{bmatrix} I_3 & 0_{3\times 3} \end{bmatrix}, \\ D_{yw} &= \begin{bmatrix} 0_{3\times 3} & I_3 \end{bmatrix}, \end{aligned}$$

i.e., a stable ($a = -1$), marginally stable ($a = 0$) and unstable ($a = 1$) system respectively). We consider an LQG control design for this set of systems specified by

$$C_z = \begin{bmatrix} I_3 \\ 0_{3\times 3} \end{bmatrix}, \quad D_{zu} = \begin{bmatrix} 0_{3\times 3} \\ I_3 \end{bmatrix},$$

resulting in

$$R = Q \propto I_3, \tag{3.42}$$

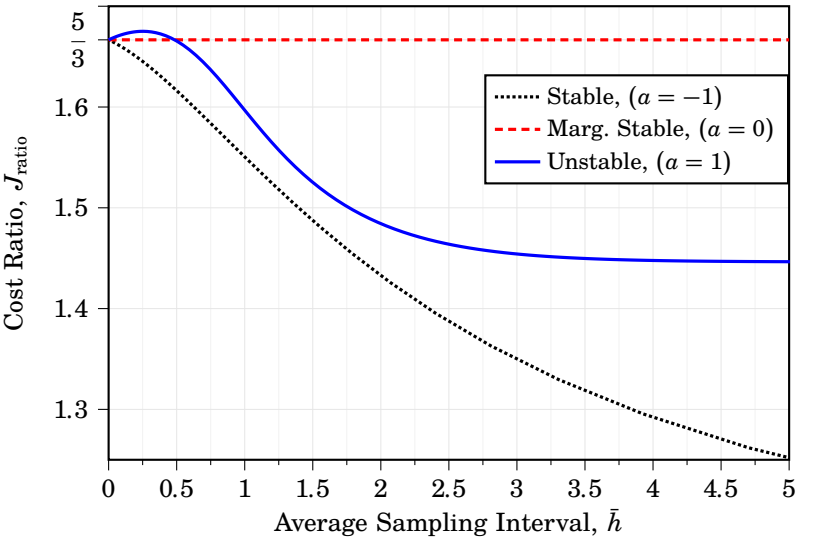

Figure 3.5 Cost ratio $J_{\text{ratio}}(\bar{h}) = J_{\text{per}}(\bar{h})/J_{\text{eb}}(\bar{h})$ between periodic and optimal event-based sampling over varying average sampling intervals $\bar{h}$ for the stable (black, dotted), marginally stable (red, dashed) and unstable (blue, solid) spiral systems in Example 3.3.

in all three cases. We note that the given design and plant model satisfy (3.38) and that Theorem 3.7 is applicable. Similar to the first-order example, we consider Mirkin's LQG controller in all three cases and compute the cost ratio J_{ratio} between periodic and optimal event-based sampling. Using (2.24) and (3.42), it is straightforward to show that the cost J_μ under periodic sampling in this example is given by

$$J_{\text{per}}(\bar{h}) = \begin{cases} \dfrac{cn}{2a}\left(\dfrac{e^{2a\bar{h}}-1}{2a\bar{h}} - 1\right), & a \neq 0, \\ \dfrac{cn\bar{h}}{2}, & a = 0. \end{cases}$$

Combined with numerical integration of (3.40) and (3.41), we can then compute J_{ratio}, see Figure 3.5.

In Figure 3.5, we see several similarities to the results for the first-order example (cf. Figure 3.2). We note that the cost ratio in the marginally stable case ($a = 0$) also is constant in this case, and it is straightforward to show that the system then satisfies Corollary 3.1. The solution to the OSP is then given by Theorem 3.4, and since the eigenvalues of the matrix RQ are all repeated we know that the ratio will equal the lower bound in Theorem 3.6,

i.e $J_{\text{ratio}} = 1 + 2/3 = 5/3$.

For the stable ($a = -1$) and unstable case ($a = 1$) we have $J_{\text{ratio}} \approx 5/3$ for small sampling intervals, and just like in the first-order example we see that J_{ratio} is monotonically decreasing in the stable case, while J_{ratio} is non-monotonic but still bounded in the unstable case. The peak cost ratio is $J_{\text{ratio}} \approx 1.68$ in the unstable case, and occurs for $\bar{h} \approx 0.25$. However, unlike the first-order case in Example 3.1, we see that the cost ratio then settles at a value smaller than $5/3$, i.e. smaller than the cost ratio achieved in the marginally stable case.

Single-Input and Single-Output Systems

Similar to the case of spiral systems, there are also certain types of single-input (SI) and single-output (SO) systems for which a solution can be obtained via a reduction to first-order dynamics. What enables the order reduction is the fact that the cost weight Q and covariance R are matrices of rank 1 in the SI and SO case, respectively. The cost and innovations process are therefore restricted to a single dimension in each respective case. This property was first explored for the SI case in [Molin et al., 2011].

Solution. Building upon [Molin et al., 2011], we have the following result for SI and SO systems [Thelander Andrén and Cervin, 2020].

Theorem 3.8
For SI systems satisfying

$$FAQ = FQA, \tag{3.43}$$

and SO systems satisfying

$$L^\intercal AR = L^\intercal RA, \tag{3.44}$$

the optimal admissible sampling policy has the trigger function

$$g(\tilde{x}, \Delta_\rho) = \Delta_\rho - |K\tilde{x}|, \tag{3.45}$$

where $K = F/\|F\|$ for the SI case, and $K = L^\intercal/\|L\|$ for the SO case. The parameter Δ_ρ is given by

$$\Delta_\rho \coloneqq \arg\min_{\Delta} \Big\{ J_\mu(\Delta) + \rho f(\Delta) \Big\},$$

where $J_\mu(\Delta)$, $f(\Delta)$ are the LQG cost and average sampling rate obtained with $g(\tilde{x}, \Delta)$, given by (3.22) and (3.23) with parameters exchanged in each case according to

$$\text{SI:} \quad a = \frac{FAF^\intercal}{\|F\|^2}, \quad r = \frac{FRF^\intercal}{\|F\|^2}, \quad q = \|Q\|, \tag{3.46}$$

$$\text{SO:} \quad a = \frac{L^\intercal AL}{\|L\|^2}, \quad r = \|R\|, \quad q = \frac{L^\intercal QL}{\|L\|^2}. \tag{3.47}$$

Proof. The proof is given in Appendix C. □

The conditions (3.43) and (3.44) in Theorem 3.8 can be quite restrictive. However, as noted in [Molin et al., 2011], there are systems that do not strictly satisfy these conditions but where the trigger function (3.45) still gives near-optimal performance. To give an indication on when this is the case, a measure $\delta \in [0,\ 1]$ was proposed in [Molin et al., 2011] for the SI case:

$$\delta := \frac{\|F(AQ - QA)\|}{\|A\|\|F\|\|Q\|}. \tag{3.48}$$

Exchanging F and Q in (3.48) for $L^\intercal$ and R respectively gives the corresponding measure for the SO case. The measure has the property that $\delta = 0$ iff the assumptions in Theorem 3.8 are satisfied, and a small value of δ suggests that the trigger function (3.45) is close to optimal. This is highlighted in the following example.

Example 3.4
Consider again the DC motor model from Example 2.1. With model and design parameters given by (2.30) and (2.31), we get

$$F = \begin{bmatrix} -1 & -1 \end{bmatrix}, \quad L = \begin{bmatrix} -3.58 & -6.42 \end{bmatrix}^\intercal,$$

and

$$Q = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 0.13 & 0.23 \\ 0.23 & 0.41 \end{bmatrix}.$$

Since the system is SISO, we can apply the measure (3.48) for both the SI (denoted δ_F) and SO (denoted δ_L) case. This gives

$$\delta_F = 0, \quad \delta_L = 0.13.$$

Theorem 3.8 is thus applicable to the SI case, which results in a first-order OSP with parameters given by (3.46). The parameters evaluate to

$$a = 0, \quad r = 0.5, \quad q = 2.$$

Since the reduced OSP is that of a first-order integrator, Theorem 3.6 tells us that the optimal sampling policy will outperform its periodic counterpart by a factor three, i.e, $J_{\text{ratio}} = 3$. This is indeed the case, as noted in Example 2.1.

Now consider the same DC motor model, but with a new LQG design where C_z is changed to

$$C_z = \begin{bmatrix} 4 & 2 \\ 2 & 4.5 \\ 0 & 0 \end{bmatrix}.$$

All other parameters from Example 2.1 are left unchanged. The new choice of C_z corresponds to a design with higher bandwidth of the closed-loop system while keeping the damping similar to that of the old design. With this new C_z, the values of L, R, and thus δ_L, remain unchanged, while F and Q become

$$F = \begin{bmatrix} -4.47 & -4.85 \end{bmatrix}, \quad Q = \begin{bmatrix} 20 & 21.68 \\ 21.68 & 23.5 \end{bmatrix},$$

which corresponds to

$$\delta_F = 0.03.$$

Theorem 3.8 is thus not directly applicable for the new design. However, since the value of δ_F is small we suspect that the trigger function (3.45) based on the SI case should still perform well. This is verified by the results in Figure 3.6, where the trade-off between normalized LQG cost J/γ_0 and average sampling interval $\bar{h}$ is shown for:

(i) the baseline LQG controller,

(ii) Mirkin's LQG controller with periodic sampling,

(iii) Mirkin's LQG controller with event-based sampling using the trigger function (3.45) from Theorem 3.8, and

(iv) Mirkin's LQG controller with optimal event-based sampling.

As hinted by the value of δ_F, we see that the performance of (iii) and (iv) are practically indistinguishable. The results in Figure 3.6 for (ii)–(iv), where Mirkin's LQG controller is used, are well-approximated by linear curves over the considered interval. A least-squares fit of the cost J_μ for periodic sampling (ii) yields

$$J_{\text{per}}(\bar{h}) \approx 10.91\bar{h}.$$

The corresponding approximation of (iii) and (iv), where event-based sampling is used, is given by

$$J_{\text{eb}}(\bar{h}) \approx 3.71\bar{h}.$$

For the new design, we thus have

$$J_{\text{ratio}} \approx \frac{10.91}{3.71} = 2.94,$$

which is still quite close to the factor three of the old design.

For the two event-based policies (iii) and (iv), the cost and average sampling interval were estimated using Monte Carlo simulations based on the optimal stopping formulation in Section 3.3. The optimal policy (iv) was approximated numerically using a *linear complementarity method* (see Section 4.3), which is one of the proposed numerical methods in this thesis that is described in detail in Chapter 4.

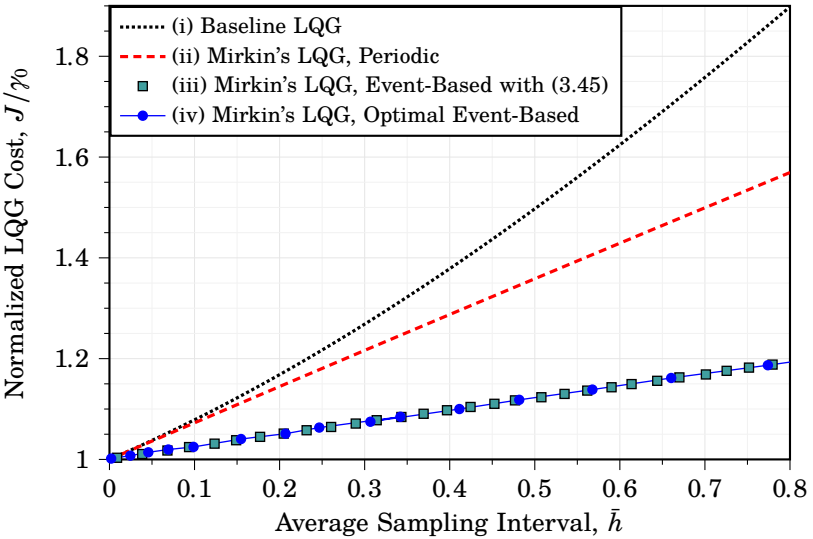

Figure 3.6 Normalized LQG cost J/γ_0 versus average sampling interval $\bar{h}$ for the DC motor system in Example 3.4 with modified C_z. Trade-off curves are shown for the baseline LQG controller (black, dotted) and Mirkin's LQG controller with periodic sampling (red, dashed), optimal event-based sampling (blue circles) and event-based sampling using the trigger function (3.45) (teal squares).

3.5 Summary

This chapter introduced the optimal sampling problem (OSP) for the LQG-optimal sampled-data controller (Mirkin's LQG controller). Optimizing the trade-off between LQG cost and the average sampling rate is an impulse control problem, characterized by a Hamilton–Jacobi–Bellman equation with free boundary. The free boundary constitutes an optimal threshold policy for sampling, and the optimal sampling policy is thus event-based by nature.

Four special system types with exact solutions to the OSP were presented. These include two types previously studied in the literature; first-order systems and multivariate integrator systems, where new results were presented for the latter in this chapter. In addition to these, two new system types were presented; spiral systems, and single-input and single-output systems of certain structure.

In all four cases, the optimal sampling policy can be represented as an ellipsoidal trigger function. This raises the question if this is a general trait of the optimal policy. To answer this question, we will consider numerical methods for solving general OSPs. This is the topic of the next chapter.

4

Numerical Methods

This chapter presents three numerical methods for solving the optimal sampling problem (OSP):

(i) value iteration,

(ii) a simulation-based method, and

(iii) a linear complementarity method.

Value iteration (i) is a standard tool in dynamic programming for solving optimal control problems (see, e.g., [Bertsekas, 2012]), and serves as a baseline for comparisons with the contributions of this chapter; the simulation-based method (ii) and linear complementarity method (iii).

The simulation-based method (ii) solves the OSP via a time-dependent reformulation of the stationary free boundary problem introduced in Chapter 3. The benefit of the time-dependent formulation is that its solution can be simulated using standard methods for partial differential equations (PDE). The stationary solution then coincides with the solution of the original problem.

The linear complementarity method (iii) is an alternative that avoids introduction of time-dependence by discretizing the stationary free boundary problem directly. The resulting finite-dimensional form can then be posed as a *linear complementarity problem*—a well-studied optimization problem that can be efficiently solved via quadratic programming.

While all three methods theoretically can provide arbitrarily fine approximations of the optimal solution, they all suffer from the curse of dimensionality. As a suboptimal, but useful, alternative for higher-order systems, we propose the use of ellipsoidal trigger functions, and present a brief outline on how optimal ellipse parameters can be obtained.

This chapter is structured as follows. An overview of each method (i)–(iii) is presented in Sections 4.1 to 4.3, respectively. In Section 4.4, all three methods are validated and compared in a numerical study. Following this

is another study in Section 4.5, where the linear complementarity method is used to explore the different solutions to the OSP for different classes of second-order systems. Based on the findings in Section 4.5, we further motivate the use of ellipsoidal trigger functions via an example in Section 4.6, and outline how optimal ellipse parameters can be obtained for higher-order systems. Finally, the chapter is summarized in Section 4.7.

4.1 Value Iteration

Value iteration is a central method in dynamic programming for solving optimal control problems, see e.g. [Bertsekas, 2012]. Based on the Bellman equation—the discrete-time analog to the HJB equation—the method produces successively improved estimates of the relative value function and minimal cost through a recursive scheme. Since value iteration assumes a problem in discrete time, we use a discrete-time approximation of the OSP for this method. Below we give an overview of the value iteration method along with the discretization of the OSP.

Discrete-Time Approximation of the OSP

In the original OSP, sampling decisions are made in continuous time. However, by restricting sampling decisions to some pre-defined sequence of points in time, $\{t_k\}$, we can formulate a discrete-time version of the OSP. Here, we consider the case of periodic decision times given by

$$t_k \coloneqq k\Delta t, \quad k \in \mathbb{N}_0,$$

for some (small) nominal period $\Delta t > 0$. The original problem is recovered as $\Delta t \to 0$, and for sufficiently small values of Δt we expect only minor differences.

Under the constraint of periodic decision times, a time-invariant stroboscopic version of the OSP is obtained via the same approach as described in Appendix A. The trajectory of $\tilde{x}$ at times $\{t_k\}$ is then described by the difference equation

$$\tilde{x}_{k+1} = (1 - s_k)A_{\Delta t}\tilde{x}_k + v_{\Delta t,k}, \tag{4.1}$$

where $\tilde{x}_k \coloneqq \tilde{x}(t_k)$. The binary variable $s_k \in \{0, 1\}$ in (4.1) represents the sampling decision at time t_k, where

$$s_k = \begin{cases} 1, & \text{if sampling,} \\ 0, & \text{otherwise.} \end{cases}$$

The random vector $v_{\Delta t,k}$ is given by [Åström and Wittenmark, 1997, Sec 10.6]

$$v_{\Delta t,k} = \int_{t_k}^{t_k+\Delta t} e^{(t_k+\Delta t-t)A} v(t)dt,$$

which corresponds to an i.i.d. sequence of zero-mean Gaussian random vectors. The matrix $A_{\Delta t}$ and noise covariance $\mathbb{E}[v_{\Delta t,k} v_{\Delta t,k}^\intercal] \coloneqq R_{\Delta t}$ are obtained from the original OSP as [Åström and Wittenmark, 1997, Sec 10.6]

$$\begin{aligned} A_{\Delta t} &= e^{\Delta t A}, \\ R_{\Delta t} &= \int_0^{\Delta t} e^{tA} R e^{tA^\intercal} dt. \end{aligned} \tag{4.2}$$

The constraint of periodic decision times also enables us to formulate the OSP cost J_{osp} in discrete-time as

$$J_{\text{osp}} = \gamma_v + \lim_{K\to\infty} \frac{1}{K\Delta t} \sum_{k=0}^{K-1} \mathbb{E}\Big[\big(1-s_k\big)\big\|\tilde{x}_k\big\|_{Q_{\Delta t}}^2 + \rho s_k\Big], \tag{4.3}$$

where

$$Q_{\Delta t} = \int_0^{\Delta t} e^{tA^\intercal} Q e^{tA} dt, \tag{4.4}$$

is the discrete-time version of the weight matrix Q (cf. (A.7)) and

$$\gamma_v = \frac{1}{\Delta t} \text{tr}\left(Q \int_0^{\Delta t} \int_0^t e^{\alpha A} R e^{\alpha A^\intercal} d\alpha dt \right), \tag{4.5}$$

is the accumulated cost from v during the intervals between sampling decisions (cf. (A.6)). We can thus equivalently consider the minimization of the discrete-time cost $J_{\text{osp},\Delta t}$, defined as

$$J_{\text{osp},\Delta t} \coloneqq \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E}\Big[\big(1-s_k\big)\big\|\tilde{x}_k\big\|_{Q_{\Delta t}}^2 + \rho s_k\Big], \tag{4.6}$$

and then retrieve the actual cost J_{osp} through the relation

$$J_{\text{osp}} = \gamma_v + \frac{J_{\text{osp},\Delta t}}{\Delta t}.$$

The Average-Cost Bellman Equation

With the discrete-time approximation of the OSP given by (4.1) and (4.6), we define the *Bellman operator* $\mathcal{T}(\cdot)$ applied to a function $V_{\Delta t} : \mathbb{R}^n \mapsto \mathbb{R}$ as

$$\mathcal{T}\big(V_{\Delta t}\big)(\tilde{x}) \coloneqq \min_{s\in\{0,1\}} \Big\{ \big(1-s\big)\|\tilde{x}\|_{Q_{\Delta t}}^2 + s\rho + \mathbb{E}\Big[V_{\Delta t}\big(\big(1-s\big)A_{\Delta t}\tilde{x} + v_{\Delta t}\big)\Big] \Big\}. \tag{4.7}$$

The Bellman operator corresponds to the optimization of the current sampling decision when the cost-to-go under the optimal policy is given by $V_{\Delta t}$. This operator is central to the *Bellman equation*, which is the discrete-time analog to the HJB equation (see Section 3.2). The Bellman equation for average-cost problems reads [Hernández-Lerma, 1989, Chapter 3, Theorem 2.2]

$$V_{\Delta t}(\tilde{x}) + \lambda_{\Delta t} = \mathcal{T}\big(V_{\Delta t}\big)(\tilde{x}), \tag{4.8}$$

for a scalar $\lambda_{\Delta t} > 0$. If a pair $(V_{\Delta t},\ \lambda_{\Delta t})$ are found satisfying the Bellman equation (4.8), then it is also a solution to the discrete-time OSP, with

$$\lambda_{\Delta t} = \min J_{\text{osp},\Delta t}.$$

The optimal sampling policy then corresponds to the action that achieves the minimum in the Bellman operator, i.e.,

$$s_k = \arg\min_{s\in\{0,1\}} \left\{(1-s)\|\tilde{x}_k\|^2_{Q_{\Delta t}} + s\rho + \mathbb{E}\Big[V_{\Delta t}\big((1-s)A_{\Delta t}\tilde{x}_k + v_{\Delta t}\big)\Big]\right\}.$$

With the notation from Chapter 3, the optimal policy can also be viewed as a static threshold policy of the form (3.6), with a trigger function given by (cf. Theorem 3.1)

$$g(\tilde{x}) = \rho + \mathbb{E}\big[V_{\Delta t}(v_{\Delta t}) - V_{\Delta t}(A_{\Delta t}\tilde{x} + v_{\Delta t})\big] - \|\tilde{x}\|^2_{Q_{\Delta t}}.$$

Existence of a solution to the Bellman equation (4.8) is guaranteed under some technical conditions [Hernández-Lerma, 1989, Chapter 3]. As described in [Molin, 2014, Section 6.3], these technical conditions are satisfied for the discrete-time OSP under the assumption that

$$\|\tilde{x}_k\| > \Delta \implies s_k = 1, \tag{4.9}$$

for some finite $\Delta > 0$, meaning that there exists a finite radius Δ, beyond which sampling is always triggered. In practice, adding the assumption (4.9) to the problem formulation is of minor consequence, since we can pick an arbitrarily large Δ.

Value Iteration

With the assumption in (4.9), it can be shown that (see [Molin, 2014, Section 6.3]) the following iterative scheme adapted from [Hernández-Lerma, 1989, Chapter 3, Theorem 4.8] converges to the solution pair $(V_{\Delta t},\ \lambda_{\Delta t})$:

$$k \in \mathbb{N}_0 : \begin{cases} \lambda^{k+1}_{\Delta t} = \mathcal{T}\big(V^k_{\Delta t}\big)(\tilde{x}_0), \\ V^{k+1}_{\Delta t}(\tilde{x}) = \mathcal{T}\big(V^k_{\Delta t}\big)(\tilde{x}) - \lambda^{k+1}_{\Delta t}. \end{cases} \tag{4.10}$$

The scheme in (4.10) defines the value iteration method, and convergence holds for any choice of $V_{\Delta t}^0$ and fixed $\tilde{x}_0$.

A benefit of value iteration is that it also provides bounds on $\lambda_{\Delta t}$. Defining

$$k \in \mathbb{N}_0 : \begin{cases} \bar{\lambda}_{\Delta t}^k \coloneqq \max\limits_{\tilde{x}} \Big(\mathcal{T}\big(V_{\Delta t}^{k-1}\big)(\tilde{x}) - V_{\Delta t}^{k-1}(\tilde{x}) \Big), \\ \underline{\lambda}_{\Delta t}^k \coloneqq \min\limits_{\tilde{x}} \Big(\mathcal{T}\big(V_{\Delta t}^{k-1}\big)(\tilde{x}) - V_{\Delta t}^{k-1}(\tilde{x}) \Big), \end{cases}$$

it holds that [Hernández-Lerma, 1989, Chapter 3, Theorem 4.8]

$$\underline{\lambda}_{\Delta t}^k \leq \lambda_{\Delta t} \leq \bar{\lambda}_{\Delta t}^k, \quad \forall k \in \mathbb{N}_0.$$

These bounds provide a useful stopping criterion for the iterative scheme in (4.10). By stopping whenever

$$\frac{\bar{\lambda}_{\Delta t}^k - \underline{\lambda}_{\Delta t}^k}{\underline{\lambda}_{\Delta t}^k} < \epsilon,$$

holds, we ensure that the relative error of the cost estimate $\lambda_{\Delta t}^k$ is less than ϵ, i.e.

$$\frac{|\lambda_{\Delta t}^k - \lambda_{\Delta t}|}{\lambda_{\Delta t}} \leq \frac{\bar{\lambda}_{\Delta t}^k - \underline{\lambda}_{\Delta t}^k}{\lambda_{\Delta t}} \leq \frac{\bar{\lambda}_{\Delta t}^k - \underline{\lambda}_{\Delta t}^k}{\underline{\lambda}_{\Delta t}^k} < \epsilon.$$

The evaluation of the Bellman operator typically requires some form of finite-dimensional approximation of $V_{\Delta t}$, e.g., gridding a bounded subset of the state space and approximate the dynamics (4.1) with a Markov decision process. For an approximation with N_{dim} grid points per state dimension n, each iteration in (4.10) has a nominal time complexity of $\mathcal{O}(N_{\text{dim}}^{2n})$. Since the method scales exponentially with n, it is only practical for lower-order systems.

4.2 A Simulation-Based Method

As noted in Section 4.1, value iteration does not explicitly use the free boundary formulation for solving the OSP. With the free boundary problem as a starting point, it is natural to consider a PDE solver method for the OSP. However, direct use of standard PDE solvers is inhibited by the fact that the boundary $\partial\Omega$ is unknown beforehand.

A way to circumvent this issue is to instead consider a time-dependent version of the free boundary problem, with a solution that coincides with the one of the original problem in stationarity. In this approach, we can choose any initial condition for the relative value function V—which corresponds to a *known* initial boundary—and proceed by simulating its evolution in time until a stationary solution is reached. This is the concept behind the proposed *simulation-based method*.

The Moving Boundary Problem

Recall the stationary free boundary problem (3.10), given as;

$$\forall \tilde{x} : \begin{cases} 0 = \Big(\mathcal{A}V(\tilde{x}) + \|\tilde{x}\|_Q^2 - \lambda\Big) \cdot \Big(- V(\tilde{x})\Big), \\ 0 \leq \mathcal{A}V(\tilde{x}) + \|\tilde{x}\|_Q^2 - \lambda, \\ 0 \leq -V(\tilde{x}). \end{cases} \tag{4.11}$$

Consider the case when the relative value function is time-dependent, i.e., $V := V(t, \tilde{x})$. A non-stationary version of (4.11)—referred to as a *moving boundary problem*—can then be posed as

$$\forall t \geq 0, \tilde{x} : \begin{cases} 0 = \Big(- \dfrac{\partial V}{\partial t}(t, \tilde{x}) + \mathcal{A}V(t, \tilde{x}) + \|\tilde{x}\|_Q^2 - \lambda\Big) \cdot \Big(- V(t, \tilde{x})\Big), \\ 0 \leq -\dfrac{\partial V}{\partial t}(t, \tilde{x}) + \mathcal{A}V(t, \tilde{x}) + \|\tilde{x}\|_Q^2 - \lambda, \\ 0 \leq -V(t, \tilde{x}), \\ \\ V(0, \tilde{x}) = V_0(\tilde{x}) \leq 0, \end{cases} \tag{4.12}$$

where $V_0 : \mathbb{R}^n \mapsto \mathbb{R}$ is an initial condition that can be chosen freely. Note that any stationary solution to the moving boundary problem (4.12) (where $\partial V / \partial t = 0$) also satisfies the original formulation in (4.11). The benefit of (4.12), however, is that we can choose $V_0(\tilde{x}) \leq 0$ freely and thus have a known solution at time $t = 0$. Using (4.12), we can then simulate V forward in time until we are sufficiently close to a stationary solution.

Simulating the Moving Boundary Problem

We proceed with a heuristic argument for how to simulate the solution of the moving boundary problem (4.12). Later, in Section 4.4, the method obtained from this argument is validated numerically.

For compactness, define

$$m(t, \tilde{x}) := \mathcal{A}V(t, \tilde{x}) + \|\tilde{x}\|_Q^2 - \lambda. \tag{4.13}$$

Now, consider the time-dependent version of V at some time point $t_k \geq 0$, and assume that

$$V(t_k, \tilde{x}) \leq 0, \quad \forall \tilde{x}.$$

Let $\bar{V}$ denote the solution of

$$\forall \Delta t \geq 0, \tilde{x} : \begin{cases} \dfrac{\partial \bar{V}}{\partial t}(t_k + \Delta t, \tilde{x}) = \bar{m}(t_k + \Delta t, \tilde{x}), \\ \\ \bar{V}(t_k, \tilde{x}) = V(t_k, \tilde{x}), \end{cases} \tag{4.14}$$

where, similar to (4.13), we have

$$\bar{m}(t, \tilde{x}) \coloneqq \mathcal{A}\bar{V}(t, \tilde{x}) + \|\tilde{x}\|_Q^2 - \lambda.$$

Note that the PDE in (4.14) corresponds to (4.12) without the moving boundary. To introduce the moving boundary, define

$$V(t_k + \Delta t, \tilde{x}) \coloneqq \min\big(\bar{V}(t_k + \Delta t, \tilde{x}),\ 0\big), \quad \forall \tilde{x},\ \Delta t \geq 0. \tag{4.15}$$

Since (4.15) implies

$$V(t_k + \Delta t, \tilde{x}) = \bar{V}(t_k + \Delta t, \tilde{x}), \quad \text{and/or} \quad V(t_k + \Delta t, \tilde{x}) = 0, \quad \forall \tilde{x},\ \Delta t \geq 0,$$

we conclude that the modified solution in (4.15) satisfies

$$\forall \Delta t \geq 0,\ \tilde{x} : \begin{cases} 0 = \Big(-\dfrac{\partial V}{\partial t}(t_k + \Delta t) + m(t_k + \Delta t, \tilde{x})\Big) \cdot \big(- V(t_k + \Delta t, \tilde{x})\big), \\ 0 \leq -V(t_k + \Delta t, \tilde{x}). \end{cases} \tag{4.16}$$

Comparing (4.16) with (4.12), we see that $V(t_k + \Delta t, \tilde{x})$ solves the moving boundary problem for $\Delta t \geq 0$ if it also satisfies

$$0 \leq -\frac{\partial V}{\partial t}(t_k + \Delta t, \tilde{x}) + m(t_k + \Delta t, \tilde{x}), \quad \forall \tilde{x},\ \Delta t \geq 0. \tag{4.17}$$

We proceed by analyzing this inequality. To this end, consider the first-order expansion of V;

$$\begin{aligned} V(t_k + \Delta t, \tilde{x}) &= V(t_k, \tilde{x}) + \int_0^{\Delta t} \frac{\partial V}{\partial t}(t_k + s, \tilde{x}) ds \\ &= V(t_k, \tilde{x}) + \frac{\partial V}{\partial t}(t_k + \Delta t, \tilde{x})\Delta t + \mathcal{O}(\Delta t^2) \quad \text{as } \Delta t \to 0, \end{aligned} \tag{4.18}$$

and the first-order expansion of $\bar{V}$;

$$\bar{V}(t_k + \Delta t, \tilde{x}) = V(t_k, \tilde{x}) + \frac{\partial \bar{V}}{\partial t}(t_k, \tilde{x})\Delta t + \mathcal{O}(\Delta t^2) \quad \text{as } \Delta t \to 0, \tag{4.19}$$

where we used that $\bar{V}(t_k, \tilde{x}) = V(t_k, \tilde{x})$. Furthermore, we note that

$$\begin{aligned} m(t_k + \Delta t, \tilde{x}) &= m(t_k, \tilde{x}) + \mathcal{O}(\Delta t) \quad \text{as } \Delta t \to 0 \\ &= \bar{m}(t_k, \tilde{x}) + \mathcal{O}(\Delta t) \quad \text{as } \Delta t \to 0, \end{aligned} \tag{4.20}$$

where, again, $\bar{V}(t_k, \tilde{x}) = V(t_k, \tilde{x})$ was used. From (4.15) and (4.18) to (4.20) we then get (as $\Delta t \to 0$)

$$\begin{aligned}
\frac{\partial V}{\partial t}(t_k + \Delta t, \tilde{x}) &= \frac{V(t_k + \Delta t, \tilde{x}) - V(t_k, \tilde{x}) - \mathcal{O}(\Delta t^2)}{\Delta t} \\
&= \frac{\min\left(\bar{V}(t_k + \Delta t, \tilde{x}), 0\right) - V(t_k, \tilde{x}) - \mathcal{O}(\Delta t^2)}{\Delta t} \\
&\leq \frac{\bar{V}(t_k + \Delta t, \tilde{x}) - V(t_k, \tilde{x}) - \mathcal{O}(\Delta t^2)}{\Delta t} \\
&= \frac{\partial \bar{V}}{\partial t}(t_k, \tilde{x}) + \mathcal{O}(\Delta t) \\
&= \bar{m}(t_k, \tilde{x}) + \mathcal{O}(\Delta t) \\
&= m(t_k + \Delta t, \tilde{x}) + \mathcal{O}(\Delta t).
\end{aligned}$$

In summary, we have thus shown (cf. (4.17))

$$0 \leq -\frac{\partial V}{\partial t}(t_k + \Delta t, \tilde{x}) + m(t_k + \Delta t, \tilde{x}) + \mathcal{O}(\Delta t) \quad \text{as } \Delta t \to 0. \tag{4.21}$$

The time evolution of V according to (4.14) and (4.15) therefore satisfies (4.16) and (4.21), which is identical to the moving boundary problem in (4.12), aside for an additional term of order Δt. For small Δt, this suggests that a simple iterative scheme based on (4.14) and (4.15) can be used to (approximately) simulate the moving boundary problem. The simulation-based method then constitutes of applying this iterative scheme until the solution is sufficiently close to stationarity, see Algorithm 1.

Algorithm 1: Simulation of the Moving Boundary Problem

Input: λ, Δt, $V_0(\tilde{x})$
Output: $V(\tilde{x})$ that (approximately) satisfies (4.11)
$t_k \leftarrow 0$
$V(t_k, \tilde{x}) \leftarrow V_0(\tilde{x})$
$\bar{V}(t_k, \tilde{x}) \leftarrow V_0(\tilde{x})$
while $\max_{\tilde{x}} |\partial V(t_k, \tilde{x})/\partial t| > \epsilon$ **do**
  Compute $\bar{V}(t_k + \Delta t, \tilde{x})$ by integrating (4.14)
  $t_k \leftarrow t_k + \Delta t$
  $V(t_k, \tilde{x}) \leftarrow \min(\bar{V}(t_k + \Delta t, \tilde{x}), 0)$
  $\bar{V}(t_k, \tilde{x}) \leftarrow V(t_k, \tilde{x})$
end
$V(\tilde{x}) \leftarrow V(t_k, \tilde{x})$
return $V(\tilde{x})$

Numerical implementation requires a finite-dimensional approximation of V. As in [Thelander Andrén et al., 2017], we use the backward time, centered space (BTCS) finite-difference scheme (see, e.g., [Hutchinson, 2015, Chapter 5]) in this thesis. The BTCS scheme is an implicit method, with a truncation error of order $\mathcal{O}(\Delta t) + \mathcal{O}(\Delta \tilde{x}^2)$, where $\Delta \tilde{x}$ denotes the spatial finite difference. With a spatial discretization of $N = N_{\text{dim}}^n$ mesh points (i.e., N_{dim} points per state dimension), the scheme first solves a linear system of N equations at the start of the simulation, followed by multiplication of an $N \times N$ matrix with an $N \times 1$ vector each iteration. The nominal time complexity per iteration of the BTCS scheme—and ultimately the complete method—is thus $\mathcal{O}(N^2) = \mathcal{O}(N_{\text{dim}}^{2n})$, i.e., the same as for value iteration.

4.3 A Linear Complementarity Method

The time discretization considered for value iteration and the simulation-based method contributes to the overall approximation error. Also, the required number of iterations to obtain an acceptable solution depends on the choice of the initial condition. It is therefore preferable to avoid a time-dependent formulation altogether and solve the stationary problem directly. To this end, we now consider a numerical method where the free boundary problem (3.10) is discretized and solved directly. The key for accomplishing this is that the finite-dimensional approximation of the free boundary problem can be posed as a *linear complementarity problem*—a form of quadratic program (QP)—for which there are several efficient numerical solvers [Murty, 1972]. Consequently, we refer to this method as the *linear complementarity method*.

The linear complementarity method in this thesis is inspired by similar methods for solving moving boundary problems in mathematical finance (see e.g. [Wilmott et al., 1995, Chapter 9]). However, this method has been adapted to the stationary problem of the OSP, with an additional proof of existence and uniqueness of the solution to the related linear complementarity problem.

Approximating the Relative Value Function

We consider a finite-dimensional approximation $\widehat{V}$ of the relative value function V in the form of a *radial basis function* (RBF) expansion;

$$\widehat{V}(\tilde{x}) = \sum_{j=1}^{N} \alpha_j \phi_j(\tilde{x}) = \alpha^\intercal \phi(\tilde{x}), \tag{4.22}$$

where

$$\alpha \coloneqq \begin{bmatrix} \alpha_1 & \dots & \alpha_j & \dots & \alpha_N \end{bmatrix}^\intercal,$$

is a vector of weights, and

$$\phi(\tilde{x}) \coloneqq \begin{bmatrix} \phi_1(\tilde{x}) & \dots & \phi_j(\tilde{x}) & \dots & \phi_N(\tilde{x}) \end{bmatrix}^\intercal,$$

is a vector of basis functions, $\phi_j : \mathbb{R}^n \mapsto \mathbb{R}$, where each function is radially symmetric around a *collocation point* $\tilde{x}_j$ in the state space. For a given set of N collocation points $\{\tilde{x}_j\}_{j=1,\dots,N}$, the objective is to find a weight vector α such that $\hat{V} \approx V$.

A benefit of the RBF approximation is that it is *mesh free*, meaning that we can choose the set of collocation points freely in the state space. With careful placement of the collocation points, an accurate solution can often be obtained even for relatively small numbers of collocation points [Fornberg and Flyer, 2015].

While there are many choices for basis functions, a popular choice, which we use here, are Gaussian basis functions:

$$\phi_j(\tilde{x}) \coloneqq \exp\Big(-c\|\tilde{x} - \tilde{x}_j\|_2^2 \Big), \quad j = 1, \dots, N. \tag{4.23}$$

The parameter $c > 0$ is a *shape parameter* that determines the decay rate of the basis functions. It is typically chosen as a trade-off between accuracy and numerical stability, where a small value of c often gives improved accuracy at the price of ill-conditioning [Fornberg and Flyer, 2015].

With a Gaussian RBF expansion given by (4.22) and (4.23), the gradient and Hessian of $\hat{V}$ are given by

$$\begin{aligned}
\nabla_{\tilde{x}} \hat{V} &= \sum_{j=1}^{N} \alpha_j \nabla \phi_j(\tilde{x}) = -2c \sum_{j=1}^{N} \alpha_j \big(\tilde{x} - \tilde{x}_j\big) \phi_j(\tilde{x}), \\
\nabla_{\tilde{x}}^2 \hat{V} &= \sum_{j=1}^{N} \alpha_j \nabla^2 \phi_j(\tilde{x}) = 2c \sum_{j=1}^{N} \alpha_j \big(2c(\tilde{x} - \tilde{x}_j)(\tilde{x} - \tilde{x}_j)^\intercal - I_n\big) \phi_j(\tilde{x}).
\end{aligned}$$

Direct insertion of $\hat{V}$ into the free boundary problem (4.11) thus yields

$$\begin{cases}
0 = \big(\alpha^\intercal \Lambda(\tilde{x}) \phi(\tilde{x}) + \|\tilde{x}\|_Q^2 - \lambda\big) \cdot \big(- \alpha^\intercal \phi(\tilde{x})\big), \\
0 \leq \alpha^\intercal \Lambda(\tilde{x}) \phi(\tilde{x}) + \|\tilde{x}\|_Q^2 - \lambda, \\
0 \leq -\alpha^\intercal \phi(\tilde{x}).
\end{cases} \tag{4.24}$$

where $\Lambda : \mathbb{R}^n \mapsto \mathbb{R}^{N \times N}$ is a diagonal matrix corresponding to the infinitesimal generator, with the j^{th} diagonal element given by

$$\Lambda_j(\tilde{x}) \coloneqq c\big(2c\|\tilde{x} - \tilde{x}_j\|_R^2 - 2\tilde{x}^\intercal A^\intercal (\tilde{x} - \tilde{x}_j) - \operatorname{tr}(R)\big), \quad j = 1, \dots, N.$$

If the conditions in (4.24) are satisfied for all $\tilde{x}$, then $\hat{V}$ is an exact solution to the free boundary problem (4.11). However, the task of finding a weight

vector α such that (4.24) holds for all $\tilde{x}$ is infinite-dimensional, and can not be considered in practice. Instead, we search for a weight vector α such that (4.24) holds point-wise at all collocation points $\{\tilde{x}_j\}$. This approximation and the resulting linear complementarity problem are presented next.

The Linear Complementarity Problem

The condition that (4.24) should hold point-wise at all collocation points results in a system of N equations subject to $2N$ inequalities. To enable a compact representation, define the vector $\beta \in \mathbb{R}^N$ and the *interpolation matrix* $\Phi \in \mathbb{R}^{N\times N}$ as:

$$\beta \coloneqq \begin{bmatrix} \|\tilde{x}_1\|_Q^2 - \lambda \\ \vdots \\ \|\tilde{x}_N\|_Q^2 - \lambda \end{bmatrix}^\intercal, \quad \Phi \coloneqq \begin{bmatrix} \phi_1(\tilde{x}_1) & \phi_2(\tilde{x}_1) & \dots & \phi_N(\tilde{x}_1) \\ \phi_1(\tilde{x}_2) & \phi_2(\tilde{x}_2) & & \vdots \\ \vdots & & \ddots & \\ \phi_1(x_N) & \dots & & \phi_N(x_N) \end{bmatrix}.$$

Due to the choice of Gaussian basis functions, we have $\Phi = \Phi^\intercal \succ 0$ [Fornberg and Flyer, 2015]. Additionally, define the matrix

$$\Psi \coloneqq \bar{\Lambda} \odot \Phi,$$

where $\odot$ denotes the Hadamard (element-wise) product, and the matrix $\bar{\Lambda} \in \mathbb{R}^{N\times N}$ is given by

$$\bar{\Lambda} \coloneqq \begin{bmatrix} \Lambda_1(\tilde{x}_1) & \Lambda_2(\tilde{x}_1) & \dots & \Lambda_N(\tilde{x}_1) \\ \Lambda_1(\tilde{x}_2) & \Lambda_2(\tilde{x}_2) & & \vdots \\ \vdots & & \ddots & \\ \Lambda_1(\tilde{x}_N) & \dots & & \Lambda_N(\tilde{x}_N) \end{bmatrix}.$$

The condition that (4.24) should hold point-wise at all collocation points can then be compactly written as

$$\forall j = 1, \dots, N : \begin{cases} 0 = \big(\Psi\alpha + \beta\big)_j \cdot (-\Phi\alpha)_j, \\ 0 \le \big(\Psi\alpha + \beta\big)_j, \\ 0 \le \big(-\Phi\alpha\big)_j, \end{cases} \tag{4.25}$$

where $(\cdot)_j$ denotes the j^{th} vector element. The relations in (4.25) can be equivalently written in the following quadratic form:

$$\begin{cases} 0 = (-\Phi\alpha)^\intercal\big(\Psi\alpha + \beta\big), \\ 0 \le \Psi\alpha + \beta, \\ 0 \le -\Phi\alpha, \end{cases} \tag{4.26}$$

where the inequalities are considered element-wise. Now, by defining

$$z \coloneqq -\Phi\alpha,$$
$$M \coloneqq -\Psi\Phi^{-1},$$

we can rewrite (4.26) as

$$\begin{cases} 0 = z^\intercal(Mz+\beta), \\ 0 \leq Mz+\beta, \\ 0 \leq z, \end{cases} \tag{4.27}$$

which is a linear complementarity problem in the vector $z \in \mathbb{R}^N$. This problem is in turn equivalent to the following QP [Murty, 1972]:

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & z^\intercal Mz + \beta^\intercal z, \\ \text{subject to} \quad & 0 \leq Mz+\beta, \\ & 0 \leq z, \end{aligned} \tag{4.28}$$

where the minimum objective equals zero. The problem (4.28) can be solved using any QP solver, and after obtaining a solution $z^\star$ we retrieve the approximation $\hat{V}$ as

$$\hat{V}(\tilde{x}) = -\phi(\tilde{x})^\intercal\Phi^{-1}z^\star.$$

Assuming an RBF approximation with $N = N_{\text{dim}}^n$ collocation points (N_{dim} points per state dimension n), a QP solver using a first or second-order method nominally requires $\mathcal{O}\big(N_{\text{dim}}^{2n}\big)$ or $\mathcal{O}\big(N_{\text{dim}}^{3n}\big)$ operations per iteration respectively when solving (4.28). Similarly to the previous methods, the linear complementarity method thus also scales exponentially with n, and a QP solver that specializes in large-scale problems is typically most efficient when solving (4.28). The computational burden can be reduced by picking fewer but more carefully placed collocation points. However, such techniques are outside the scope of this thesis.

Existence and Uniqueness of Solution

A benefit of the linear complementarity method is that we can give an explicit condition for when the numerical problem of finding $\hat{V}$ is well-posed, i.e., when there exists a unique solution to (4.28). Central to the condition is the notion of a *P-matrix*:

Definition 4.1—P-matrix
A P-matrix is a matrix with only positive principal minors. ♢

Note that a P-matrix is not necessarily symmetric, but in the case of symmetry it is by definition also positive definite. The condition for well-posedness is then straightforward [Murty, 1972, Theorem 4.2]:

THEOREM 4.1
The linear complementarity problem (4.27) has a unique solution for each $\beta \in \mathbb{R}^N$ iff M is a P-matrix.

Proof. See [Murty, 1972, Theorem 4.2] □

The question then arises whether M in (4.27) is a P-matrix or not. The following result sheds some light on this:

THEOREM 4.2
Assume that $R \neq 0$. Then, for any given set of unique collocation points $\{\tilde{x}_j\}$, there exists a finite constant $\underline{c} > 0$ such that for any shape parameter $c > \underline{c}$ the corresponding matrix M in (4.27) is a P-matrix.

Proof. The proof is given in Appendix C. □

Theorem 4.2 implies that a well-posed problem can always be obtained by making the shape parameter c sufficiently large. However, the lower bound $\underline{c}$ is not given explicitly, and we therefore need to verify if M is a P-matrix or not for a given c—and increase the value of c if it is not.

Unfortunately, determining if a matrix is a P-matrix or not is an NP-hard problem, and straightforward evaluation of the $2^N - 1$ principal minors requires $\mathcal{O}(N^3 2^N)$ operations [Rump, 2003]. This verification can thus only be performed in practice when the number of collocation points N is small. However, there exist several sufficient conditions of less computational complexity that can be used. For instance, M is a P-matrix if $M + M^\intercal \succ 0$, which requires $\mathcal{O}(N^3)$ operations to verify. In all examples where the linear complementarity method is used in this thesis, we use this sufficient condition combined with a simple binary search to find as small shape parameter c as possible for which the problem is guaranteed to be well-posed. Additional sufficient conditions can be found in [Tsatsomeros, 2002].

4.4 Numerical Validation and Performance Comparison

We now proceed by validating the three presented methods numerically against a set of examples with known solutions. To measure the accuracy of each method, we track the normalized maximum absolute error (MAE),

$$\max_{\tilde{x}} \frac{|V(\tilde{x}) - \hat{V}_N(\tilde{x})|}{\rho},$$

between the exact solution V and the approximation $\hat{V}_N$ obtained using N discretization points (i.e., mesh or collocation points). The normalization is motivated by the fact that $V(\tilde{x}) \in [-\rho, 0]$ for all $\tilde{x}$, and the resulting

measure is an estimate of the relative error. In addition to verifying that the MAE is decreasing with increasing N for all three methods, we are also interested in comparing performance in terms of

(i) the accuracy obtained for a given number of discretization points N,

(ii) the accuracy obtained for a given solver time, and

(iii) the trade-off between accuracy and solver time.

Setup

The set of examples considered in the validation consists of second-order multivariate integrator systems, whose exact solution to the OSP was presented in Section 3.4. The OSP for this type of system always satisfies $R \succ 0$ and $Q \succ 0$, and thus there always exists a linear transformation such that they can be formulated as

$$\begin{aligned} A &= 0_{2\times 2}, \\ R &= I_2, \\ Q &= \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix}, \end{aligned} \tag{4.29}$$

for some $q_1 > 0$ and $q_2 > 0$. Without loss of generality, we can thus consider the example systems to be of the form in (4.29), where the corresponding OSP is uniquely determined by the three parameters q_1, q_2 and ρ. For the validation, we pick 100 systems of this form with randomly selected parameters in the ranges $\rho \in [0.01,\ 1]$ and $q_1, q_2 \in [1,\ 10]$.

For each of these 100 systems, we apply the three numerical methods with $N = 5^2,\ 6^2,\ \ldots,\ 60^2$ uniformly distributed discretization points. The two time-dependent methods (value iteration and the simulation-based method) use a fixed time step $\Delta t = 2.5 \cdot 10^{-3}$, and share the same stopping condition

$$\frac{|\hat{V}_N(t_k, \tilde{x}) - \hat{V}_N(t_{k-1}, \tilde{x})|}{\Delta t} < 10^{-6}.$$

The operator splitting quadratic program (OSQP) solver [Stellato et al., 2020] is used to solve the QP arising in the linear complementarity method. All three methods are implemented in the Julia programming language [Bezanson et al., 2017] and evaluated on a PC with a quad-core Intel i7-7500 processor and 16 GB RAM.

Results and Discussion

The normalized MAE for varying numbers of discretization points N are presented in Figure 4.1. A numerical method is said to have a convergence rate of order p if it satisfies

$$|V(\tilde{x}) - \hat{V}_N(\tilde{x})| \leq CN^{-p},$$

for some constant $C > 0$. In Figure 4.1, a least-squares fit of the curve for the linear complementarity method give an estimated convergence rate of $p = 1$. For $N \leq 200$, a similar convergence rate is observed for the simulation-based method, while the MAE of the value iteration method remains large and almost unaffected. However, for $N > 200$, the value iteration method starts to converge rapidly until finally settling at a constant normalized MAE value of roughly $6 \cdot 10^{-3}$. Similarly, the simulation-based method also settles for large N at roughly $1.1 \cdot 10^{-2}$—approximately twice as large.

The convergence to a constant MAE observed for the simulation-based method and value iteration is due to their time discretization, which produces a constant error for a fixed time step Δt that becomes dominant for large N. This error can be reduced by decreasing Δt, but at the price of more iterations and therefore longer solver times to obtain a solution. The linear complementarity method avoids this dilemma, and generally achieves better accuracy for all N compared to both time-dependent methods.

In addition to Figure 4.1, we also present the normalized MAE versus the average time to compute a solution in Figure 4.2. Here we see that the linear complementarity method generally outperforms both the time-dependent methods also in terms of solver time. We also note that the simulation-based method is generally more efficient than value iteration for shorter solver times, but is then outperformed by value iteration since the simulation-based method settles at a larger MAE.

We noted earlier that the simulation-based method settles at a larger MAE value than value iteration for large N. While this initial observation favors value iteration, it does not take solver times into account. For instance, if value iteration gives half the MAE but takes, e.g., ten times longer to run, then the simulation-based method has a better trade-off between accuracy and solver time and is arguably more useful. To this end, we present the results in Figure 4.3 where the ratio of median MAE versus the ratio of average solver time between all three methods is shown. The results show that the simulation-based method indeed has a better trade-off compared to value iteration. The corresponding comparison between the linear complementarity and simulation-based method is inconclusive, with a roughly equal division of data points favoring each method. Finally, the results in Figure 4.3 suggest that the linear complementarity method has an equal or better trade-off compared to value iteration.

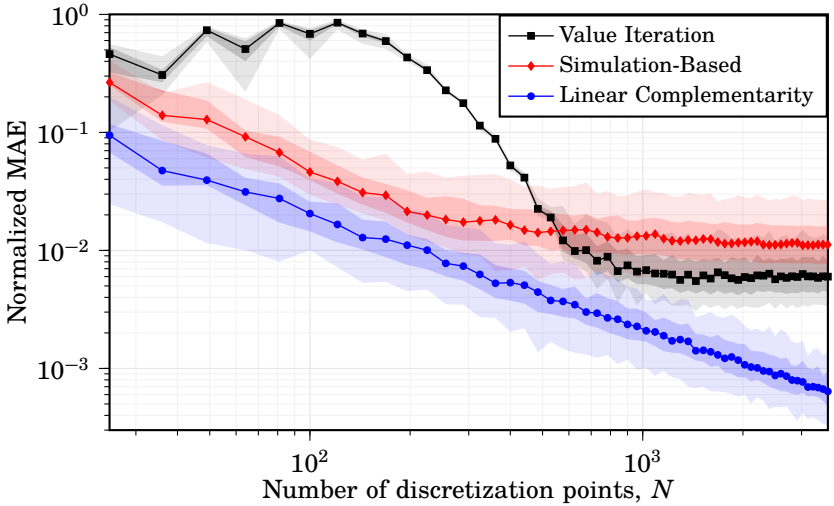

Figure 4.1 Normalized MAE versus number of discretization points N for value iteration (black squares), the simulation-based method (red diamonds) and the linear complementarity method (blue circles). For each method, the markers represents median values, the darker shaded area represents the first and third quartiles (containing 50% of the data) and the lighter shaded area represents the first and fourth quartiles (containing all the data).

Conclusions

The conclusions of the previous section are summarized as follows:

- The numerical accuracy of all three methods improves with the number of discretization points N.
- The linear complementarity method has an estimated convergence rate of order $p = 1$ and outperforms the other two methods in terms of numerical accuracy for both a given number of discretization points N and solver time. Furthermore, it has a similar or better trade-off between accuracy and solver time compared to value iteration.
- Due to their time discretizations, value iteration and the simulation-based method asymptotically approach constant accuracy for large N.
- The simulation-based method has a better trade-off between accuracy and solver time compared to value iteration, arguably making it the better option of the two.

Based on these conclusions, we consider the linear complementarity method the overall best option of the three methods.

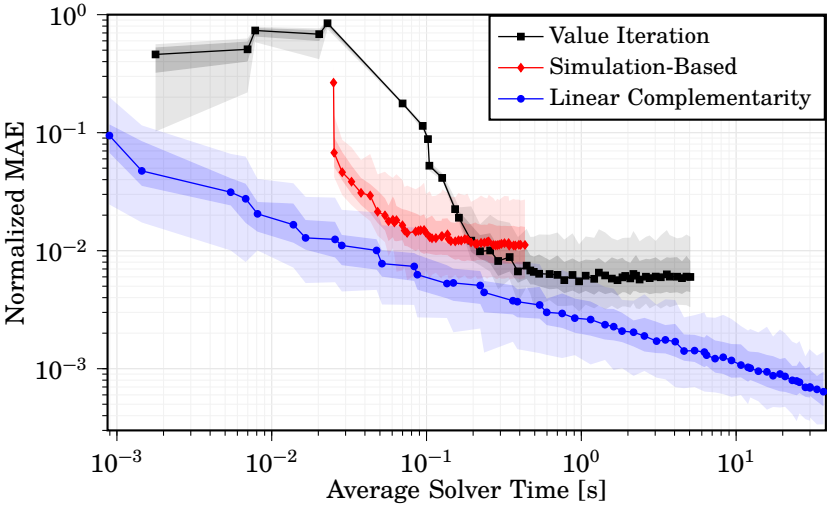

Figure 4.2 Normalized MAE versus average solver time for value iteration (black squares), the simulation-based method (red diamonds) and the linear complementarity method (blue circles). For each method, the markers represents median values, the darker shaded area represents the first and third quartiles (containing 50% of the data) and the lighter shaded area represents the first and fourth quartiles (containing all the data).

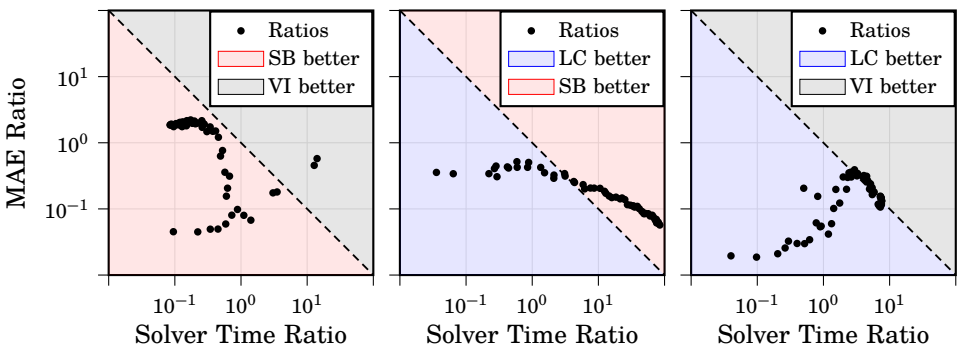

Figure 4.3 Ratio of median MAE versus ratio of average solver time between the simulation-based (SB) method and value iteration (VI) (left), the linear complementarity (LC) and SB method (mid) and the LC method and VI (right), respectively. Ratios (black circles) in the red, gray and blue shaded areas correspond to a better trade-off between accuracy and solver time for the simulation-based method, value iteration and the linear complementarity method, respectively.

4.5 Optimal Policies for Second-Order Systems

With the presented numerical methods we can investigate how different system dynamics (i.e, A-matrices) affect the shape of the threshold $\partial\Omega$ of the optimal sampling policy. As concluded in Section 3.5, the optimal sampling policy in all presented special cases can be represented with ellipsoidal trigger functions. This raises the question if this also holds in general, and if not, what other shapes the optimal threshold $\partial\Omega$ can take.

To investigate this, we solve the OSP numerically for a set of systems with different A-matrices. Since all three numerical methods scale poorly with the system order n, we restrict the investigation to second-order systems. The choice of systems is based on the different classes of equilibrium points for linear systems, given by:

Class	A-matrix	Poles
Saddle	$\begin{bmatrix} 0 & 15 \\ 15 & 0 \end{bmatrix}$	± 15
Node	$\begin{bmatrix} 15 & 10 \\ 10 & 15 \end{bmatrix}$	5, 25
Center	$\begin{bmatrix} 0 & -18 \\ 12.5 & 0 \end{bmatrix}$	$\pm 15i$
Focus	$\frac{1}{\sqrt{2}}\begin{bmatrix} 15 & -22.5 \\ 10 & 15 \end{bmatrix}$	$\frac{1}{\sqrt{2}}(15 \pm 15i)$
Double integrator	$\begin{bmatrix} 0 & 5 \\ 0 & 0 \end{bmatrix}$	0, 0

In all cases, we restrict the investigation to $R = Q = I_2$ and $\lambda = 1$, as we are specifically interested in the impact of the A-matrix.

Remark 4.1
Note that we have not included a system with a star-type equilibrium point, i.e., a system with with an A-matrix of the form

$$A = aI_2, \quad a \in \mathbb{R}. \tag{4.30}$$

This is because (4.30) together with the assumption $R = Q = I_2$ satisfies the definition in (3.38) of a *spiral system*. By Theorem 3.7 we thus already know that its optimal sampling policy has an ellipsoidal trigger function.$\diamondsuit$

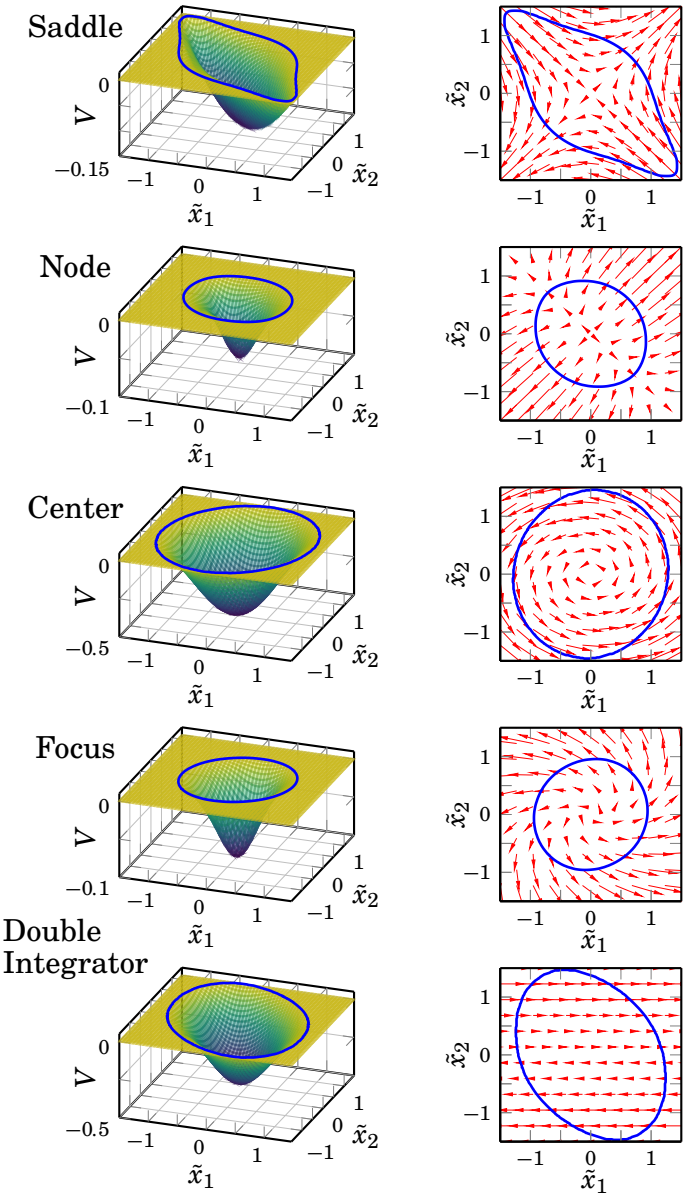

Figure 4.4 Numerical solutions of the OSP for different classes of second-order systems, with the relative value functions V (surface) in the left column, and the corresponding optimal threshold $\partial\Omega$ (blue, solid) with the system's phase portrait (red arrows) in the right column.

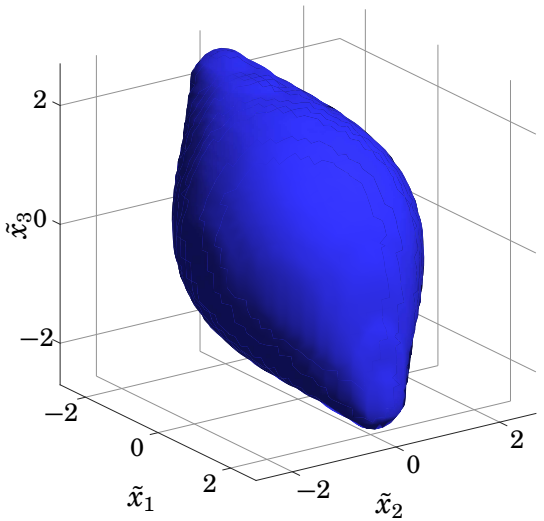

Figure 4.5 Optimal threshold $\partial\Omega$ (blue surface) for a third-order saddle-point system, with $A = [0\,0\,5; 0\,5\,0; 5\,0\,0]$, $R = Q = I_3$ and $\lambda = 4$. The solution was computed using the linear complementarity method with $30 \times 30 \times 30 = 27\,000$ uniformly distributed collocation points.

Using the linear complementarity method from Section 4.3, we compute the relative value function V for the considered systems and retrieve the corresponding optimal threshold $\partial\Omega$. In all cases, the solution is computed using $40 \times 40 = 1\,600$ collocation points, distributed on a uniform grid over the domain $\tilde{x}_1, \tilde{x}_2 \in [-1.5, 1.5]$. The results are presented in Figure 4.4, where the (approximate) relative value function V and the corresponding threshold $\partial\Omega$ for each system is shown. The phase portrait of each system is also shown for reference.

The results in Figure 4.4 clearly invalidate the hypothesis of ellipsoidal thresholds in the general case. However, most of the thresholds seem to be well-approximated by ellipses. This is a promising observation for the design of suboptimal but simpler event-based sampling policies, and by using a trigger function with a simple ellipsoidal parametrization we can expect near-optimal performance in most cases.

The notable exception in Figure 4.4 is the saddle-point system, which has a non-convex threshold $\partial\Omega$. This curious result also appears in higher-order systems, as demonstrated by the third-order example in Figure 4.5. While the threshold itself has a radically different shape, it remains to be quantified how performance-critical the exact shape really is.

4.6 On Ellipsoidal Trigger Functions

In Section 3.4, we observed that all considered special cases has an optimal policy that can be represented with an ellipsoidal trigger function of the form

$$g(\tilde{x}) = \Delta - \|\tilde{x}\|_P, \tag{4.31}$$

for some scalar $\Delta > 0$ and matrix $P \succeq 0$. Furthermore, most of the problem classes considered in the numerical study in Section 4.5 were shown to have near-ellipsoidal optimal thresholds. These observations suggest that ellipsoidal trigger functions can be a simple, but useful, alternative to the optimal solution.

A critical benefit of ellipsoidal trigger functions is their simple parametrization, with a parameter space that grows only quadratically (rather than exponentially) with the system order. This stands in contrast to the previously described numerical methods in this chapter, which in principle can provide arbitrarily fine approximations of the optimal solution, but suffer from the curse of dimensionality. However, to find optimal ellipse parameters for higher-order systems we require an efficient method for evaluating and differentiating the LQG cost and average sampling period. One promising method is the Monte Carlo approach for optimal stopping problems proposed in [Becker et al., 2019], which we briefly review here and evaluate in the context of the OSP. First, however, we address the notable exception found in Section 4.5; saddle-point systems, which have non-convex optimal thresholds.

Example 4.1—Ellipsoidal Trigger Function for a Saddle-Point System Consider the following LQG design problem for an unstable second-order system, with parameters:

$$\begin{aligned} A &= \begin{bmatrix} 0 & 15 \\ 15 & 0 \end{bmatrix}, \\ B_w = C_z^\intercal &= \begin{bmatrix} 3.35 & 0 & 0 & 0 \\ -3.27 & 0.72 & 0 & 0 \end{bmatrix}, \\ B_u = C_y^\intercal &= \begin{bmatrix} 28.71 & 0 \\ 28.64 & 2 \end{bmatrix}, \\ D_{zu} = D_{yw}^\intercal &= \begin{bmatrix} 0_2 \\ I_2 \end{bmatrix}. \end{aligned} \tag{4.32}$$

The corresponding OSP has the parameters

$$Q = R = I_2,$$

i.e., the OSP obtained for the LQG design problem in (4.32) is identical to that considered for the saddle-point system in Section 4.5. For reference, the

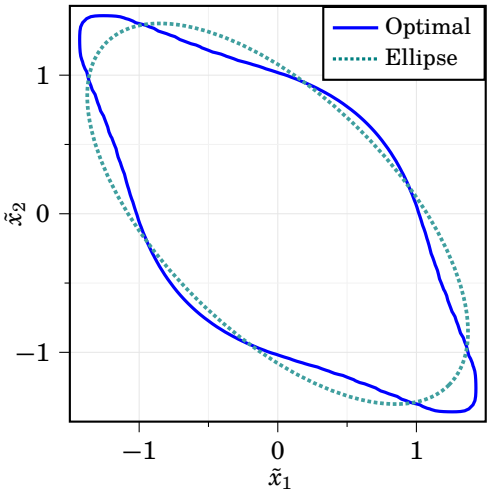

Figure 4.6 The optimal sampling threshold (blue, solid) for the saddle-point system in Example 4.1. The ellipse (teal, dotted) has been fitted to the optimal threshold using the least-squares method.

optimal threshold is shown in Figure 4.6, which also shows an ellipse fitted to the optimal threshold using least squares. While the ellipse is similarly oriented as the optimal threshold, the fit is overall quite poor. Given a threshold policy based on this naive ellipse approximation, we consider how it performs in comparison to both periodic sampling and the optimal event-based policy.

Performance is evaluated by computing the cost J_μ for varying average sampling intervals $\bar{h}$ for each sampling policy. Data for periodic sampling are obtained via (2.24), while Monte Carlo simulations based on the optimal stopping formulation in Section 3.3 with a time step of $\Delta t = 10^{-4}$ are used for the optimal and ellipsoidal threshold policies.

The trade-off results are presented in Figure 4.7, where the cost J_μ is normalized by γ_0. First, we observe that both event-based policies clearly outperform periodic sampling, with a gap in the LQG cost that increases with $\bar{h}$ (note the logarithmic scale in Figure 4.7). Most importantly, however, we note that the performance of the ellipse threshold policy and the optimal policy are practically identical. That is, even when the optimal policy is non-convex, a simple ellipsoidal threshold performs remarkably well. Of course, the naive approach used to obtain the ellipsoidal threshold in this example is not viable in general as it requires the optimal threshold.

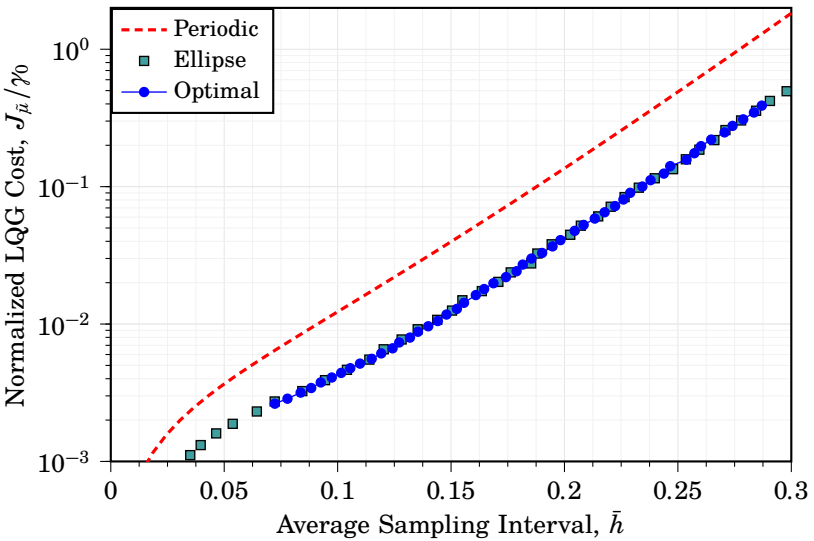

Figure 4.7 Normalized LQG cost J_μ/γ_0 versus average sampling interval $\bar{h}$ for the saddle-point system in Example 4.1. While periodic sampling (red, dashed) is clearly outperformed by event-based sampling, the difference is negligible between sampling based on the optimal policy (blue circles) and the ellipsoidal threshold policy (teal squares).

Parameter Optimization via Monte Carlo

Based on the previous example, we note that;

(i) an ellipsoidal trigger function can give near-optimal performance even for saddle-point systems, and

(ii) performance seems quite robust in terms of ellipse parameters, due to the heuristic and quite arbitrary method with which they were obtained.

These two observations further motivate the use of ellipsoidal trigger functions, especially for higher-order systems where numerical approximations of the optimal solution break down. Additionally, the second observation is promising when it comes to parameter optimization, as it suggests that approximate methods might be sufficient. One such approach is to consider parameter optimization based on Monte Carlo estimates of the LQG cost and average sampling interval. Below, we give an outline of how a Monte Carlo method can be implemented for the OSP based on [Becker et al., 2019].

Method Overview. Similar to the value iteration method, the Monte Carlo approach requires a problem in discrete time. Therefore, we consider here the discrete-time approximation of the OSP used for the value iteration method, see Section 4.1.

Optimization in the Monte Carlo approach is based on the optimal stopping formulation of the OSP described in Section 3.3. Using the approach described in Appendix A, a discrete-time version of the OSP cost J_{osp} in optimal stopping form is obtained as

$$J_{\text{osp}} = \gamma_v + \frac{\mathbb{E}\Big[\sum_{k=0}^{\tau} \|\tilde{x}_k\|^2_{Q_{\Delta t}}\Big] + \rho}{\Delta t \mathbb{E}[\tau]} \coloneqq \gamma_v + \frac{J_{\text{osp},\Delta t}}{\Delta t},$$

where Δt is the nominal time step of the discretization, and $Q_{\Delta t}$ and γ_v are given by (4.5) and (4.4) respectively. Furthermore,

$$\tau \coloneqq \min\Big\{k : s(\tilde{x}_k) = 1\Big\},$$

is the index of the first stopping time (i.e., first sampling time) for a given threshold policy, which assigns a stopping decision $s : \mathbb{R}^n \mapsto \{0, 1\}$ as

$$s(\tilde{x}) \coloneqq \begin{cases} 1, & g(\tilde{x}) \leq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4.33}$$

For convenience, define

$$\begin{aligned} J_{Q_{\Delta t}} &\coloneqq \mathbb{E}\Big[\sum_{k=0}^{\tau} \|\tilde{x}_k\|^2_{Q_{\Delta t}}\Big], \\ \bar{\tau} &\coloneqq \mathbb{E}[\tau]. \end{aligned} \tag{4.34}$$

The discrete-time approximation of the OSP can then be expressed as

$$\min_{\tau} J_{\text{osp},\Delta t} = \min_{\tau} \frac{J_{Q_{\Delta t}} + \rho}{\bar{\tau}}. \tag{4.35}$$

The stopping decision $s(\tilde{x})$ is explicitly introduced in the expressions for $J_{Q_{\Delta t}}$ and $\bar{\tau}$ via the following reformulations

$$\begin{aligned} J_{Q_{\Delta t}} &= \mathbb{E}\Big[\sum_{k=0}^{\tau} \|\tilde{x}_k\|^2_{Q_{\Delta t}}\Big] \\ &= \sum_{k=0}^{\infty} \mathbb{E}\Big[\sum_{j=0}^{k} \|\tilde{x}_j\|^2_{Q_{\Delta t}} s(\tilde{x}_k) \prod_{i=0}^{k-1} \big(1 - s(\tilde{x}_i)\big)\Big] \\ &= \sum_{k=0}^{\infty} \sum_{j=0}^{k} \mathbb{E}\Big[\|\tilde{x}_j\|^2_{Q_{\Delta t}} s(\tilde{x}_k) \prod_{i=0}^{k-1} \big(1 - s(\tilde{x}_i)\big)\Big], \end{aligned} \tag{4.36}$$

and

$$\bar{\tau} = \mathbb{E}[\tau] = \sum_{k=0}^{\infty} k\mathbb{E}\Big[s(\tilde{x}_k)\prod_{i=0}^{k-1}\big(1-s(\tilde{x}_i)\big)\Big]. \tag{4.37}$$

With a set of N trajectories $\{\tilde{x}_{n,k}\}_{k=0,\dots,K}^{n=1,\dots,N}$ generated over a sufficiently long finite time-horizon K, we can form Monte Carlo estimates of (4.36) and (4.37) as

$$J_{Q_{\Delta t}} \approx \hat{J}_{Q_{\Delta t}} \coloneqq \sum_{k=0}^{K}\sum_{j=0}^{k}\frac{1}{N}\sum_{n=1}^{N}\big\|\tilde{x}_{n,j}\big\|_{Q_{\Delta t}}^2 s(\tilde{x}_{n,k})\prod_{i=0}^{k-1}\big(1-s(\tilde{x}_{n,i})\big), \tag{4.38}$$

$$\bar{\tau} \approx \hat{\bar{\tau}} \coloneqq \sum_{k=0}^{K} k\frac{1}{N}\sum_{n=1}^{N} s(\tilde{x}_{n,k})\prod_{i=0}^{k-1}\big(1-s(\tilde{x}_{n,i})\big). \tag{4.39}$$

The final estimate of the objective in (4.35) is then given by

$$J_{\text{osp},\Delta t} \approx \hat{J}_{\text{osp},\Delta t} \coloneqq \frac{\hat{J}_{Q_{\Delta t}} + \rho}{\hat{\bar{\tau}}}.$$

With an ellipsoidal trigger function of the form (4.31), we can then find an (approximately) optimal set of parameters through the minimization

$$\min_{P} \hat{J}_{\text{osp},\Delta t}, \tag{4.40}$$

for some fixed value of Δ. Since the precise value of Δ is just a matter of scaling, we can simply use, e.g., $\Delta = 1$ in the optimization without loss of generality.

An obstacle when solving (4.40) is the fact that the sampling policy given by (4.33) is non-differentiable with respect to the ellipse parameters, leaving only zeroth-order methods available in the optimization. To practically perform optimization over large parameter spaces, we therefore need to introduce a smooth approximation. In [Becker et al., 2019], a logistic function approximation $\hat{s} : \mathbb{R}^n \to (0,1)$ of the sampling decision $s(\tilde{x})$ is proposed according to

$$\hat{s}(\tilde{x}) = \frac{1}{1+\exp(-\alpha g(\tilde{x}))}, \tag{4.41}$$

where the parameter $\alpha > 0$ determines the slope of the logistic function, with $\hat{s} \to s$ as $\alpha \to \infty$. The choice of α is a trade-off between bias and variance of the estimates $\hat{J}_{Q_{\Delta t}}$ and $\hat{\bar{\tau}}$.

Remark 4.2
Relating to the later topic of stochastic triggering in Chapter 6, it is worth noting that the smooth approximation can be directly interpreted as sampling being triggered according to a probability given by $\hat{s}(\tilde{x})$. To see this,

let the sampling decisions $\{s_k\}_{k=0}^{\infty}$ be a sequence of random variables that are conditionally independent given $\{\tilde{x}_k\}_{k=0}^{\infty}$. We then have

$$\begin{aligned}
J_{Q_{\Delta t}} &= \sum_{k=0}^{\infty}\sum_{j=0}^{k}\mathbb{E}\Big[\big\|\tilde{x}_j\big\|_{Q_{\Delta t}}^2 s_k \prod_{i=0}^{k-1}\big(1-s_i\big)\Big] \\
&= \sum_{k=0}^{\infty}\sum_{j=0}^{k}\mathbb{E}\Big[\big\|\tilde{x}_j\big\|_{Q_{\Delta t}}^2 \mathbb{E}\big[s_k \prod_{i=0}^{k-1}\big(1-s_i\big) \mid \tilde{x}_0,\dots,\tilde{x}_k\big]\Big] \\
&= \sum_{k=0}^{\infty}\sum_{j=0}^{k}\mathbb{E}\Big[\big\|\tilde{x}_j\big\|_{Q_{\Delta t}}^2 \mathbb{E}\big[s_k \mid \tilde{x}_0,\dots,\tilde{x}_k\big]\prod_{i=0}^{k-1}\mathbb{E}\big[1-s_i \mid \tilde{x}_0,\dots,\tilde{x}_i\big]\Big] \\
&= \sum_{k=0}^{\infty}\sum_{j=0}^{k}\mathbb{E}\Big[\big\|\tilde{x}_j\big\|_{Q_{\Delta t}}^2 \Pr\big(s_k=1 \mid \tilde{x}_0,\dots,\tilde{x}_k\big)\prod_{i=0}^{k-1}\Pr\big(s_i=0 \mid \tilde{x}_0,\dots,\tilde{x}_i\big)\Big],
\end{aligned}$$

where we used the law of total probability in the second equality and the conditional independence in the third equality. The stochastic sampling interpretation is then apparent by identifying

$$\Pr(s_k=1 \mid \tilde{x}_0,\dots,\tilde{x}_k) = \Pr(s_k=1|\tilde{x}_k) \coloneqq \hat{s}(\tilde{x}_k). \qquad \diamondsuit$$

With the smooth approximation (4.41), both $J_{Q_{\Delta t}}$ and $\bar{\tau}$ become differentiable in the ellipse parameters, enabling the use of first or second-order optimization methods. Since the number of trajectories N is typically very large, it is beneficial from a computational perspective to only consider subsets of the trajectories when evaluating the gradient or Hessian—i.e., mini-batch or stochastic methods [Goodfellow et al., 2016, Chapter 4]. These methods can then be combined with other extensions common in the field of machine learning, such as momentum, decay, and adaptive step size, etc. For more information on mini-batch and stochastic methods, we refer to [Goodfellow et al., 2016, Chapters 4 & 8].

Numerical Validation. We conclude by validating the Monte Carlo approach numerically against a set of higher-order OSPs with known solutions. Similar to the validation done in Section 4.4, we consider sets of multivariate integrator systems of the form:

$$\begin{aligned}
A &= 0_{n\times n}, \\
R &= I_n, \\
Q &= \begin{bmatrix} q_1 & & \\ & \ddots & \\ & & q_n \end{bmatrix},
\end{aligned} \tag{4.42}$$

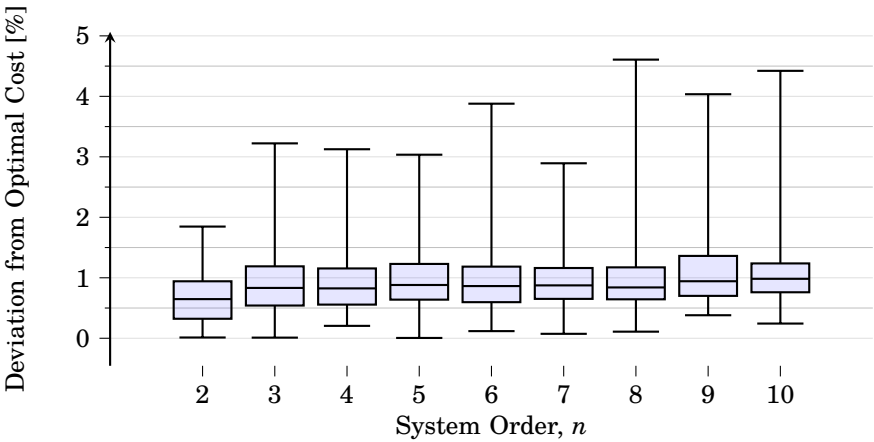

Figure 4.8 Box plot showing the distributions of relative cost difference between the optimal policy and the policy obtained via the Monte Carlo approach. The lower, mid, and upper line in each box correspond to the first, second (median) and third quartile, respectively, while the whiskers correspond to the minimum and maximum. The distribution for each system order n is based on a set of 100 randomized multivariate integrator OSPs.

where $q_i > 0$ for all $i = 1, \ldots, n$. However, in contrast to Section 4.4, we consider varying system orders n in this validation. For each system order n, we pick 100 systems of the form in (4.42), with randomly selected parameters in the ranges $\rho \in [0.01, 1]$ and $q_i \in [1, 10]$ for all $i = 1, \ldots, n$. We choose to validate against this class of OSP since it has an analytic solution for any order n, with an optimal threshold policy in the form of an ellipsoid. When optimizing ellipse parameters using the Monte Carlo approach, we then have a known optimum available for comparison.

In the Monte Carlo approach, each OSP is discretized using a nominal time step $\Delta t = 10^{-3}$, and a total of $N = 10\,000$ trajectories are then generated for the optimization, divided into 100 mini-batches. A smooth approximation according to (4.41) is employed, with $\alpha = 70$. The parameter optimization is then performed using the ADAM algorithm [Kingma and Ba, 2014] for 100 epochs. The optimization is implemented using the machine learning library `Flux.jl` [Innes et al., 2018] for the Julia programming language [Bezanson et al., 2017]. When a solution is found, the cost for the corresponding ellipsoidal trigger function is estimated using (4.38) and (4.39) with the exact policy (4.33) using a new set of $N = 10\,000$ trajectories.

The results are presented in Figure 4.8, where deviations from the optimal cost for the policy obtained from the Monte Carlo approach are shown for different system orders n. We see that roughly half of the data is clus-

tered around 1 % for all system orders and that the maximum deviation varies between roughly 3–4.5 %. The difference in cost can be made even smaller by using a smaller time step Δt and a larger number of trajectories N. However, for all practical purposes, the results in Figure 4.8 demonstrate that the Monte Carlo approach can be used to obtain (near) optimal parameters for ellipsoidal trigger functions for higher-order systems.

4.7 Summary

This chapter presented three methods for solving the optimal sampling problem (OSP) numerically; value iteration, a simulation-based method, and a linear complementarity method. Value iteration is a standard tool in dynamic programming, while the simulation-based and linear complementarity method are contributions of this thesis. The simulation-based method is derived directly from the free boundary problem of the OSP and features a time-dependent formulation that can be simulated using standard PDE tools. The linear complementarity method is proposed as an alternative that avoids time-dependence.

The three methods were validated and compared in a numerical study. While the linear complementarity method has an estimated convergence rate of order 1, the simulation-based method and value iteration both settle at a constant error for increasing mesh sizes. A comparison showed that the simulation-based method has a better trade-off between accuracy and solver time compared to value iteration, arguably making it the better option. The linear complementarity method was also seen to outperform both the simulation-based method and value iteration in terms of accuracy.

The relation between system dynamics and the shape of the optimal sampling threshold was investigated using the linear complementarity method. Results from this investigation showed that most second-order systems have a near-ellipsoidal optimal threshold. A notable exception is saddle-point systems, which have non-convex optimal thresholds.

The last part of the chapter was dedicated to ellipsoidal trigger functions, which seem like a promising alternative to the optimal policy. A numerical example demonstrated that ellipsoidal trigger functions can give near-optimal performance even in cases when the optimal threshold is non-convex. To find suitable ellipse parameters for any system order, we proposed the Monte Carlo approach described in [Becker et al., 2019]. This approach was validated numerically against the known solution for the multivariate integrator of different system orders.

Ellipsoidal trigger functions constitute the first step from optimal to suboptimal, but simpler, event-based policies in this thesis. This focus is retained in the next chapter, with the topic of event-based PID control.

5

Event-Based PID from an LQG Perspective

This chapter considers results on LQG-optimal sampled-data control from Chapters 2 and 3 as a starting point for the design of event-based proportional-integral-derivative (PID) control. Since the pioneering work in [Årzén, 1999], a rich literature of event-based PID control has emerged with many different design heuristics. The purpose of the LQG perspective in this chapter is to better understand which design choices are actually performance-critical, and use this understanding to propose an implementation that strikes a balance between performance and simplicity.

The link between the results on LQG-optimal sampled-data control (Mirkin's LQG controller, see Section 2.5) and PID control are established via a special benchmark LQG design problem, whose solution is an ideal PID controller. This benchmark problem was first presented in [Cervin and Thelander Andrén, 2020], which in turn was influenced by [Cervin, 2016]. Since the solution can be viewed from the perspectives of both LQG and PID control, we can interpret Mirkin's LQG controller as a form of sampled-data PID controller. While this optimal event-based PID controller may be difficult to implement in practice, it still serves as a useful baseline for comparisons of suboptimal but simpler designs.

This chapter is structured as follows. In Section 5.1, we formulate the benchmark LQG design problem. Based on the benchmark problem, we interpret Mirkin's LQG controller as a sampled-data PID controller in Section 5.2. In Section 5.3, we discuss various design choices for event-based PID control, and how they relate to the optimal solution. These design choices are then evaluated in a simulation study in Section 5.4, based on which we give a proposal for a simple, yet well-performing, implementation. Finally, the chapter is summarized in Section 5.5.

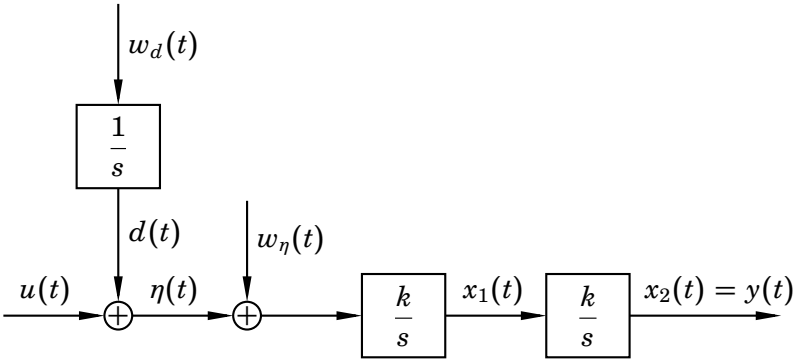

Figure 5.1 The considered process model for the benchmark problem, consisting of a double integrator and an integral input disturbance.

5.1 The Benchmark Problem

Consider the system in Figure 5.1, consisting of a double integrator and a low-frequency input disturbance d—a very basic control model for, e.g., simple mechanical systems such as a satellite or a cart on a rail. We begin by showing how an LQG design problem for this system can be formulated such that the solution is an ideal PID controller.

LQG Design

For the system in Figure 5.1, assume that the gain $k > 0$ is a given scalar parameter and that w_d and w_η are two scalar and independent continuous-time Gaussian white processes with intensities $r_d > 0$ and 1, respectively. Consider the following LQG design objective:

$$J := \lim_{T\to\infty} \mathbb{E}\Big[\frac{1}{T}\int_0^T \big(q_y y^2(t) + 2q_{y\eta} y(t)\eta(t) + \eta^2(t)\big)dt\Big], \tag{5.1}$$

where $q_y > 0$ and $q_{y\eta}^2 \leq q_y$ are scalar weights. We refer to the minimization of (5.1) for the system in Figure 5.1 as the *benchmark problem*. Note in (5.1) that we penalize $\eta = u + d$ rather than the control signal u. By also penalizing the low-frequency input disturbance d, the optimal controller is ensured to feature integral action to suppress it. This is a common construction in LQG control design, see, e.g., [Glad and Ljung, 2000, Chapter 9]

We now proceed by deriving the optimal analog LQG controller for the benchmark problem. Since the state d of the input disturbance is not stabilizable, the optimal state feedback can not be obtained directly via the algebraic Riccati equation (ARE) (2.8). Instead, assume for now that we control η directly. In the framework of Chapter 2, the corresponding sub-

problem is defined by

$$G:\begin{cases}\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix}=\begin{bmatrix}0 & 0\\ k & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}+\begin{bmatrix}k\\ 0\end{bmatrix}w_\eta(t)+\begin{bmatrix}k\\ 0\end{bmatrix}\eta(t),\\ z(t)=\begin{bmatrix}0 & \sqrt{q_y}\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}+\begin{bmatrix}\frac{q_{y\eta}}{\sqrt{q_y}}\\ \sqrt{1-\frac{q_{y\eta}^2}{q_y}}\end{bmatrix}\eta(t),\\ y(t)=\begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix},\end{cases}\tag{5.2}$$

where the objective is given by

$$J=\lim_{T\to\infty}\mathbb{E}\Big[\frac{1}{T}\int_0^T\|z(t)\|_2^2dt\Big].$$

In contrast to the original benchmark problem, the optimal state feedback for (5.2) can be obtained directly via the ARE (2.8), yielding

$$\eta(t)=f_1x_1(t)+f_2x_2(t),\tag{5.3}$$

with

$$f_1=-\sqrt{2\big(\sqrt{q_y}-q_{y\eta}\big)},\tag{5.4}$$

$$f_2=-\sqrt{q_y}.\tag{5.5}$$

Returning to the original benchmark problem, we see that the objective in (5.1) is minimized if the control signal u is chosen such that the relation (5.3) holds. With complete state information (including d) the optimal feedback law is thus given by

$$u(t)=f_1x_1(t)+f_2x_2(t)-d(t).\tag{5.6}$$

We proceed by deriving the optimal state estimator. Because of the lack of measurement noise in the process model, the states x_1 and x_2 can be reconstructed exactly from the measured output y as

$$\begin{aligned}x_2(t)&=y(t),\\ x_1(t)&=\frac{\dot{x}_2(t)}{k}=\frac{\dot{y}(t)}{k}.\end{aligned}\tag{5.7}$$

Thus, only the disturbance d remains to be estimated. The lack of process noise on x_1 allows us to to obtain a reduced-order Kalman–Bucy filter for d as

$$\dot{\hat{d}}(t)=-l_d\Big(\frac{\dot{x}_1(t)}{k}-u(t)-\hat{d}(t)\Big)=-l_d\Big(\frac{\ddot{y}(t)}{k^2}-u(t)-\hat{d}(t)\Big),\tag{5.8}$$

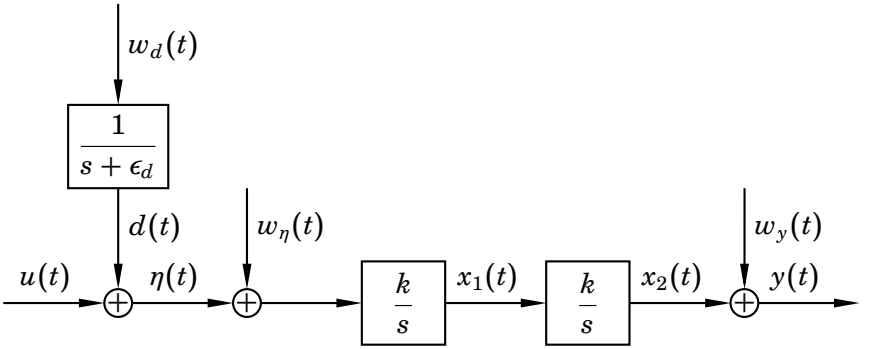

Figure 5.2 A slightly perturbed version of the benchmark process model in Figure 5.1. Here, a white Gaussian process w_y with intensity r_y has been added to the measured output, and the integrator of the disturbance d has been changed into a stable first-order system with a pole in $-\epsilon_d < 0$. The original system is recovered in the limit $r_y, \epsilon_d \to 0^+$.

where $\hat{d}$ is the MMSE estimate of d, and the gain l_d is obtained by solving the associated Riccati equation (2.5), yielding

$$l_d = -\sqrt{r_d}. \tag{5.9}$$

Combining the optimal state feedback (5.6) with the state estimator (5.7) and (5.8), we obtain the complete analog LQG controller K for the benchmark problem as

$$K : \begin{cases} \hat{x}_1(t) = \dfrac{\dot{y}(t)}{k}, \\ \hat{x}_2(t) = y(t), \\ \dot{\hat{d}}(t) = -l_d\Big(\dfrac{\ddot{y}(t)}{k^2} - u(t) - \hat{d}(t)\Big), \\ u(t) = f_1\hat{x}_1(t) + f_2\hat{x}_2(t) - \hat{d}(t). \end{cases} \tag{5.10}$$

The derivation of the analog LQG controller above did not use the standard LQG assumptions **A1** (due to d not being stabilizable) and **A2** (due to the lack of measurement noise) (see Section 2.1). An alternative approach is to instead consider a slightly perturbed version of the original system, see Figure 5.2. Here, measurement noise in the form of a scalar white Gaussian process signal w_y with (low) intensity $r_y > 0$ has been added to x_2, and the integrator of the disturbance d has been changed into a stable first-order system with a pole in $-\epsilon_d$, for some small value $\epsilon_d > 0$. With the modified process model in Figure 5.2, the conditions **A1** and **A2** are satisfied and the

analog LQG controller can be computed according to the standard framework described in Section 2.2. The original benchmark problem is recovered by taking the limit $r_y, \epsilon_d \to 0^+$, at which point the solution obtained for the perturbed system coincides with the one given in (5.10).

PID Interpretation

We now proceed by showing that the analog LQG controller in (5.10) is in fact equivalent to a PID controller. To this end, consider the change of variable

$$x_i(t) \coloneqq -\hat{d}(t) - \frac{l_d}{k}\hat{x}_1(t) + \frac{l_d f_1}{k}\hat{x}_2(t),$$

which by (5.10) evolves according to

$$\dot{x}_i(t) = -l_d f_2 \hat{x}_2(t) = -l_d f_2 y(t).$$

By introducing x_i, the feedback law of the analog LQG controller can be rewritten as

$$\begin{aligned}
u(t) &= f_1\hat{x}_1(t) + f_2\hat{x}_2(t) - \hat{d}(t) \\
&= \frac{f_1 k + l_d}{k}\hat{x}_1(t) + \frac{f_2 k - l_d f_1}{k}\hat{x}_2(t) + x_i(t) \\
&= \frac{f_2 k - l_d f_1}{k}\Big(\hat{x}_2(t) + \frac{k}{f_2 k - l_d f_1}x_i(t) + \frac{f_1 k + l_d}{f_2 k - l_d f_1}\hat{x}_1(t)\Big) \\
&= -\frac{l_d f_1 - k f_2}{k}\Big(y(t) + \frac{l_d k f_2}{l_d f_1 - k f_2}\int_0^t y(s)ds + \frac{-f_1 k - l_d}{l_d k f_1 - k^2 f_2}\dot{y}(t)\Big) \\
&\coloneqq -K\Big(y(t) + \frac{1}{T_i}\int_0^t y(s)ds + T_d\dot{y}(t)\Big),
\end{aligned} \tag{5.11}$$

which is an ideal PID controller on parallel form, with

$$\begin{aligned}
K &= \frac{l_d f_1 - k f_2}{k}, \\
T_i &= \frac{l_d f_1 - k f_2}{l_d k f_2}, \\
T_d &= \frac{-f_1 k - l_d}{l_d k f_1 - k^2 f_2}.
\end{aligned} \tag{5.12}$$

Thus, we have shown that the optimal controller for the benchmark problem has two equivalent representations; the analog LQG controller in (5.10) and the ideal PID controller in (5.11).

Remark 5.1
For a given ideal PID controller with positive parameters K, T_i, and T_d it is also possible to find a corresponding (non-unique) benchmark problem. For details, see Appendix B. ♢

A Note on Integral Action

The state x_i represents the integral action of the PID controller, since

$$\dot{x}_i(t) = -l_d f_2 y(t) = -\frac{K}{T_i} y(t). \tag{5.13}$$

The standard representation in (5.13) is only one of many possible formulations, and here we give a brief note on these.

First, define the PI part of the control signal as

$$u_{\text{pi}}(t) \coloneqq -Ky(t) + x_i(t).$$

The integral action (5.13) can then alternatively be represented as

$$\begin{aligned} \dot{x}_i(t) &= -\frac{1}{T_i}\big(Ky(t) + u_{\text{pi}}(t) - u_{\text{pi}}(t)\big) \\ &= -\frac{1}{T_i}\big(2Ky(t) - x_i(t) + u_{\text{pi}}(t)\big), \end{aligned} \tag{5.14}$$

i.e., expressed in terms of x_i and u_{pi} in addition to y. More generally, by introducing a parameter $\alpha \in \mathbb{R}$ and splitting up the PI part according to

$$u_{\text{pi}}(t) = \alpha u_{\text{pi}}(t) + (1-\alpha)u_{\text{pi}}(t) = -\alpha K y(t) + \alpha x_i(t) + (1-\alpha)u_{\text{pi}}(t),$$

and inserting this relation into (5.14), we retrieve a family of possible representations for the integral action as

$$\dot{x}_i(t) = -\frac{1}{T_i}\big((1+\alpha)Ky(t) - \alpha x_i(t) + \alpha u_{\text{pi}}(t)\big). \tag{5.15}$$

We see that $\alpha = 0$ corresponds to the original expression in (5.13), while $\alpha = 1$ gives (5.14). With $\alpha = -1$ we eliminate y from (5.15), resulting in

$$\dot{x}_i(t) = -\frac{1}{T_i}\big(x_i(t) - u_{\text{pi}}(t)\big). \tag{5.16}$$

Integral action of the form in (5.16) can be recognized as the classical *automatic reset* realization [Åström and Murray, 2008, Chapter 10]. In a continuous feedback setting, the behavior of the integral action is identical regardless of the choice of α. However, in an event-based implementation, different realizations may yield different results depending on when and how the variables are communicated between the sensor, controller, and actuator.

5.2 LQG-Optimal Event-Based PID

Having established that the analog LQG controller for the benchmark problem can be interpreted as an ideal PID controller, we now consider the sampled-data case. In Chapter 2, it was established that Mirkin's LQG controller (i.e., the optimal sampled-data controller) has a representation closely connected to the analog LQG controller. For the benchmark problem, this implies that Mirkin's LQG controller also can be interpreted as a PID controller.

Optimal Sampled-Data LQG Design

While the derivation of Mirkin's LQG controller assumes that **A1**–**A3** are satisfied (see Section 2.1), we can, similarly to [Braksmayer and Mirkin, 2017], obtain a formal solution to the benchmark problem by considering the perturbed system Figure 5.2, followed by taking the limit $r_y, \epsilon_d \to 0^+$. Recalling that Mirkin's LQG controller can be represented as a Kalman–Bucy filter on the sensor side, and a simulation-based LQR on the actuator side (see Figure 2.7), the limiting solution is given in terms of the reduced-order filter in (5.7) and (5.8), and state feedback in (5.6). Thus, on the sensor side of Mirkin's LQG controller, the MMSE estimates are generated according to

$$\text{Sensor:} \begin{cases} \hat{x}_1(t) = \dfrac{\dot{y}(t)}{k}, \\ \hat{x}_2(t) = y(t), \\ \dot{\hat{d}}(t) = -l_d\big(\dfrac{\ddot{y}(t)}{k^2} - u(t) - \hat{d}(t)\big), \end{cases}$$

while on the actuator side, the control signal is generated according to

$$\text{Actuator:} \begin{cases} \dot{x}_{1,a}(t) = k d_a(t) + k u(t), \\ \dot{x}_{2,a}(t) = k x_{1,a}(t), \\ \dot{d}_a(t) = 0, \\ u(t) = f_1 x_{1,a}(t) + f_2 x_{2,a}(t) - d_a(t), \end{cases}$$

where the actuator states $x_{1,a}$, $x_{2,a}$ and d_a are reset at sampling times $\{\tau_k\}$ according to

$$k \in \mathbb{N}_0 : \begin{cases} x_{1,a}(\tau_k) = \hat{x}_1(\tau_k), \\ x_{2,a}(\tau_k) = \hat{x}_2(\tau_k), \\ d_a(\tau_k) = \hat{d}(\tau_k). \end{cases}$$

PID Interpretation of Mirkin's LQG Controller

With the following definitions

$$\begin{aligned} y_a(t) &\coloneqq x_{2,a}(t), \\ x_{i,a}(t) &\coloneqq -d_a(t) - \frac{l_d}{k} x_{1,a}(t) + \frac{l_d f_1}{k} x_{2,a}(t), \end{aligned}$$

we can use the same derivation as in Section 5.1 to show that Mirkin's LQG controller for the benchmark problem is equivalent to

$$\text{K:} \begin{cases} \ddot{y}_a(t) = k f_1 \dot{y}_a(t) + k^2 f_2 y_a(t), \\ \dot{x}_{i,a}(t) = -\dfrac{K}{T_i} y_a(t), \\ \\ y_a(\tau_k) = y(\tau_k), \\ x_{i,a}(\tau_k) = x_i(\tau_k) = -\dfrac{K}{T_i} \displaystyle\int_0^{\tau_k} y(t)dt, \\ \dot{y}_a(\tau_k) = \dot{y}(\tau_k), \\ \\ u(t) = -K y_a(t) + x_{i,a}(t) - K T_d \dot{y}_a(t), \quad t \in [\tau_k, \tau_{k+1}),\ k \in \mathbb{N}_0. \end{cases} \tag{5.17}$$

In (5.17), we see that the control signal u is generated by an analog PID controller based on the simulated output y_a. On the sensor side, the "observer" tracks the measured output y, its integral and its derivative, and transmits $(y,\ x_i,\ \dot{y})$ to the actuator at each sampling instance.

This controller structure can be difficult to realize in practice, but from a theoretical point of view it sheds some light on how an "ideal" sampled-data PID is structured, and provides a useful performance baseline for comparisons to more practical implementations.

Sampling Policy

In Chapter 3 it was established that the optimal sampling policy for Mirkin's LQG controller is of threshold type (see Theorem 3.1). The threshold policy is based on the error vector $\begin{bmatrix} x_{1,a} - \hat{x}_1 & x_{2,a} - \hat{x}_2 & d_a - \hat{d} \end{bmatrix}^\mathsf{T}$, which after a linear change of coordinates is equivalent to

$$\begin{bmatrix} \tilde{y} \\ \tilde{x}_i \\ \dot{\tilde{y}} \end{bmatrix} \coloneqq \begin{bmatrix} y_a - y \\ x_{i,a} - x_i \\ \dot{y}_a - y \end{bmatrix}. \tag{5.18}$$

Simply put; (5.18) implies that sampling should be triggered whenever the P, I or D part of the control signal deviates too much from that of the analog PID controller.

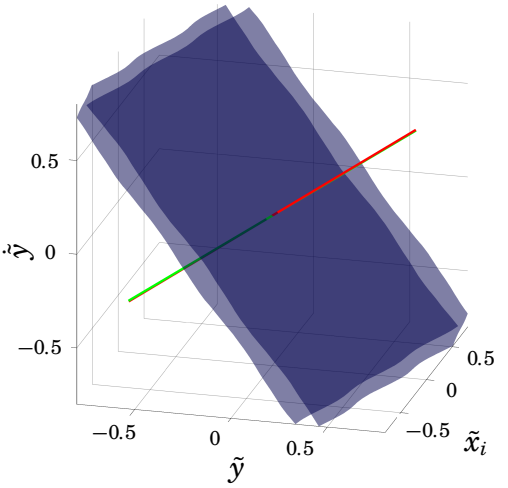

Figure 5.3 Numerical approximation of the optimal sampling threshold $\partial\Omega$ (blue surfaces) for the benchmark problem with parameters $k = r_d = 1$, $q_y = 4$ and $q_{y\eta} = 0$. The approximation was computed using the linear complementarity method described in Section 4.3 with $25 \times 25 \times 25 = 15\,625$ uniformly distributed collocation points. The threshold is well-approximated by two parallel planes with a normal illustrated by the red line. The green line is parallel to the vector $[K, -1, KT_d]$, and differs only by 1° from the red line.

To obtain the optimal threshold, we can employ the numerical methods presented in Chapter 4. As illustrated by the example in Figure 5.3, the resulting threshold for the benchmark problem is well-approximated by two parallel planes. In Figure 5.3, the red line illustrates the normal of a plane fitted to the optimal threshold (blue surfaces) using least squares, while the green line illustrates the direction of the vector $\begin{bmatrix} K & -1 & KT_d \end{bmatrix}$. The red and green lines differ only by approximately 1°, and we therefore expect near-optimal performance from a trigger function of the form

$$g(\tilde{y}, \tilde{x}_i, \dot{\tilde{y}}) = \Delta - \left| K\tilde{y} - \tilde{x}_i + KT_d\dot{\tilde{y}} \right|, \tag{5.19}$$

where Δ is a parameter chosen as a trade-off between LQG-performance and average sampling rate (cf. Section 3.4). In comparisons with other PID controllers, we use the (near optimal) trigger function in (5.19) for the controller in (5.17).

5.3 Implementation Aspects of Event-Based PID

Event-based implementations of PID controllers are usually motivated by improved resource efficiency, especially in networked control systems, where savings in energy and bandwidth can be achieved by transmitting data less often. However, the computational capacity in the sensor and actuator nodes are usually limited in embedded implementations, which makes complex triggering conditions and signal generators infeasible. Arguably, this is the case for the optimal controller of the previous section, which motivates the need for suboptimal but simpler implementations. In this section, we highlight some features of previously proposed event-based PID controllers from the literature and discuss how they relate to the optimal solution.

Baseline: Årzén's Simple Event-Based PID Controller

Most proposals of event-based PID controllers in the literature can be traced back to Årzén's simple event-based PID controller proposed in the seminal work [Årzén, 1999]. Therefore, we proceed by briefly reviewing its implementation and use it as a baseline in the remainder of this chapter.

On the sensor side of Årzén's controller, the system output y is monitored periodically with a short, fixed, period h_{nom}. The event of sampling and transmission of y to the actuator side is triggered based on a send–on–delta rule combined with a time-out $h_{\max}$. With the time interval between two events defined as

$$h_k \coloneqq \tau_k - \tau_{k-1},$$

the event trigger of Årzén's controller is given by

$$h_k = \min_{l=1,\ldots,\infty} \left\{ h = l h_{\text{nom}} : |y(h + \tau_{k-1}) - y(\tau_{k-1})| > \Delta \text{ or } h \geq h_{\max} \right\}, \quad (5.20)$$

where $\Delta > 0$ is a parameter chosen as a trade-off between performance and average sampling rate. Compared to the sampling policy in (5.19), which uses all three PID components, we see that Årzén's controller only considers the measured output y (i.e., only the P part). Furthermore, the measured output y is compared to the last sampled value rather than a time-varying prediction. However, the policy (5.20) has the practical benefit of being very simple to implement, as demonstrated by the following pseudo-code:

```
y := AnalogIn();
h := h + h_nom;
IF abs(y - y_old) >= delta OR h >= h_max THEN
  Send(y);
  y_old := y;
ENDIF
```

On the actuator side of Årzén's controller, the control signal is generated by a discretized version of an analog PID controller with derivative filter, combined with a zero-order hold. Using forward and backward difference approximations for the integral action and the derivative part respectively, the control signal u is generated according to

$$\text{Actuator:} \begin{cases} x_i(\tau_{k+1}) = x_i(\tau_k) - \dfrac{K}{T_i} h_k y(\tau_k), \\ u_d(\tau_{k+1}) = \dfrac{T_d}{N h_{k+1} + T_d} u_d(\tau_k) - \dfrac{K N T_d}{N h_{k+1} + T_d} \big(y(\tau_{k+1}) - y(\tau_k)\big), \\ \qquad u(t) = -K y(\tau_k) + x_i(\tau_k) + u_d(\tau_k), \quad t \in [\tau_k, \tau_{k+1}), \ k \in \mathbb{N}_0, \end{cases} \tag{5.21}$$

where u_d represents the derivative part of the PID controller and N is the maximum derivative gain. Without any measurement noise, we can let $N \to \infty$ and the derivative part becomes a pure backward difference. In contrast to the optimal actuator policy in (5.17), Årzén's controller holds a constant control signal over each sampling interval. By using zero-order hold, the implementation does not require any custom analog circuitry for signal generation, and the actuator policy has the following straightforward digital implementation:

```
y := Receive();
h := Time() - time_old;
a_d := T_d / (N * h + T_d);
u_d := a_d * u_d - K * N * a_d * (y - y_old);
u := -K*y + x_i + u_d;
AnalogOut(u);
x_i := x_i - K / T_i * h * y;
y_old := y;
time_old := Time();
```

Choice of Sampling Policy

Årzén's controller, and many subsequent ones, use a send–on–delta rule [Miskowicz, 2015] based on the difference between the current and last sampled value of the measured output y. Aside from the possible addition of a time-out and different scaling of y, the send–on–delta rule can be represented by a trigger function of the following form:

$$g(\tilde{y}) = \Delta - \big|K\tilde{y}\big|, \tag{5.22}$$

where

$$\tilde{y}(t) \coloneqq y(t) - y(\tau_k), \quad t \in [\tau_k, \tau_{k+1}), \ k \in \mathbb{N}_0. \tag{5.23}$$

Because send–on–delta essentially is a policy based on the P part of a PID controller, we refer to it from now on as *P triggering*.

It has been pointed out in several works, e.g., [Miskowicz, 2007; Tiberi et al., 2012], that sampling based on the integral of y can be useful as it is less sensitive to noise and also eliminates the deadband effect. Similar to [Tiberi et al., 2012], we can take the integral of y into account with a trigger function of the form:

$$g(\tilde{y}, \tilde{x}_i) = \Delta - \left|K\tilde{y} - \tilde{x}_i\right|, \tag{5.24}$$

where

$$\tilde{x}_i(t) \coloneqq x_i(t) - x_i(\tau_k), \quad t \in [\tau_k,\ \tau_{k+1}),\ k \in \mathbb{N}_0, \tag{5.25}$$

and $\tilde{y}$ is given by (5.23). We refer to (5.24) as *PI triggering*.

Based on the sampling policy (5.19) obtained for the benchmark problem, it is natural to also consider sampling policies that include the derivative part of a PID controller. The first option has a trigger function of the form:

$$g(\tilde{y}, \tilde{\dot{y}}) = \Delta - \left|K\tilde{y} + KT_d\tilde{\dot{y}}\right|, \tag{5.26}$$

where

$$\tilde{\dot{y}}(t) \coloneqq \dot{y}(t) - \dot{y}(\tau_k), \quad t \in [\tau_k,\ \tau_{k+1}),\ k \in \mathbb{N}_0, \tag{5.27}$$

and $\tilde{y}$ is given by (5.23). We refer to (5.26) as *PD triggering*. The other option includes all three components according to

$$g(\tilde{y}, \tilde{x}_i, \tilde{\dot{y}}) = \Delta - \left|K\tilde{y} - \tilde{x}_i + KT_d\tilde{\dot{y}}\right|, \tag{5.28}$$

with $\tilde{y}$, $\tilde{x}_i$ and $\tilde{\dot{y}}$ given by (5.23), (5.25) and (5.27), respectively. We refer to (5.28) as *PID triggering*. Note that there is a subtle difference between the PID trigger (5.28) and the policy (5.19), i.e., the PID trigger does not compare $(y,\ x_i,\ \dot{y})$ to the model-based predictions $(y_a,\ x_{i,a},\ \dot{y}_a)$, but rather to the last transmitted values.

While a true continuous-time implementation of the PI/PD/PID triggers requires custom analog circuitry, they would be relatively simple to realize in conjunction with an analog prefilter—a feature in virtually all practical implementations. Based on the argument in [Åström and Hägglund, 2006, Section 13.3], consider a second-order prefilter of the form

$$Y_f(s) = \frac{1}{1 + sT_f + (sT_f)^2/2} Y(s), \tag{5.29}$$

where Y, Y_f are the Laplace representations of the raw and filtered measurement y and y_f respectively, and T_f is the filter's time constant. As shown in Figure 5.4, the filter in (5.29) can be constructed as a feedback circuit with a stable first-order filter and an integrator. This setup combines prefiltering with easy access to both y_f and $\dot{y}_f$ for feedback. With one additional integrator in Figure 5.4 we also gain access to x_i.

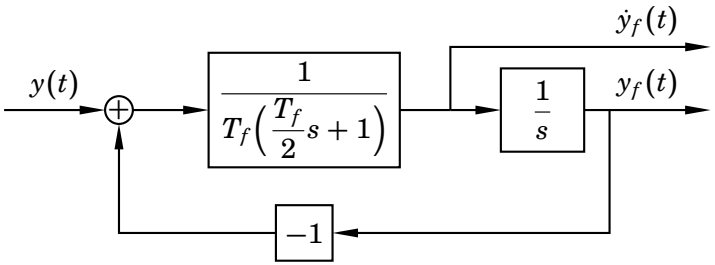

Figure 5.4 Block diagram of analog second-order prefilter with easy access to both the filtered output y_f and its derivative $\dot{y}_f$.

Choice of Data to Communicate

Most heuristic methods only communicate the sensor value y to the actuator side. The policy (5.19) obtained for the benchmark problem, however, transmits $(y,\ x_i,\ \dot{y})$. Sending a few extra bytes in a network packet costs very little, and, depending on if PI, PD, or PID triggering is used, some combination of y, x_i, and $\dot{y}$ are already available in the sensor node and should therefore be communicated to the actuator.

Choice of Integrator Implementation

Since a standard finite-difference approximation is used for the integral part in Årzén's controller, it behaves poorly for long inter-event times with large overshoots as a result. Furthermore, since the integral part is updated after the control signal, the period used for scaling in the integral update in (5.21) is delayed by one sampling event. These two drawbacks were addressed in [Durand and Marchand, 2009a], where the delay was removed and a forgetting factor was introduced to alleviate the problem of overshoots during long sampling periods. The result is an integral part computed according to

$$x_i(\tau_k) = x_i(\tau_{k-1}) - \frac{K}{T_i}(\tau_k - \tau_{k-1})e^{h_{\text{nom}}-(\tau_k-\tau_{k-1})}y(\tau_k). \tag{5.30}$$

We refer to (5.30) as the Durand–Marchand integrator.

For zero-order hold, an elegant alternative to the Durand–Marchand integrator is an implementation based on the automatic reset form in (5.16). Integrating (5.16) over one sampling period then yields

$$x_i(\tau_k) = e^{-\frac{1}{T_i}(\tau_k-\tau_{k-1})}x_i(\tau_{k-1}) + \left(1 - e^{-\frac{1}{T_i}(\tau_k-\tau_{k-1})}\right)u_{\text{pi}}(\tau_{k-1}). \tag{5.31}$$

This implementation has for instance been adopted in the PIDPLUS commercial controller [Song et al., 2006], which shows promise for application in event-based PI control [Tiberi et al., 2012].

Choice of Control Signal Generator

At the actuator side, zero-order hold between events is a standard assumption in heuristic implementations. Ideally, the feedback gain should then be adapted according to the length of each hold interval. One method is to match poles between the closed-loop systems under analog and sampled-data control as in [Åström and Wittenmark, 1997, Section 8.3]. A lookup table of PID parameters corresponding to different interval lengths can then be computed. However, the event-based sampling scheme prevents us from knowing the length of the next hold interval beforehand. A very simple heuristic is to assume that the duration of the current interval is equal to the last experienced one. We refer to this heuristic method as *adaptive gain*.

Since a suitable feedback gain under zero-order hold requires knowledge of the duration of the next hold interval, it may be better to instead consider impulse generators or generalized hold circuits as discussed in [Åström, 2008]. Indeed, Mirkin's LQG controller, and in particular the solution (5.17) of the benchmark problem, both utilize generalized hold circuits in the actuator. While generalized hold circuits have the potential of improving performance, their drawback is the increased complexity of the implementation. Based on the control signal generator of the benchmark solution (5.17), we propose two options. The first generates a control signal in between sampling times according to

$$t \in [\tau_k, \tau_{k+1}) : \begin{cases} \ddot{y}_a(t) = kf_1\dot{y}_a(t) + k^2 f_2 y_a(t), \\ u(t) = -Ky_a(t) + x_{i,a} - KT_d\dot{y}_a(t), \end{cases} \tag{5.32}$$

which we denote as a *PD generator*. While y_a and $\dot{y}_a$ are generated according to the differential equation, $x_{i,a}$ remains constant throughout the hold interval. The other option is the generator of the benchmark solution (5.17), which also includes the integrator:

$$t \in [\tau_k, \tau_{k+1}) : \begin{cases} \ddot{y}_a(t) = kf_1\dot{y}_a(t) + k^2 f_2 y_a(t), \\ \dot{x}_{i,a}(t) = -\dfrac{K}{T_i} y_a(t), \\ u(t) = -Ky_a(t) + x_{i,a}(t) - KT_d\dot{y}_a(t). \end{cases} \tag{5.33}$$

We denote this option as a *PID generator*. For both generators above, the signals y_a, $x_{i,a}$ and $\dot{y}_a$ are updated at sampling times with either the exact values (if they are collected and transmitted by the sensor) or some finite-difference approximation based on y.

5.4 Performance Evaluation

In this section, we evaluate the performance of the different design choices discussed in Section 5.3. We use Årzén's controller as a base implementation in each evaluation, and consider different options for each of the following aspects:

(i) sampling policy,

(ii) communicated data,

(iii) integrator implementation, and

(iv) control signal generator.

With higher resource efficiency being one of the main motivators for event-based control, we focus on the trade-off between average sampling rate (corresponding to mean network usage or sensor energy consumption) and the LQG cost as measured by (5.1). Two different setups are considered:

A. The process model of the benchmark problem in Figure 5.1, with gain, cost, and noise parameters given by $k = 1$, $q_y = 4$, $q_{y\eta} = 0$, $r_d = 1$. The optimal analog controller for this setup is a PID controller with the parameters $K = 4$, $T_i = 2$, $T_d = 0.75$.

B. The stable third-order process in Figure 5.5 with the cost and noise parameters $q_y = 5.5$, $q_{y\eta} = 0$, $r_d = 0.1$. For this higher-order process, the optimal analog controller is *not* a PID controller. However, we can find the best possible PID parameters using nonlinear optimization, yielding $K = 2.15$, $T_i = 2.67$, $T_d = 1.23$.

Setup A matches the benchmark problem exactly, while Setup B is representative of a lag-dominated stable process from process industry. In both cases, the LQG costs found in the evaluation have been normalized so that the continuous PID controller has a relative cost of 1.

The performance of the different controllers is evaluated through simulations in TrueTime [Henriksson et al., 2002]. Throughout, $h_{\text{nom}} = 0.01$ s is used as the simulation time step (and hence the smallest possible event detection interval). For each scenario, a 2 000 s simulation is run using the same noise input sequence. For the event-triggered algorithms, the trigger parameter Δ is swept over a range of values to generate different average sampling rates and LQG costs.

For reference, we also include results for Mirkin's LQG controller in each evaluation. Its implementation follows Section 5.2 for Setup A (where it has a PID interpretation given by (5.17)), and the general framework in

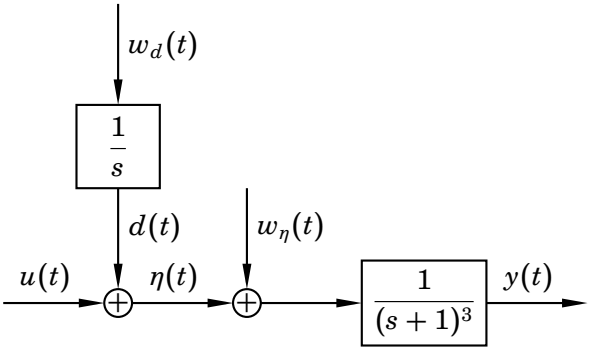

Figure 5.5 Process considered for Setup B in the performance comparisons.

Section 2.5 for Setup B. In both setups, the trigger function used by Mirkin's LQG controller is of the form (cf. Chapter 3)

$$g(\tilde{x}) = \Delta - |F\tilde{x}|,$$

which for Setup A is equivalent to (5.19).

Sampling Policy

We first study how the choice of sampling policy impacts the performance of the event-based PID controller. The sampling policies under study are:

(i) P trigger (5.22) (as in Årzén's controller),

(ii) PI trigger (5.24),

(iii) PD trigger (5.26), and

(iv) PID trigger (5.28).

The trade-off results for both setups are presented in Figure 5.6. In both setups we see only minor differences between P and PI triggering, indicating that the addition of the integral state x_i to the sampling policy only has a minor performance impact. In contrast, we see a significant improvement with the addition of the derivative $\dot{y}$, as indicated by the results from PD and PID triggering. The small difference between PD and PID triggering also confirms the minor role played by x_i in improving performance.

Communicated Data

The second implementation aspect under study is the choice of communicated data. The configurations of data communicated between the sensor and actuator node are:

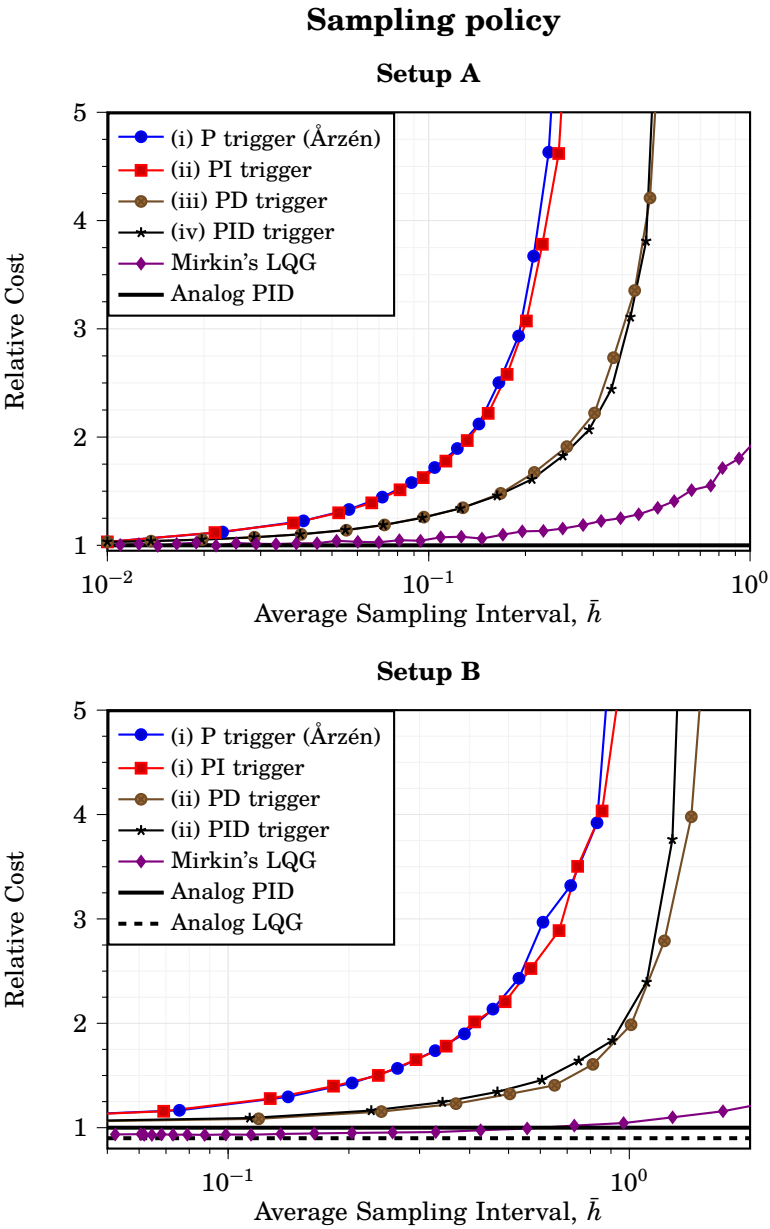

Figure 5.6 Relative LQG cost versus average sampling interval $\bar{h}$ for Setup A (upper) and B (lower) using different sampling policies.

(i) y (as in Årzén's controller),

(ii) (y, x_i),

(iii) $(y, \dot{y})$, and

(iv) $(y, x_i, \dot{y})$.

The trade-off results for both setups are presented in Figure 5.7. Similar to the evaluation of the sampling policy, we observe that the addition of derivative information $\dot{y}$ gives improved performance (albeit the improvement is smaller here). However, the addition of integral information x_i seems to have little or no effect for Setup A and actually results in somewhat worse performance for small sampling intervals in Setup B.

Integrator Implementations

We now study the choice of integrator implementation, with the following options:

(i) forward difference approximation (5.21) (as in Årzén's controller),

(ii) Durand–Marchand's integrator (5.30), and

(iii) automatic reset integrator (5.31).

The trade-off results for both setups are presented in Figure 5.8. We observe that the difference in performance between the different integral action implementations is small. In Setup A, the difference is practically indistinguishable, whereas in Setup B the implementation by Durand–Marchand is slightly worse than Årzén's forward difference and automatic reset. A possible explanation for why only minor differences are seen in Figure 5.8, is that the low-frequency behavior of the controller becomes less important when the process is subject to white noise disturbances of relatively high intensity.

Control Signal Generators

The fourth and final implementation aspect under study is the choice of control signal generator on the actuator side. We consider the following options:

(i) zero-order hold (ZOH) (as in Årzén's controller),

(ii) ZOH with adaptive gain,

(iii) PD generator (5.32), and

(iv) PID generator (5.33).

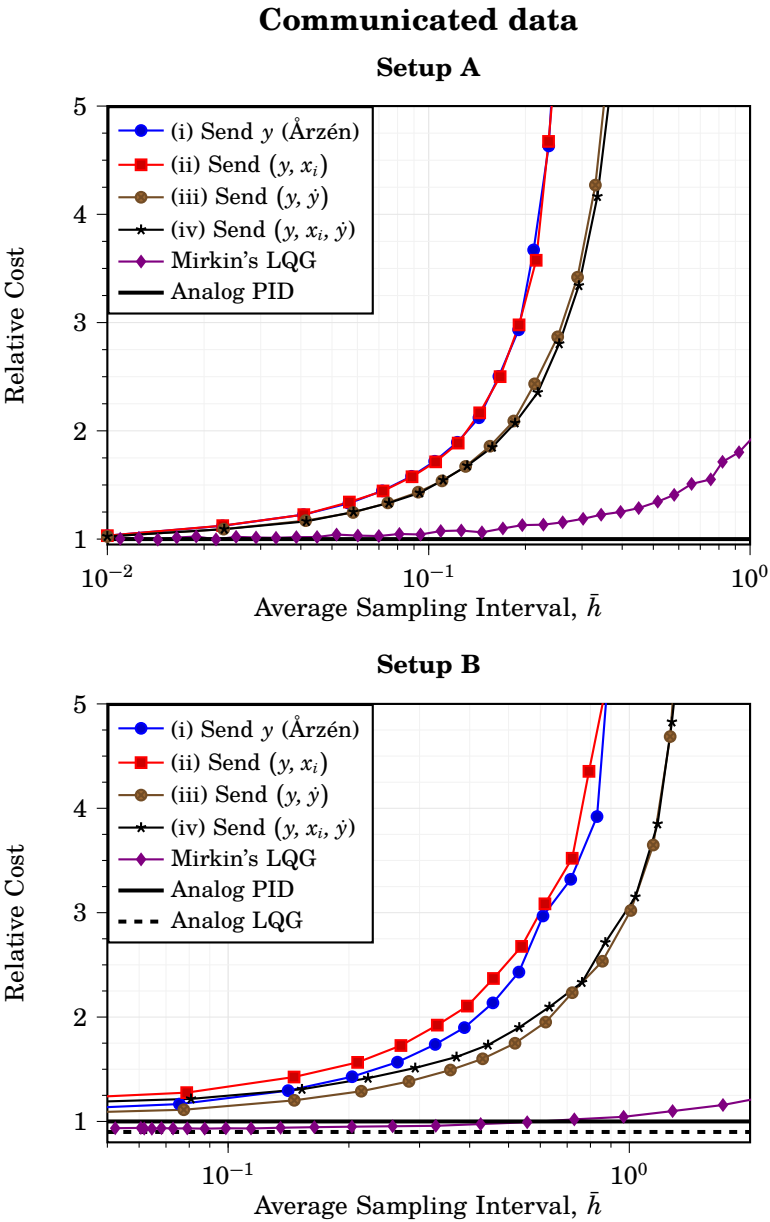

Figure 5.7 Relative LQG cost versus average sampling interval $\bar{h}$ for Setup A (upper) and B (lower) using different configurations of communicated data between the sensor and actuator node.

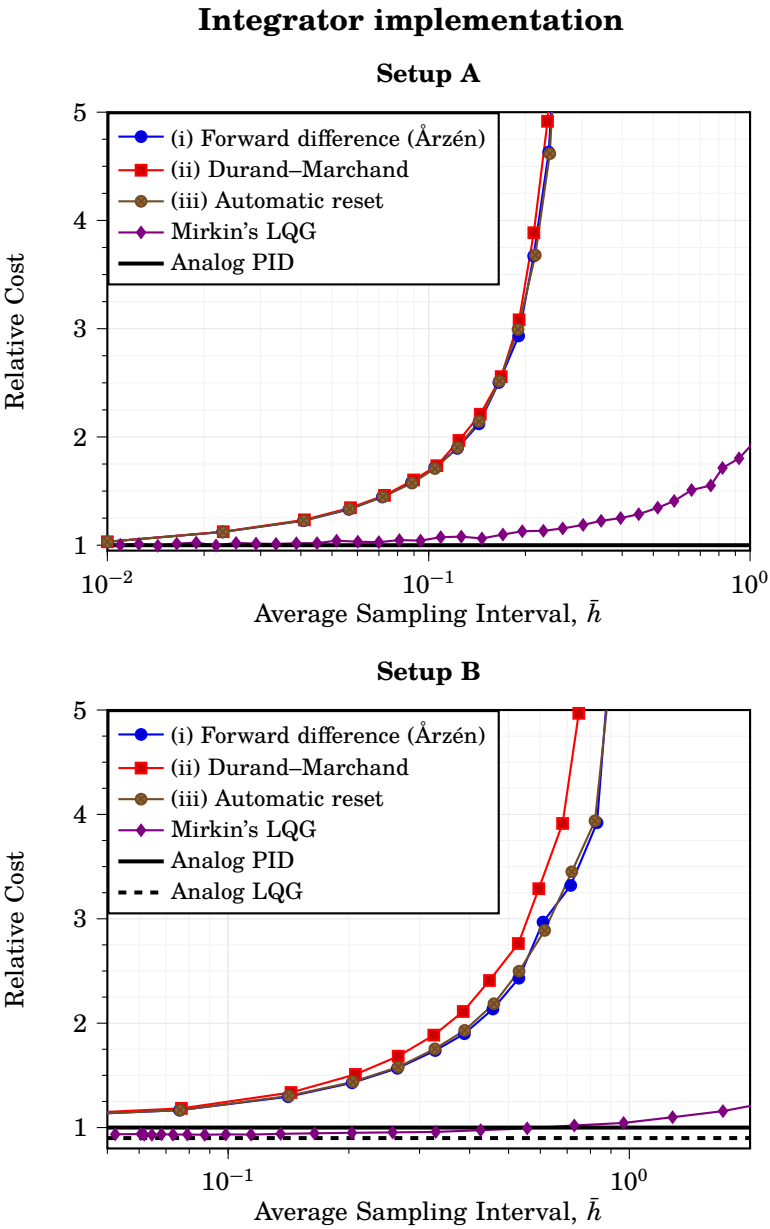

Figure 5.8 Relative LQG cost versus average sampling interval $\bar{h}$ for Setup A (upper) and B (lower) using different integrator implementations.

The simulated model in the PD and PID generators are based on the benchmark process model in both setups. In Setup A we use the exact model parameters, whereas in Setup B we obtain suitable model parameters via the PID parameters as detailed in Appendix B with $k = 0.75$, which in the corresponding benchmark model gives a phase margin of roughly 58°. Since the PD and PID generator ideally should be reset with analog signals, we even the field and transmit $(y, \dot{y})$ for implementations (i)–(iii) and $(y, x_i, \dot{y})$ for (iv) respectively.

The trade-off results for both setups are presented in Figure 5.9. In Setup A, we observe somewhat improved performance for large sampling intervals when using adaptive gain and the model-based signal generators. This is expected behavior, as it is only for longer intervals that the control signal from the adaptive gain and model-based signal generator start to deviate significantly from the baseline zero-order hold implementation by Årzén. However, we observe a different performance ranking for Setup B, where the adaptive gain and PD generator give very similar results to the baseline zero-order hold, and the PID generator has the worst performance of all the options for small sampling intervals. This behavior suggests that adaptive gain and signal generation based on the benchmark problem generalize poorly to other systems.

Conclusions and Final Evaluation

Based on the evaluations in the previous sections, we conclude the following:

- the main performance contributor is the inclusion of the derivative $\dot{y}$ in the sampling policy and transmitted data,
- performance seems largely insensitive to the implementation of the integral action, and
- the adaptive gain and control signal generators based on the benchmark problem give a moderate performance boost when applied to the matching process model (Setup A) but generalize poorly (Setup B).

Arguably, it is more worthwhile to keep the implementation model-free, rather than having the potential of an (at best) moderate performance boost for long sampling intervals. In a trade-off between simplicity and performance, we therefore propose an implementation based on Årzén's controller with the following modifications:

- PD triggering (5.26) for sampling policy, and
- $(y, \dot{y})$ for communicated data.

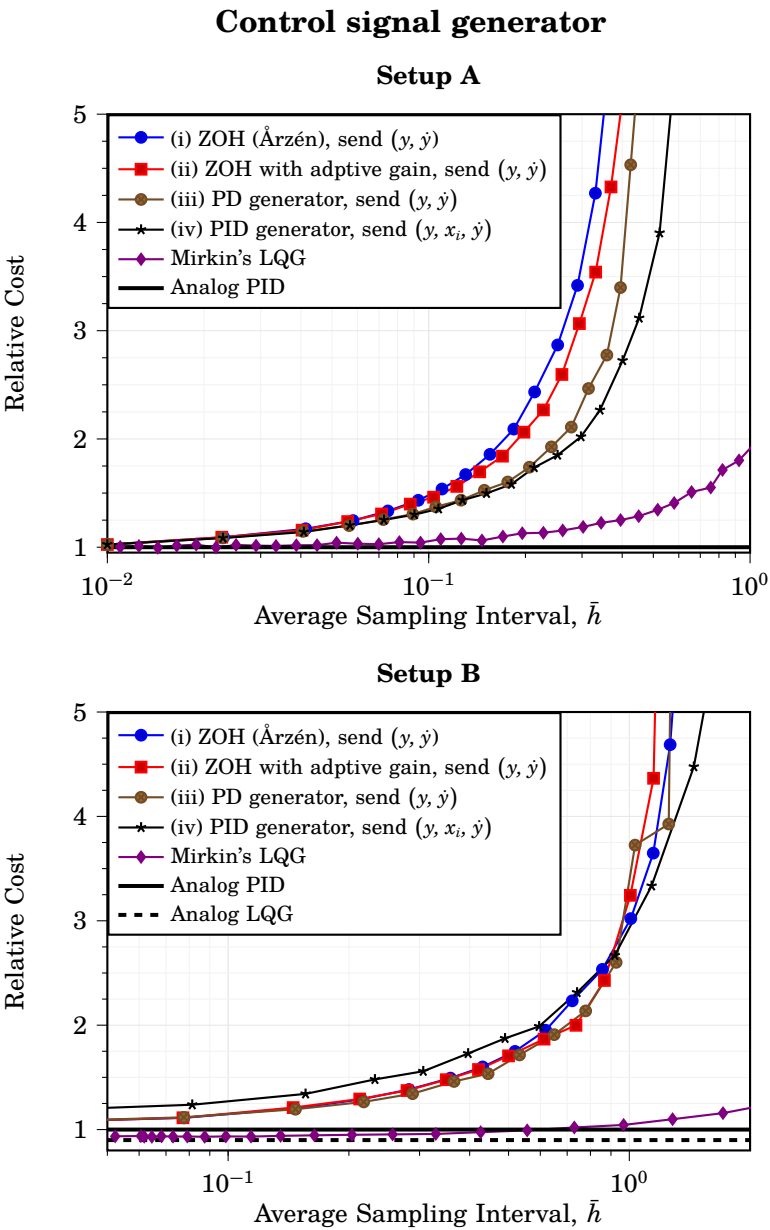

Figure 5.9 Relative LQG cost versus average sampling interval $\bar{h}$ for Setup A (upper) and B (lower) using different control signal generators.

In a final evaluation, we study the performance of this proposal. Since there was no clear winner between forward difference and automatic reset implementation of the integral action, we evaluate both options for the proposed controller above. The baseline periodic LQG controller from Section 2.4 is also included in the comparison for reference.

The trade-off results for both setups are shown in Figure 5.10. Drastic improvement is seen in both setups from the proposed modifications to Årzén's controller. In Setup A, both variants of the proposed implementation are even on par with the "ideal" event-based PID controller (Mirkin's LQG controller) (5.17) up until roughly $\bar{h} = 0.3$ s. The performance gap in the corresponding comparison in Setup B is expected since the PID structure is fundamentally suboptimal for this setup. Still, the proposed event-based PID controller with automatic reset outperforms the baseline periodic LQG implementation for large sampling intervals (roughly $\bar{h} \geq 0.3$ s).

Both variants of integral action seem to perform roughly the same in Setup A, with automatic reset showing slightly better performance for large sampling intervals. The improvement seen from automatic reset is even more pronounced in Setup B. These observations, combined with previously reported benefits in, e.g., [Tiberi et al., 2012], make automatic reset the preferred choice of integral action.

5.5 Summary

This chapter built upon the results in Chapters 2 and 3 and studied a benchmark problem for which the LQG-optimal controller is a PID controller. Based on Mirkin's LQG-optimal sampled-data controller, we studied what a theoretically "ideal" sampled-data implementation of a PID controller looks like. In turn, this inspired a range of design options for different implementations of event-based PID control, with complexity varying from Årzén's simple controller [Årzén, 1999] to the aforementioned "ideal" implementation.

A numerical evaluation of these design options suggested that a practical, yet well-performing, event-based PID controller should trigger sampling not only on the measured output y but also its derivative $\dot{y}$—here referred to as *PD triggering*. Furthermore, the sensor should also transmit both y and $\dot{y}$ to the controller at sampling times. Additional improvements for long sampling intervals were noted for integral action based on an automatic reset implementation.

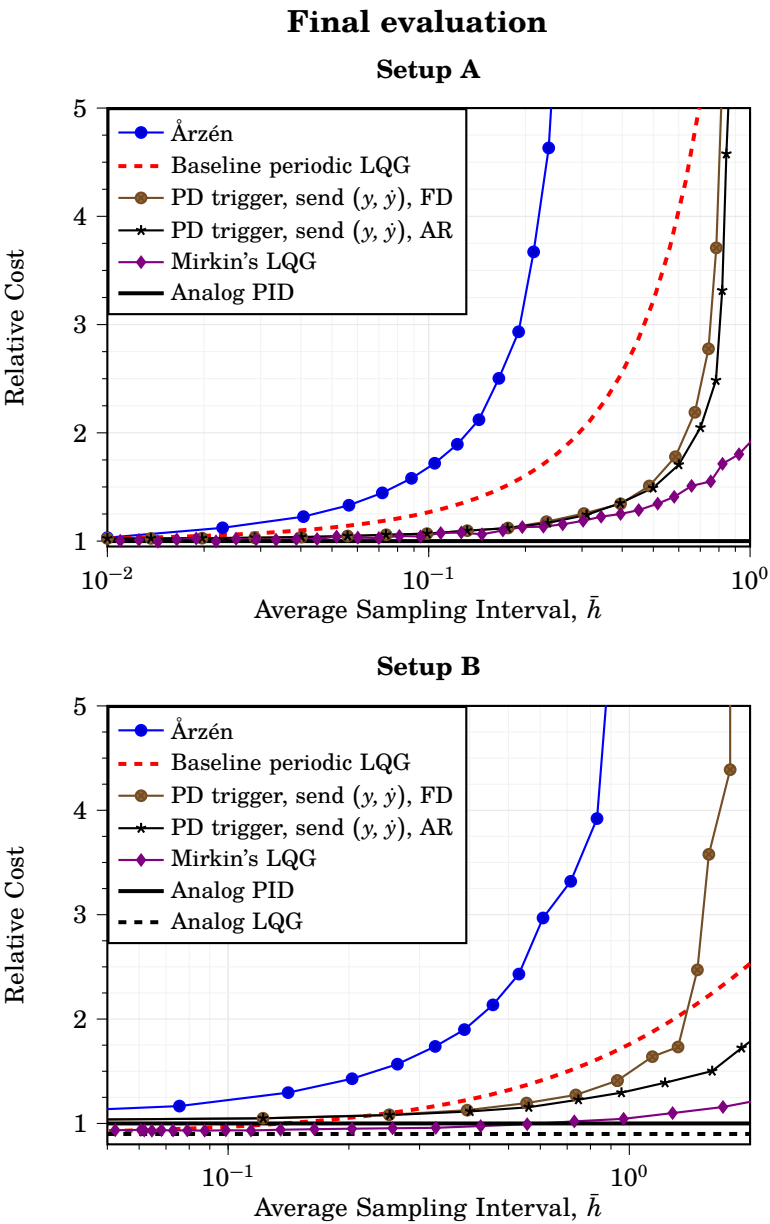

Figure 5.10 Relative LQG cost versus average sampling interval $\bar{h}$ for Setup A (upper) and B (lower) using the proposed event-based PID controller with integral action using forward difference (FD) and automatic reset (AR).

6

Stochastic Triggering in Event-Based State Estimation

This chapter considers a different type of event-based trigger policy—stochastic triggering. In contrast to the threshold-type policies considered so far, stochastic triggering assumes that events are triggered according to a certain probability. The probability of triggering an event is close to zero for small deviations from expected behavior, but transitions smoothly to one as deviations grow larger. Effectively, this can be viewed as a trigger policy with a "fuzzy" threshold.

Stochastic triggering is motivated by its potential for more tractable analysis. While deterministic threshold policies typically require Monte Carlo methods to estimate the state distribution, a stochastic policy can often provide useful bounds or even exact expressions. In particular, the stochastic approach has proven useful for event-based state estimation based on linear Gaussian models, where a suitable choice of trigger probability can preserve Gaussianity of the state distribution at all times [Han et al., 2015]. It is in this context that we consider stochastic triggering in this chapter.

The estimation problem in this chapter exhibits a degree of duality to the previously considered optimal sampling problem (OSP) (see Chapter 3). While triggering in the OSP is essentially based on deviations between open and closed-loop control actions, triggering in the estimation problem is based on deviations between monitored and predicted plant measurements. Both problems consider an LQG setup, where the quadratic cost in the estimation problem represents the estimation error covariance. However, unlike the OSP, a discrete-time setting is considered for the estimation problem. Furthermore, stochastic triggering enables us to employ a Bayesian approach for analysis rather than a Hamilton–Jacobi–Bellman formulation.

To the author's knowledge, the original framework of stochastic triggering for event-based state estimation was first introduced in [Han et al., 2015], with extended analysis in [Shi et al., 2016]. In the subsequent works [Thelander Andrén and Cervin, 2016; Schmitt et al., 2019; Thelander Andrén, 2020], several different policies were presented, aimed at improving the trigger policies in [Han et al., 2015]. This is mainly achieved by considering different predictions in the signal used for triggering. In this chapter, we present the general framework shared by the aforementioned works, and review the different proposed trigger policies. In particular, the contributions of this chapter are based on the work in [Thelander Andrén and Cervin, 2016; Thelander Andrén, 2020].

The chapter is structured as follows. In Section 6.1 we introduce the remote state estimation problem, where event-based triggering of data transmissions is considered. This is followed by a general derivation of the optimal state estimator under stochastic triggering in Section 6.2. The different trigger policies are described in Section 6.3, which are then evaluated and compared in a simulation study presented in Section 6.4. Finally, the chapter is summarized in Section 6.5.

6.1 The Remote State Estimation Problem

We study a prototypical remote state estimation problem in discrete time, see Figure 6.1. It involves a process subject to disturbances w, with an output signal corrupted by noise v and measured by a sensor as y. Based on the history of y, the sensor decides if new measurement data should be transmitted to a state estimator, situated remotely across a wireless network channel. Based on received measurements and knowledge of the sensor's policy for triggering transmissions, the state estimator produces the estimate $\hat{x}$ of the process state vector x. Transmissions should be triggered only when necessary since the radio in a wireless sensor typically stands for a large part of the total power consumption [Rault et al., 2014].

We assume a linear, discrete-time, state space model for the process:

$$\begin{aligned} x_{k+1} &= \Phi x_k + w_k, \\ y_k &= C x_k + v_k, \end{aligned} \tag{6.1}$$

where $x \in \mathbb{R}^n$ is the process state vector, $y \in \mathbb{R}^p$ is the measured output collected by the sensor, and $w \in \mathbb{R}^n$ and $v \in \mathbb{R}^p$ are uncorrelated white Gaussian processes with covariances $R_w \succeq 0$ and $R_v \succ 0$, respectively. The pair (Φ, C) is assumed to be observable.

The objective of the remote estimator is to produce the optimal estimate of the state vector x in the minimum mean square error (MMSE) sense. When the sensor transmits y every time step, the MMSE estimator for

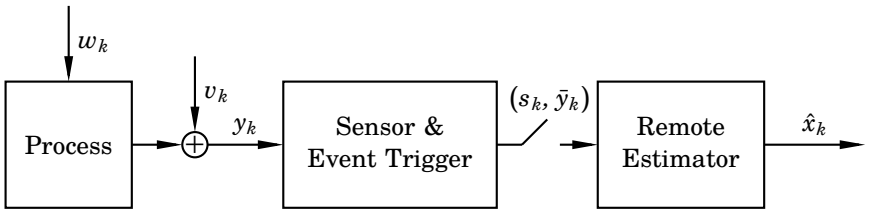

Figure 6.1 Setup for the remote state estimation problem. The estimate $\hat{x}$ of the internal state of the process are computed in the remote estimator based on intermittently communicated measurement information from the sensor.

the process in (6.1) is the standard Kalman filter [Kalman, 1960]. However, when transmissions occur sporadically based on a triggering rule, the problem instead becomes event-based.

Let $s_k \in \{0, 1\}$ denote the sensor's decision to transmit ($s_k = 1$) or not ($s_k = 0$), and let

$$\bar{y}_k \coloneqq \begin{cases} y_k, & \text{if transmit,} \\ \emptyset, & \text{otherwise,} \end{cases}$$

denote the measurement available to the remote estimator at time step k. Assuming that no data is lost in the transmission, the combined history of measurements and sensor decisions $\mathcal{I}_k$ available to the remote estimator at time k is

$$\mathcal{I}_k \coloneqq \Big\{(s_0, \bar{y}_0),\ (s_1, \bar{y}_1),\ \ldots,\ (s_k, \bar{y}_k)\Big\}.$$

Conditioned on $\mathcal{I}_k$, the MMSE estimate $\hat{x}$ and one-step prediction $\hat{x}^-$ of the state vector x at time k are given by

$$\begin{aligned} \hat{x}_k &\coloneqq \mathbb{E}\big[x_k \mid \mathcal{I}_k\big], \\ \hat{x}_k^- &\coloneqq \mathbb{E}\big[x_k \mid \mathcal{I}_{k-1}\big]. \end{aligned}$$

The covariances of the corresponding estimation and prediction errors are

$$\begin{aligned} P_k &\coloneqq \mathbb{E}\big[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T \mid \mathcal{I}_k\big], \\ P_k^- &\coloneqq \mathbb{E}\big[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T \mid \mathcal{I}_{k-1}\big]. \end{aligned}$$

In the standard Kalman filter, the data $(\hat{x}, P^-, P)$ is easily computed via closed-form recursive equations (see Section 2.4). By considering stochastic triggering, the same property can be retained also in the event-based case.

6.2 The Optimal Estimator under Stochastic Triggering

We proceed by introducing the mathematical framework behind stochastic triggering. After demonstrating how it can be used to preserve Gaussianity of the state estimates, we present the optimal state estimator under stochastic triggering.

Stochastic Triggering

The framework of stochastic triggering, as first presented in [Han et al., 2015], is as follows. At each time step k, the sensor samples a uniformly distributed random variable $\zeta \in [0, 1]$. The sampled value of ζ is then compared to the value of a trigger function $g(y-\mu) : \mathbb{R}^p \to [0, 1]$, whose argument is the difference between the measurement y and a prediction μ. The prediction is based on common information between both sensor and remote estimator, and should preferably require only minor computational effort from the sensor. With a sample ζ_k drawn at time k, the sensor decision is formed according to:

$$s_k = \begin{cases} 1, & \text{if } \zeta_k > g\big(y_k - \mu_k\big), \\ 0, & \text{otherwise.} \end{cases} \tag{6.2}$$

Under this scheme, the following property holds:

$$\begin{aligned} \Pr(s_k = 0) &= g\big(y_k - \mu_k\big), \\ \Pr(s_k = 1) &= 1 - g\big(y_k - \mu_k\big). \end{aligned}$$

This means that the output of the trigger function g equals the probability of *not* triggering a transmission. Intuitively, g should satisfy

$$\begin{aligned} g(y-\mu) &\to 1, \text{ as } \|y-\mu\| \to 0, \\ g(y-\mu) &\to 0, \text{ as } \|y-\mu\| \to \infty, \end{aligned}$$

i.e., the trigger probability should be small when the measurement y does not deviate significantly from the prediction μ.

Preserving Gaussianity

Now, consider a trigger function of the form

$$g(y-\mu) = \exp\Big(-\frac{1}{2}\|y-\mu\|_Y^2\Big), \tag{6.3}$$

i.e., a scaled Gaussian density function. The parameter $Y \succ 0$ in (6.3) is a design variable, which determines the decay rate of g. Assuming that the initial state x_0 has a Gaussian distribution, i.e.,

$$x_0 \sim \mathcal{N}(\hat{x}_0, P_0),$$

with known mean $\hat{x}_0 \in \mathbb{R}^n$ and covariance $P_0 \succeq 0$, we will show that the distribution of $x_k \mid \mathcal{I}_k$ remains Gaussian at all times k when stochastic triggering with (6.3) is used.

For the derivation, we adopt the following shorthand notation for the Gaussian probability density function (pdf):

$$G\big(x, \hat{x}, P\big) \coloneqq \frac{1}{\sqrt{(2\pi)^n \det(P)}} \exp\Big(-\frac{1}{2}\|x - \hat{x}\|^2_{P^{-1}} \Big). \tag{6.4}$$

Furthermore, we make use of the following two identities:

Lemma 6.1
For two Gaussian pdfs $G(x, \hat{x}, P)$ and $G(\bar{x}, \Phi x, R)$, it holds that

$$G\big(\bar{x}, \Phi x, R\big)G\big(x, \hat{x}, P\big) = G\big(x, d, D\big)G\big(\bar{x}, \Phi\hat{x}, \Phi P \Phi^\intercal + R\big), \tag{6.5}$$

where

$$\begin{aligned} D &\coloneqq \Big(P^{-1} + \Phi^\intercal R^{-1}\Phi\Big)^{-1}, \\ d &\coloneqq DP^{-1}\hat{x} + D\Phi^\intercal R^{-1}\bar{x}. \end{aligned}$$

Proof. The result is obtained by insertion of (6.4) into (6.5), and then completing the square of the exponent. □

Corollary 6.1
For two Gaussian pdfs $G(x, \hat{x}, P)$ and $G(\bar{x}, \Phi x, R)$, it holds that

$$\int_{\mathbb{R}^n} G\big(\bar{x}, \Phi x, R\big)G\big(x, \hat{x}, P\big)dx = G\big(\bar{x}, \Phi\hat{x}, \Phi P\Phi^\intercal + R\big). \tag{6.6}$$

Proof. The result is obtained by applying Lemma 6.1 to the product in (6.6) and noting that the resulting factor $G\big(\bar{x}, \Phi\hat{x}, \Phi P\Phi^\intercal + R\big)$ is independent of the integrand x. □

We now proceed with the derivation. First, consider the case when

$$x_{k-1} \mid \mathcal{I}_{k-1} \sim \mathcal{N}(\hat{x}_{k-1}, P_{k-1}), \tag{6.7}$$

where the mean $\hat{x}_{k-1} \in \mathbb{R}^n$ and covariance $P_{k-1} \succeq 0$ are known quantities. This holds by assumption for $k = 0$, and if we can show that

$$x_k \mid \mathcal{I}_k \sim \mathcal{N}(\hat{x}_k, P_k), \tag{6.8}$$

for some $\hat{x}_k$ and P_k, then Gaussianity holds for all $k \geq 0$ by induction.

By Bayes' theorem, the pdf of $x_k \mid \mathcal{I}_k$ satisfies

$$p(x_k \mid \mathcal{I}_k) = p(x_k \mid s_k, \bar{y}_k, \mathcal{I}_{k-1}) \propto p(s_k, \bar{y}_k \mid x_k) p(x_k \mid \mathcal{I}_{k-1}), \quad (6.9)$$

From (6.7) and Corollary 6.1, the prior distribution in (6.9) is given by

$$\begin{aligned} p(x_k \mid \mathcal{I}_{k-1}) &= \int_{\mathbb{R}^n} p(x_k \mid x_{k-1}) p(x_{k-1} \mid \mathcal{I}_{k-1}) dx_{k-1} \\ &= \int_{\mathbb{R}^n} G(x_k, \Phi x_{k-1}, R_w) G(x_{k-1}, \hat{x}_{k-1}, P_{k-1}) dx_{k-1} \\ &= G(x_k, \Phi \hat{x}_{k-1}, \Phi P_{k-1} \Phi^\intercal + R_w). \end{aligned} \quad (6.10)$$

We identify the MMSE one-step prediction and its covariance in (6.10) as

$$\begin{aligned} \hat{x}_k^- &= \mathbb{E}[x_k \mid \mathcal{I}_{k-1}] = \Phi \hat{x}_{k-1}, \\ P_k^- &= \mathbb{E}[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T \mid \mathcal{I}_{k-1}] = \Phi P_{k-1} \Phi^\intercal + R_w. \end{aligned}$$

We now proceed with the distribution of the measurement update in (6.9). In the case when a transmission is triggered ($s_k = 1$) we have

$$\begin{aligned} p(s_k = 1, \bar{y}_k \mid x_k) &= \Pr(s_k = 1 \mid y_k, x_k) p(y_k \mid x_k) \\ &= \Pr(s_k = 1 \mid y_k) p(y_k \mid x_k) \propto p(y_k \mid x_k) = G(y_k, Cx_k, R_v), \end{aligned} \quad (6.11)$$

where the measurement equation in (6.1) was used for the second equality. In the case of no transmission ($s_k = 0$) we have

$$\begin{aligned} p(s_k = 0, \bar{y}_k \mid x_k) &= \Pr(s_k = 0 \mid x_k) = \int_{\mathbb{R}^p} \Pr(s_k = 0 \mid y_k) p(y_k \mid x_k) d_{y_k} \\ &\propto \int_{\mathbb{R}^p} G(y_k, \mu_k, Y^{-1}) G(y_k, Cx_k, R_v) d_{y_k} = G(\mu_k, Cx_k, R_v + Y^{-1}), \end{aligned} \quad (6.12)$$

where Corollary 6.1 was used in the last equality.

Finally, by combining the two measurement updates (6.11) and (6.12) we see that (6.9) can be expressed as

$$p(x_k \mid \mathcal{I}_k) \propto G(s_k y_k + (1 - s_k)\mu_k, Cx_k, R_v + (1 - s_k)Y^{-1}) G(x_k, \hat{x}_k^-, P_k^-).$$

Using Lemma 6.1, the above expression can be rewritten as

$$p(x_k \mid \mathcal{I}_k) \propto G(x_k, \hat{x}_k, P_k), \quad (6.13)$$

where $\hat{x}_k$ and P_k are given by:

$$\begin{aligned} P_k &= \left((P_k^-)^{-1} + C^\intercal (R_v + (1 - s_k)Y^{-1})^{-1} C \right)^{-1} \\ \hat{x}_k &= P_k (P_k^-)^{-1} \hat{x}_k^- \\ &\quad + P_k C^\intercal (R_v + (1 - s_k)Y^{-1})^{-1} (s_k y_k + (1 - s_k)\mu_k). \end{aligned} \quad (6.14)$$

Since $G(x_k, \hat{x}_k, P_k)$ is already a normalized pdf, we in fact have equality in (6.13) and we have affirmed that (6.8) holds. By induction, Gaussianity therefore holds for all time steps $k \geq 0$, and the MMSE estimate and its covariance are given by (6.14).

The Optimal Estimator

We summarize the derivation of the previous section as follows:

THEOREM 6.1—THE MMSE ESTIMATOR
Consider a linear discrete-time process of the form in (6.1). Under the stochastic trigger policy (6.2) with a Gaussian trigger function (6.3), the MMSE estimator is given by the following recursive scheme:

$$\text{Time update:} \begin{cases} P_k^- = \Phi P_{k-1} \Phi^\intercal + R_w, \\ \hat{x}_k^- = \Phi \hat{x}_{k-1}, \end{cases}$$

$$\text{Measurement update:} \begin{cases} K_k = P_k^- C^\intercal \big(C P_k^- C^\intercal + R_v + (1 - s_k) Y^{-1} \big)^{-1}, \\ P_k = P_k^- - K_k C P_k^-, \\ \hat{x}_k = \hat{x}_k^- + K_k \big(s_k y_k + (1 - s_k) \mu_k - C \hat{x}_k^- \big). \end{cases}$$

Proof. The result follows from the derivation in the previous section, along with straightforward manipulations of the expressions for $\hat{x}_k$ and P_k in (6.14) using the Woodbury matrix identity. □

In Theorem 6.1 we see that the optimal remote estimator is a Kalman filter with an event-dependent measurement update. When the measurement y_k is transmitted ($s_k = 1$), the update follows the standard Kalman filter algorithm, whereas in the case of no transmission ($s_k = 0$), the update is made with the sensor prediction μ_k instead. The additional uncertainty in the latter case is reflected by the extra term Y^{-1} in the covariance of the estimate.

6.3 Sensor Prediction Policies

In Section 6.2 we introduced the prediction μ, which the sensor compares to the measurement y when deciding if a transmission should be triggered or not. If μ matches y well, the probability of triggering remains small, and the measurement update of the remote estimator is based on a value that is close to y. The opposite is true if μ is a poor prediction of y. Therefore, the choice of policy for producing the sensor prediction μ is critical for good estimator performance. We proceed in this section by giving brief reviews of the policies proposed in [Han et al., 2015; Shi et al., 2016; Schmitt et

al., 2019], along with more in-depth descriptions of the proposals based on [Thelander Andrén and Cervin, 2016; Thelander Andrén, 2020].

Open Loop

The simplest proposal for the sensor predictor, referred to as the *open loop* (OL) policy, was first studied in [Han et al., 2015]. In this policy, the sensor prediction is given by

$$\text{OL:} \quad \mu_k \coloneqq 0.$$

This simple choice is intended for stable systems, where

$$\lim_{k\to\infty} \mathbb{E}[y_k] = 0.$$

However, the OL policy is not suited for unstable systems. Owing to its simplicity, an exact expression for the average transmission rate under the OL policy is available, see [Han et al., 2015, Theorem 3].

Stochastic Send-on-Delta

Another simple proposal for the sensor predictor is the *stochastic send–on–delta* (SSOD) policy, which, to the author's knowledge, was first introduced in [Shi et al., 2016, Chapter 6]. In this policy, the sensor keeps track of the last transmitted measurement and uses it as its prediction. Assuming that the last transmission occurred $l \geq 1$ time steps ago, the prediction of the SSOD policy is given by

$$\text{SSOD:} \quad \mu_k \coloneqq y_{k-l}.$$

Like the OL policy, the SSOD policy requires no computations in the sensor and offers a reasonable prediction for slow-varying processes and short durations l. However, the prediction generally becomes poor as l grows.

Stochastic Send-on-Delta with a Simple Prediction

To strike a balance between simplicity and prediction quality, we propose an SSOD policy modified with a simple model-based prediction. This proposal, denoted SSODP, was first presented in [Thelander Andrén and Cervin, 2016] and uses a pre-computed look-up table of factors $\{S_l\}$ to rescale y_{k-l} based on the model dynamics. The predictions are of the form

$$\text{SSODP:} \quad \mu_k \coloneqq S_l y_{k-l}.$$

Since the scaling factors are computed based on the assumption of a stationary state distribution, the SSODP is only applicable for stable systems. In the following, we derive the scaling factors $\{S_l\}$.

Derivation of Scaling Factors. Consider an l step prediction of y_k based on the last transmitted measurement y_{k-l} according to

$$\mu_k = \mathbb{E}\big[y_k \mid y_{k-l}\big]. \tag{6.15}$$

To find an expression for (6.15), we introduce the following lemma from [Åström, 1970, Chapter 7, Theorem 3.2]:

LEMMA 6.2
Let x and y be two jointly Gaussian vectors. The distribution of $x \mid y$ is then Gaussian, with expectation

$$\mathbb{E}\big[x \mid y\big] = \mathbb{E}[x] + P_{xy}P_y^{-1}\Big(y - \mathbb{E}[y]\Big),$$

where

$$P_{xy} \coloneqq \mathbb{E}\big[(x - \mathbb{E}[x])(y - \mathbb{E}[y])^\intercal\big],$$
$$P_y \coloneqq \mathbb{E}\big[(y - \mathbb{E}[y])(y - \mathbb{E}[y])^\intercal\big].$$

Proof. See [Åström, 1970, Chapter 7, Theorem 3.2]. □

Since y_k and y_{k-l} are jointly Gaussian, Lemma 6.2 gives

$$\mathbb{E}\big[y_k \mid y_{k-l}\big] = \mathbb{E}[y_k] + P_{y_k y_{k-l}}P_{y_{k-l}}^{-1}\Big(y_{k-l} - \mathbb{E}[y_{k-l}]\Big). \tag{6.16}$$

Assuming that the process (6.1) is stable, and that the state vector has converged to its stationary distribution, we have

$$\mathbb{E}[y_k] = \mathbb{E}[y_{k-l}] = 0,$$

which reduces (6.16) to

$$\mathbb{E}\big[y_k \mid y_{k-l}\big] = P_{y_k y_{k-l}}P_{y_{k-l}}^{-1}y_{k-l} \coloneqq S_l y_{k-l}. \tag{6.17}$$

Under the same assumption, we can also obtain closed-form expressions for the matrices $P_{y_k y_{k-l}}$ and $P_{y_{k-l}}$, which together form the scaling factor $S_l = P_{y_k y_{k-l}}P_{y_{k-l}}^{-1}$. Starting with $P_{y_k y_{k-l}}$, we have

$$P_{y_k y_{k-l}} = \mathbb{E}[y_k y_{k-l}^\intercal] = \mathbb{E}\big[(Cx_k + v_k)(Cx_{k-l} + v_{k-l})^\intercal\big] = C\mathbb{E}[x_k x_{k-l}^\intercal]C^T$$
$$= C\mathbb{E}\Big[\big(\sum_{j=1}^{l}\Phi^{j-1}w_{k-j} + \Phi^l x_{k-l}\big)x_{k-l}^\intercal\Big]C^\intercal = C\Phi^l\mathbb{E}[x_{k-l}x_{k-l}^\intercal]C^\intercal \coloneqq C\Phi^l\Sigma C^\intercal.$$

Here $\Sigma \succeq 0$ denotes the stationary covariance of the state vector, which satisfies the discrete-time Lyapunov equation

$$\Sigma = \Phi\Sigma\Phi^\intercal + R_w. \tag{6.18}$$

Using similar calculations, the matrix $P_{y_{k-l}}$ is obtained as

$$P_{y_{k-l}} = \mathbb{E}[y_{k-l}y_{k-l}^\intercal] = \mathbb{E}\big[(Cx_{k-l} + v_{k-l})(Cx_{k-l} + v_{k-l})^\intercal\big] = C\Sigma C^T + R_v.$$

In summary, the set of scaling factors $\{S_l\}$ used in the SSODP policy is given by

$$S_l = C\Phi^l \Sigma C^\intercal (C\Sigma C^T + R_v)^{-1}. \tag{6.19}$$

As the set $\{S_l\}$ is known beforehand, a look-up table can be computed offline and stored in the sensor for improved efficiency. In practice, the exact value of S_l can for sufficiently large l be approximated by the limit

$$\lim_{l\to\infty} S_l = 0.$$

Relation to the OL and SSOD Policies. From (6.19), we deduce that fast-decaying processes (i.e., when the eigenvalues of Φ are of small magnitude) with high-intensity measurement noise (i.e., the eigenvalues of R_v are of large magnitude) achieve $S_l \approx 0$ after relatively few time steps l. The reason is that the correlation between y_{k-l} and y_k rapidly approaches zero with increasing l, and the SSODP policy then essentially coincides with the OL policy.

In contrast, when there is a strong positive correlation between y_k and y_{k-l}, we have $S_l \approx I_p$, and the SSODP policy coincides with the SSOD policy. This occurs for slow-decaying, non-resonant processes (i.e., when the eigenvalues of Φ are positive, real, and close to 1) with measurement noise of relatively small magnitude.

From this, we note that there are processes where neither OL nor SSOD are particularly well-suited. In particular, none of them are well-suited for resonant systems, where the correlation between y_k and y_{k-l} periodically becomes negative. However, resonant systems pose no problem to the SSODP policy as the negative correlation is taken into account in the set of scaling factors $\{S_l\}$.

Stochastic Send-on-Delta with a FIR-Based Prediction

A clear limitation of the SSODP policy is its requirement of a stable process model. This limitation was tackled in [Schmitt et al., 2019], where the authors expanded upon the SSODP policy by introducing a finite impulse response (FIR) prediction. This policy, denoted SSODP-FIR, use a sensor prediction of the form

$$\text{SSODP-FIR:} \quad \mu_k \coloneqq C\Phi^{l+1}\hat{x}_{k-l}^{\text{FIR}}, \tag{6.20}$$

where $\hat{x}_{k-l}^{\text{FIR}}$ denotes a FIR-based estimate of x_{k-l}. The estimate is computed in the sensor at times of transmission, and is based on the solution of a

weighted least-squares problem using the m last measured outputs from the process according to

$$\hat{x}_k^{\text{FIR}} = (M^\intercal V^{-1} M)^{-1} M^\intercal V^{-1} \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-(m-1)} \end{bmatrix},$$

where $M \in \mathbb{R}^{pm \times n}$ is given by

$$M \coloneqq \begin{bmatrix} C \\ C\Phi^{-1} \\ \vdots \\ C\Phi^{-(m-1)} \end{bmatrix},$$

and $V \in \mathbb{R}^{pm \times pm}$ have the block entries

$$V_{i,j} \coloneqq C\Big(\sum_{k=1}^{\min(i,j)-1} \Phi^{-(i-k)} R_w (\Phi^{-(j-k)})^\intercal \Big) C^\intercal + \delta_{i,j} R_v,$$

where $i, j \in \{1, \ldots, m\}$ and $\delta_{i,j}$ denotes the Kroenecker delta.

At each transmission, the tuple $(y_k, \hat{x}_k^{\text{FIR}})$ is transmitted from the sensor to the remote estimator. By including $\hat{x}_k^{\text{FIR}}$ in the transmission, the estimator receives information from the last m measured outputs from the process, including measurements collected between events. Since (6.20) is an unbiased prediction of y_k, the SSODP-FIR policy is useful also for unstable processes. However, since the matrices M and V depend on Φ^{-1}, the SSODP-FIR policy is not directly applicable for process models with pure time-delays (i.e., when Φ has one or more zero eigenvalues). The number m of buffered measurements in the sensor is chosen as a trade-off between computational load and performance, and in the limit $m \to \infty$ the estimate $\hat{x}^{\text{FIR}}$ produced by the sensor is equivalent to that of the standard Kalman filter.

Closed Loop

Another policy introduced in [Han et al., 2015] is the *closed loop* policy. The name comes from the fact that it requires bidirectional communication between the sensor and remote estimator. The sensor prediction used by the CL policy is the one-step prediction of the remote estimator, i.e,

$$\text{CL:} \quad \mu_k \coloneqq C\hat{x}_k^-.$$

The proposed implementation in [Han et al., 2015] requires transmissions from the remote estimator to the sensor at each time instant. Arguably, this goes against the motivation of event-based transmissions, as it prohibits the sensor from powering down its radio in between events.

Closed Loop with Buffered Measurements

Inspired by the previous policies, we propose a new policy that combines the predictions of the CL policy with the buffered measurements of the SSODP-FIR policy. This policy, referred to as *closed loop with measurement buffer* (CLMB) [Thelander Andrén, 2020], also uses the prediction of the remote estimator in the sensor, i.e,

$$\text{CLMB:} \quad \mu_k \coloneqq C\hat{x}_k^-.$$

However, in contrast to [Han et al., 2015], we propose that transmissions from the remote estimator to the sensor only take place at events, i.e., when the sensor takes the initiative to communicate. Since bidirectional communication only takes place on the sensor's initiative, it can then safely power down its radio in between events. When communication takes place at a time k, the remote estimator transmits a predicted trajectory $\{C\hat{x}_{k+1}^-, \ldots, C\hat{x}_{k+m}^-\}$ of length m, computed under the assumption that no transmissions will be received for the coming m time steps. If no transmissions have been triggered after m time steps, the sensor triggers a new transmission regardless. To the author's knowledge, this approach for remote state estimation was first proposed in [Ruuskanen and Cervin, 2020] and is a simple solution for avoiding a constantly powered on radio or a local copy of the remote estimator in the sensor.

So far, the CLMB and CL policies essentially only differ in the proposed implementation and are mathematically equivalent in the limit $m \to \infty$. The second part of the CLMB policy is the addition of a measurement buffer in the sensor. The sensor then stores up to a maximum of m measurements, and then transmits the entire buffer $\{y_k, y_{k-1}, \ldots, y_{k-(m-1)}\}$ to the remote estimator. The remote estimator then updates its estimate based on the full measurement history—in effect reproducing the state estimation of the standard Kalman filter at transmission times. This means that the estimates of the CLMB policy only differ from those of the standard Kalman filter during the intervals between transmission events.

Choosing the Trajectory Length. The fact that the CLMB policy coincides with the standard Kalman filter at events is also useful for analysis. In particular, it allows us to compute the probability of triggering a transmission within a given time interval, which can guide the choice of m.

Let P_{kf}^- denote, which satisfies the discrete-time ARE [Åström and Wittenmark, 1997, Theorem 11.6]

$$P_{\text{kf}}^- = \Phi P_{\text{kf}}^- \Phi^\intercal + R_w - \Phi P_{\text{kf}}^- C^\intercal \left(C P_{\text{kf}}^- C^\intercal + R_v\right)^{-1} C P_{\text{kf}}^- \Phi^\intercal.$$

Since the CLMB policy coincides with the standard Kalman filter at events, the estimate and prediction covariances of the two policies are identical

immediately after a transmission. Let P_l^- denote the prediction covariance of the CLMB policy after l time steps since the last transmission. We thus have

$$P_l^- = \begin{cases} P_{\text{kf}}^-, & l = 1, \\ \Phi(P_{l-1}^- - K_{l-1} C P_{l-1}^-)\Phi^\intercal + R_w, & l \geq 2, \end{cases} \tag{6.21}$$

where

$$K_l \coloneqq P_l^- C^\intercal (C P_l^- C^\intercal + R_v + Y^{-1})^{-1}. \tag{6.22}$$

Now, let $\mathcal{I}$ denote the transmission history available at the time of the last event. With some abuse of notation, the probability of triggering a transmission at the l^{th} time step since the last event is given by

$$\begin{aligned} &\Pr(s_l = 1, s_{1:l-1} = 0 \mid \mathcal{I}) \\ &= \Pr(s_l = 1 \mid s_{1:l-1} = 0, \mathcal{I}) \Pr(s_{l-1} = 0 \mid s_{1:l-2} = 0, \mathcal{I}) \dots \Pr(s_1 = 0 \mid \mathcal{I}). \end{aligned} \tag{6.23}$$

The conditional probabilities in (6.23) are of the form

$$\begin{aligned} \Pr\big(s_l = 0 \mid s_{1:l-1} = 0, \mathcal{I}\big) &= \int_{\mathbb{R}^p} \Pr\big(s_l = 0 \mid y_l\big) p\big(y_l \mid s_{1:l-1} = 0, \mathcal{I}\big) dy_l \\ &= \sqrt{(2\pi)^n \det(Y^{-1})} \int_{\mathbb{R}^p} G\big(y_l, C\hat{x}_l^-, Y^{-1}\big) G\big(y_l, C\hat{x}_l^-, C P_l^- C^\intercal + R_v\big) dy_l \\ &= \frac{1}{\sqrt{\det\big(I_p + (C P_l^- C^\intercal + R_v) Y\big)}} \coloneqq q(P_l^-). \end{aligned} \tag{6.24}$$

Thus, by using (6.24) we obtain the probability (6.23) for $l \geq 2$ as

$$\Pr(s_l = 1, s_{1:l-1} = 0 \mid \mathcal{I}) = \big(1 - q(P_l^-)\big) \prod_{j=1}^{l-1} q(P_j^-), \tag{6.25}$$

and for $l = 1$:

$$\Pr(s_1 = 1 \mid \mathcal{I}) = 1 - q(P_1^-).$$

Finally, the probability of triggering a transmission within m time steps is given by the sum

$$\Pr(s_l = 1, s_{1:l-1} = 0, 1 \leq l \leq m \mid \mathcal{I}) = \sum_{l=1}^{m} \Pr(s_l = 1, s_{1:l-1} = 0 \mid \mathcal{I}). \tag{6.26}$$

By evaluating (6.26) for different values of m, we see how probable it is that the sensor is forced to trigger a transmission due to a full measurement buffer. Ideally, the probability in (6.26) should be close to 1, but the resulting data size of each transmission and computational load on the remote estimator also has to be taken into account.

Average Transmission Rate and Mean-Square Error. The analysis in the previous section can also be used to find expressions for the average transmission rate and mean square error (MSE) of the CLMB policy. The MSE is given by $\operatorname{tr}(\bar{P})$, where $\bar{P}$ is the asymptotic covariance of the estimation error

$$\bar{P} \coloneqq \lim_{k\to\infty} \mathbb{E}\big[P_k\big]. \tag{6.27}$$

Similarly, the average transmission rate f is defined as

$$f \coloneqq \lim_{k\to\infty} \mathbb{E}[s_k]. \tag{6.28}$$

The key observation for computing (6.27) and (6.28) for the CLMB policy, is that the set of possible covariance matrices and interval durations between events is finite. As in the previous section, let $1 \leq l \leq m$ denote the number of time steps since the last transmission event, and define

$$P_l \coloneqq \begin{cases} P_{\mathrm{kf}}, & l = 1, \\ P_l^- - K_l C P_l^-, & 2 \leq l \leq m, \end{cases}$$

where P_{kf} denotes the stationary estimation error covariance of the standard Kalman filter, and P_l^- and K_l where defined in (6.21) and (6.22), respectively. The transitions between different states of covariance can then be modeled as a Markov chain. The transition probabilities are given by (6.24) for $1 \leq l < m$, while for $l = m$ we trigger a transmission with probability 1. This results in the following transition matrix:

$$M_{\mathrm{trans}} = \begin{bmatrix} 1-q(P_1^-) & q(P_1^-) & 0 & \cdots & 0 \\ 1-q(P_2^-) & 0 & q(P_2^-) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1-q(P_{m-1}^-) & 0 & 0 & \cdots & q(P_{m-1}^-) \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

This Markov chain is both irreducible and aperiodic, which guarantees the existence of a stationary distribution $\bar{\pi} = \begin{pmatrix} \pi_1 & \dots & \pi_m \end{pmatrix}$, satisfying

$$\bar{\pi} M_{\mathrm{trans}} = \bar{\pi}. \tag{6.29}$$

After solving (6.29), the asymptotic covariance $\bar{P}$ under the CLMB policy is found as

$$\bar{P} = \lim_{k\to\infty} \mathbb{E}\big[P_k\big] = \mathbb{E}\big[P_l\big] = \sum_{j=1}^{m} \pi_j P_j. \tag{6.30}$$

The stationary distribution $\bar{\pi}$ also lets us compute the average transmission rate as

$$f = \lim_{k\to\infty} \mathbb{E}[s_k] = \mathbb{E}[s_l] = \mathbb{E}\big[\mathbb{E}[s_l \mid s_{1:l-1} = 0, \mathcal{I}]\big]$$
$$= \mathbb{E}\big[\Pr(s_l \mid s_{1:l-1} = 0, \mathcal{I})\big] = \mathbb{E}[1 - q(P_l^-)] = \sum_{j=1}^{m} \pi_j \big(1 - q(P_j^-)\big), \quad (6.31)$$

where the law of total expectation was used in the third equality.

6.4 Performance Comparison

With a range of trigger policies available from Section 6.3, we proceed by evaluating and comparing their respective performances in a simulation study. Two of the presented policies, SSODP-FIR and CLMB, have an additional parameter m that represents the maximum buffer size in the sensor. To simplify comparisons, we let $m \to \infty$ in this study. As noted in [Schmitt et al., 2019], the performance of the SSODP-FIR policy then coincides with that of the CL policy.

The study is divided into two parts. In the first part, we evaluate estimation performance over four second-order systems, each an example of a stable/unstable and resonant/damped process. In the second part, we consider estimation performance and feedback control of a simulated flexible servo process—a resonant fourth-order process commonly used in teaching of control theory.

For each policy, we consider the trade-off between MSE, $\mathrm{tr}(\bar{P})$, and average transmission rate f. These quantities are evaluated via (6.30) and (6.31) for the CLMB policy, while simulations are used for the other policies to obtain estimates according to

$$\bar{P} \approx \frac{1}{N-1} \sum_{k=1}^{N} \big(x_k - \hat{x}_k\big)\big(x_k - \hat{x}_k\big)^\mathsf{T},$$
$$f \approx \frac{1}{N} \sum_{k=1}^{N} s_k,$$

where N is the total number of time steps in the simulation. Unless otherwise stated, we use $N = 10^5$. Furthermore, the MSE is normalized in all results such that the standard Kalman filter (KF) has a relative MSE of 1.

Second-Order Systems

Setup. Here we consider the performance trade-off for a set of four second-order systems, categorized according to stable/unstable and reso-

nant/damped dynamics:

(i) stable, critically damped:

$$\Phi = \begin{bmatrix} 0.8 & 0.3 \\ 0 & 0.8 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad R_w = \begin{bmatrix} 1.0 & 0.7 \\ 0.7 & 0.5 \end{bmatrix}, \quad R_v = 1,$$

(ii) unstable, critically damped:

$$\Phi = \begin{bmatrix} 1.0 & 0.3 \\ 0 & 1.0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad R_w = \begin{bmatrix} 1.0 & 0.7 \\ 0.7 & 0.5 \end{bmatrix}, \quad R_v = 1,$$

(iii) stable, resonant:

$$\Phi = \begin{bmatrix} -0.85 & -0.35 \\ 0.35 & -0.85 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad R_w = \begin{bmatrix} 0.01 & 0 \\ 0 & 1. \end{bmatrix}, \quad R_v = 0.1,$$

(iv) unstable, resonant:

$$\Phi = \begin{bmatrix} 1.1 & -0.35 \\ 0.35 & 1.1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad R_1 = \begin{bmatrix} 0.01 & 0 \\ 0 & 1. \end{bmatrix}, \quad R_2 = 0.1.$$

As previously noted in Section 6.3, the OL and SSODP policies are not applicable for unstable systems and are therefore omitted for systems (ii) and (iv) above.

Results and Discussion. Trade-off results are presented in Figure 6.2 for systems (i) and (ii) (critically damped systems), and in Figure 6.3 for systems (iii) and (iv) (resonant systems) respectively. We observe that the OL and SSOD policies, which both lack a time-varying prediction in the sensor, give the overall poorest trade-offs. For the stable systems (upper plots in Figures 6.2 and 6.3), we note that the OL policy performs better than the SSOD policy in the resonant case, while the opposite is true in the critically damped case. This behavior is explained by the fact that zero, rather than the last transmitted measurement, is a better prediction on average of a measurement signal y which is oscillating around the origin. Better still is the simple prediction used by the SSODP policy, which outperforms both the OL and SSOD policies.

For systems (ii) and (iv) (lower plots in Figures 6.2 and 6.3), we immediately see that SSOD is not suitable for unstable systems and that a model-based prediction in the sensor is required. We see a drastic improvement for the CL/SSODP-FIR and CLMB policies, where predicted trajectories are transmitted from the remote estimator to the sensor. By also buffering measurements in the sensor, the CLMB policy gains slightly better performance compared to the CL/SSODP-FIR policies, especially for the unstable resonant system (iv) in the lower plot of Figure 6.3.

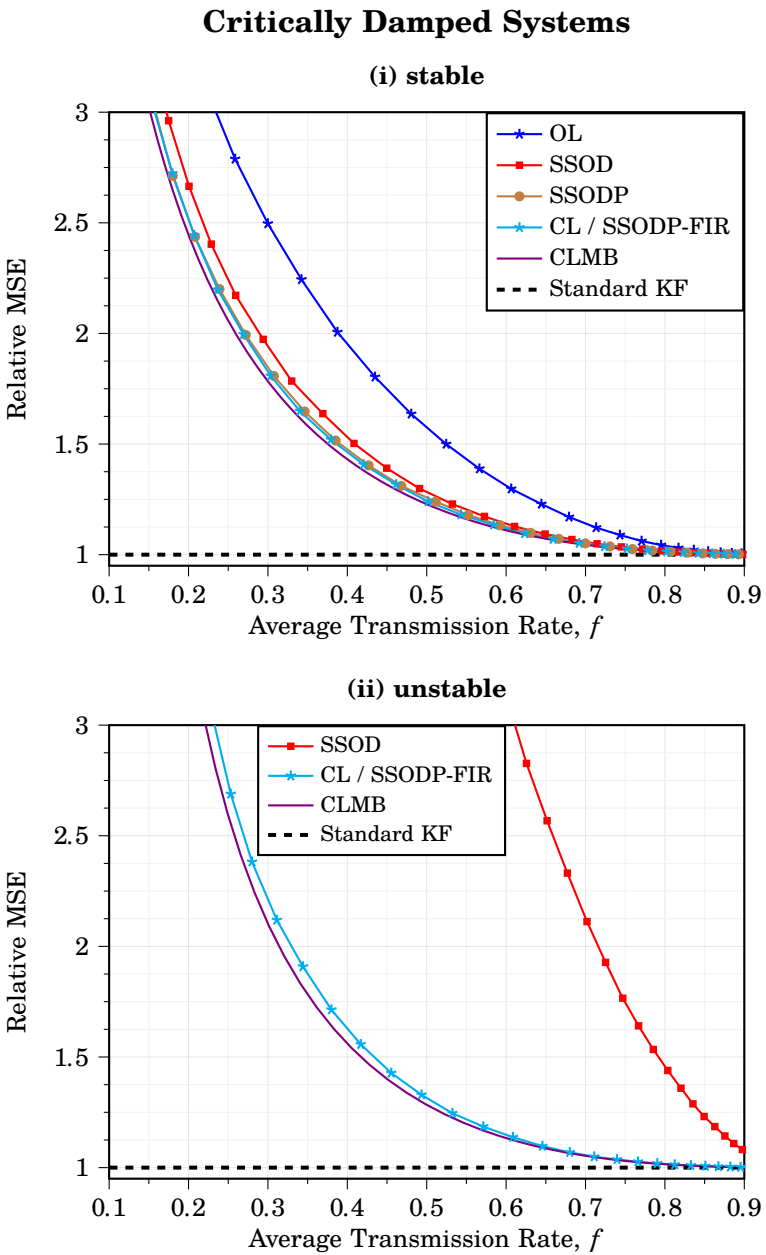

Figure 6.2 Relative MSE versus average transmission rate f for the stable (upper) and unstable (lower) critically damped second-order systems.

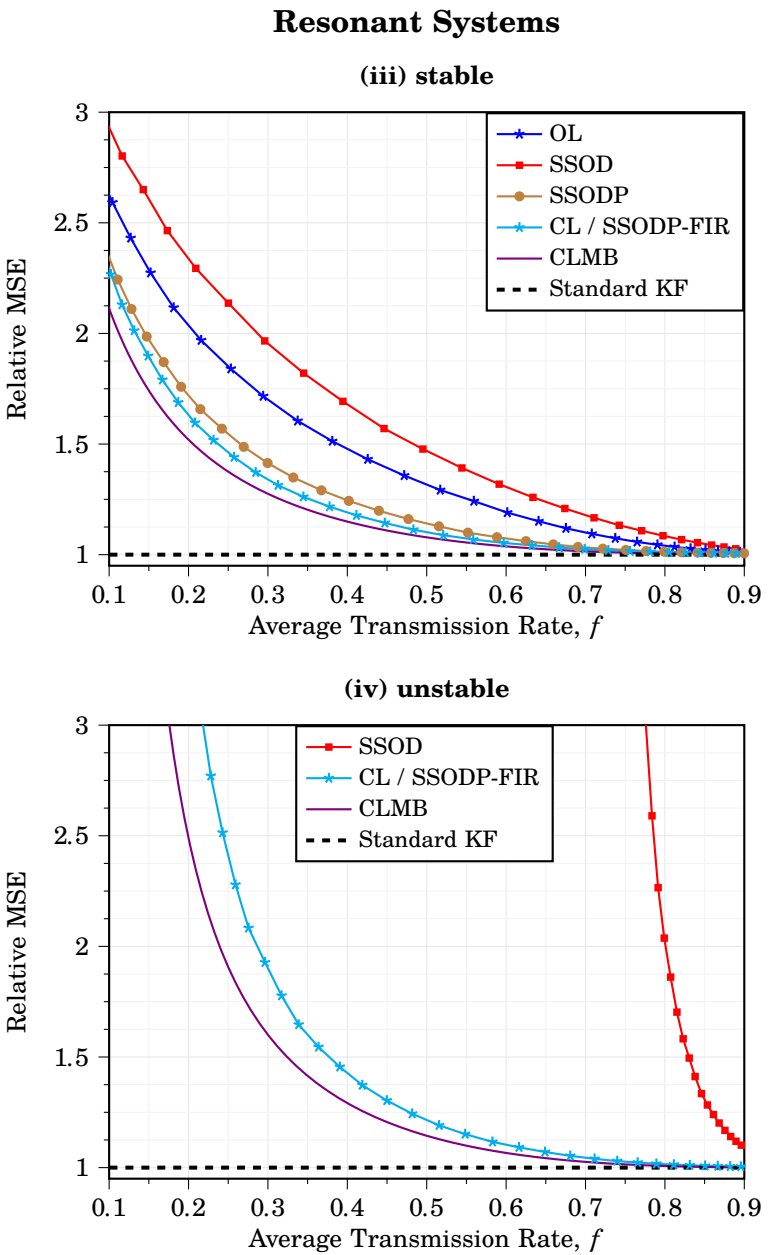

Figure 6.3 Relative MSE versus average transmission rate f for the stable (upper) and unstable (lower) resonant second-order systems.

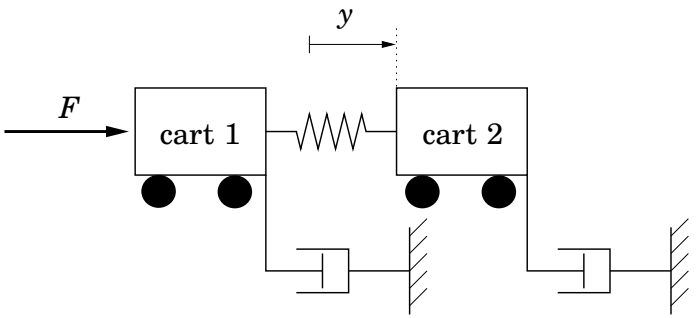

Figure 6.4 The flexible servo process.

Conclusions. The conclusions are summarized as follows:

- the OL and SSOD policies have the overall poorest performance, while the CLMB policy has the overall best performance,
- just as the OL and SSODP policies, the SSOD policy is not suitable for unstable systems,
- for stable systems, the SSODP policy performs almost as well as the more complex SSODP-FIR policy and the bidirectional CL policy.

Flexible Servo

Setup. We now consider state estimation and feedback control of a flexible servo process, illustrated in Figure 6.4. The process consists of two interconnected carts moving on a track. The interconnection is flexible, and a damper is connected to each cart. A voltage u is supplied to a DC-motor, which in turn exerts a force F to the first cart. The available sensor measurement y from the process is the position of the second cart.

A representative LTI model, corresponding to a nominal time step of $\Delta t = 0.01$ s, is given by

$$\begin{aligned} x_{k+1} &= \Phi x_k + \Gamma u_k + w_k, \\ y_k &= C x_k + v_k, \end{aligned} \tag{6.32}$$

with parameters

$$\Phi = \begin{bmatrix} 1 & 0.01 & 0.01 & 0 \\ -1.72 & 0.98 & 1.72 & 0.01 \\ 0.01 & 0 & 1 & 0.01 \\ 1.93 & 0.01 & -1.93 & 0.97 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0 \\ 0.01 \\ 0 \\ 0 \end{bmatrix}^{\mathsf{T}}, \quad C = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^{\mathsf{T}}.$$

The state vector $x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^\mathsf{T}$ corresponds to the position and velocity of the first (x_1 and x_2) and second cart (x_3 and x_4) respectively, given in units $[m]$ and $[ms^{-1}]$. Covariances of w_k and v_k are given by

$$R_w = 10^{-6} \times \begin{bmatrix} 0.99 & -0.85 & 0.01 & 0.96 \\ -0.85 & 2.98 & 0.86 & -2.23 \\ 0.01 & 0.86 & 0.99 & -0.96 \\ 0.96 & -2.23 & -0.96 & 3.48 \end{bmatrix}, \quad R_v = 10^{-6}.$$

Trade-Off Comparison. First, we consider the trade-off between MSE and average transmission rate f for the uncontrolled process (i.e., $u = 0$ in (6.32)). Since the flexible servo is a marginally stable process, the SSODP policy can not be applied directly. Instead, the SSODP design is based on a slightly modified version of (6.32), where the marginally stable pole has been perturbed to produce an asymptotically stable process model. For a small perturbation, we have $S_l \approx 1$ for all l, and the SSODP policy thus coincides with the SSOD policy.

The trade-off results for the uncontrolled flexible servo process are shown in Figure 6.5. We observe that the OL policy is performing significantly worse than the other policies, while the SSOD(P) policy performs only slightly worse than the CL/SSODP-FIR policies. As in the previous examples, the CLMB policy provides the best performance, with a clear improvement over the CL/SSODP-FIR policies for low trigger rates.

Feedback Control Based on Event-Based State Estimation. Based on the trade-off results, we now consider the SSOD(P), CL/SSODP-FIR, and CLMB policies in a final example, where we combine a linear state feedback controller with the state estimates provided by each policy. The state feedback gain is given by

$$F = -\begin{bmatrix} 59.9 & 9.5 & -30.2 & 2.2 \end{bmatrix},$$

and is designed to give a damped closed-loop system. For evaluation, we consider the response of the closed-loop system to two input impulse disturbances, representing two sudden pushes to the first cart. The response based on feedback from a standard Kalman filter is used as a nominal case, which is used for comparisons with the corresponding responses for the event-based policies.

The closed-loop system is simulated over a 12 s interval for each policy, corresponding to a total of 1 200 discrete time steps. The two impulse disturbances are applied at 3 s and 6 s, respectively. For each event-based policy, the trigger parameter Y is tuned such a total of 100 events are triggered, i.e., roughly 8 % of the transmissions used in the nominal case. The sequence of measurement noise is identical in all simulations.

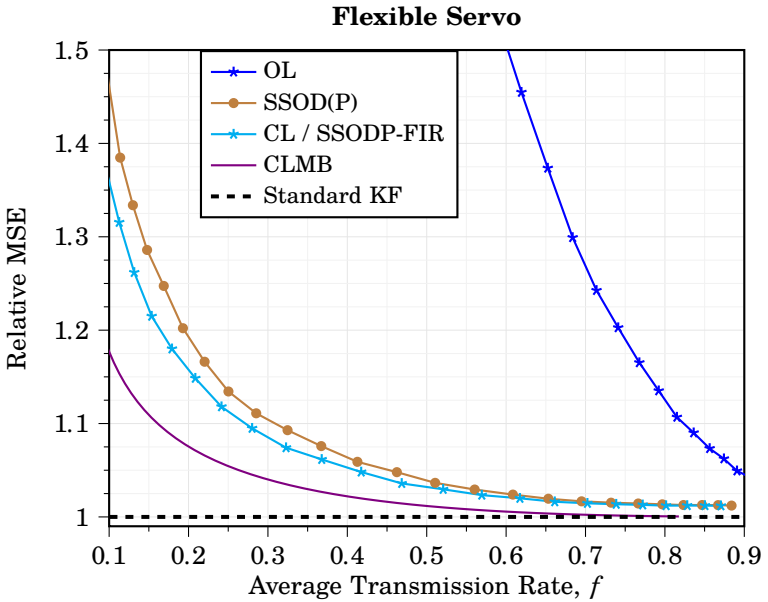

Figure 6.5 Relative MSE versus average transmission rate f for the flexible servo system.

The results are presented in Figure 6.6, where the impulse response for each state estimator policy is seen in the upper plot. Deviations from the nominal response are strikingly small for all event-based policies—in the order of a few millimeters—which is seen in the box plot in the lower part of Figure 6.6. That only minor deviations are seen—despite only using 8 % of the available transmissions—suggests that there is a large redundancy of transmissions in the nominal case. Only minor differences are observed between the responses of the event-based policies, and they all essentially perform equally well in this example.

6.5 Summary

This chapter considered stochastic trigger policies for a prototypical event-based remote estimation problem. By choosing a stochastic policy in the form of a scaled Gaussian function, the conditional distribution of the process state remains Gaussian even when no new measurements are transmitted. This enables a straightforward derivation of the MMSE estimator, which has the form of a Kalman filter with an event-dependent measurement update.

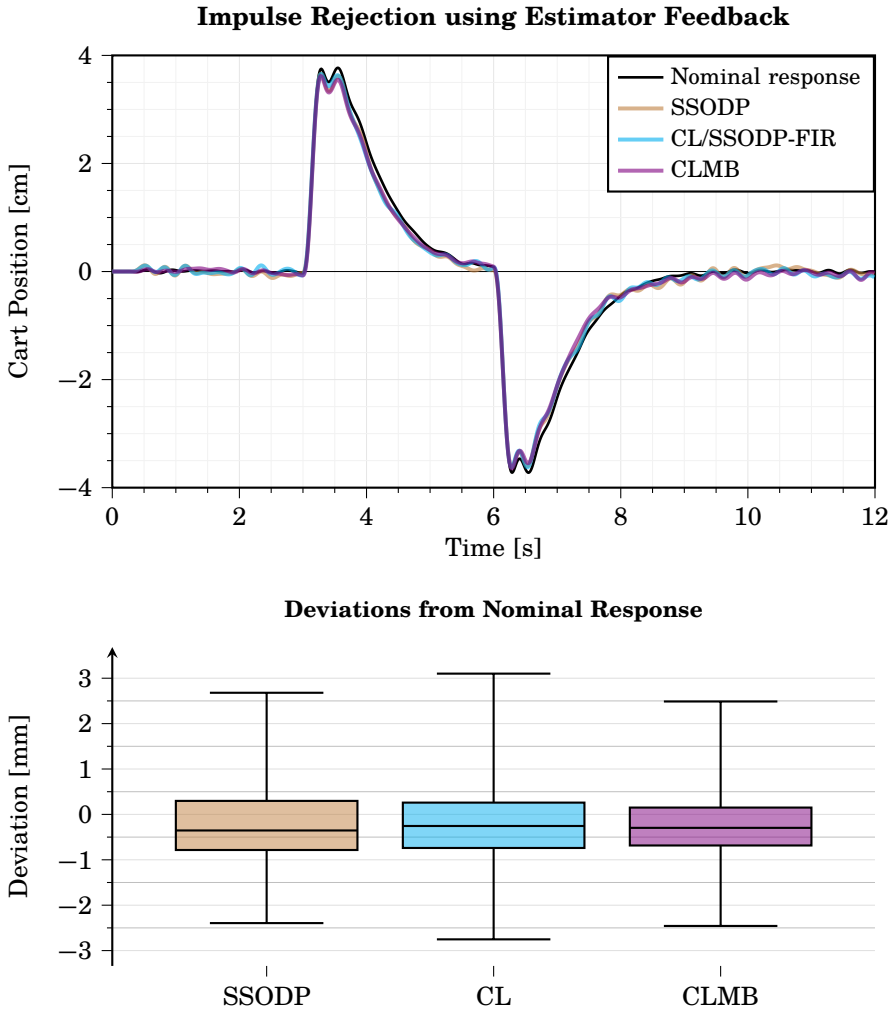

Figure 6.6 Impulse rejection for the flexible servo system with feedback based on different state estimators, showing the time responses (upper) and distribution of deviations from the nominal response (lower). In the box plot of the distributions, the lower, mid, and upper line in each box corresponds to the first, second (median), and third quartile, respectively, while the whiskers correspond to the minimum and maximum. The nominal response is produced with a standard Kalman filter, while the other responses are based on state estimates from the SSODP, CL/SSODP-FIR, and CLMB policies. All three event-based estimators have been tuned such that they trigger 100 times out of 1 200 time steps during the simulations.

For the MMSE estimator, we considered different policies for sensor prediction—a key part in achieving a good trade-off between MSE and average transmission rate. We propose two new policies in this thesis; stochastic send-on-delta with a simple prediction (SSODP) and closed loop with a measurement buffer (CLMB). The SSODP policy expands upon previously proposed policies by incorporating a simple prediction based on a steady-state analysis of the process model. By taking the process dynamics into account, the SSODP is capable of dealing with systems where other simple policies perform poorly, such as resonant systems.

The CLMB policy is a useful alternative when bidirectional transmission between the sensor and remote estimator is allowed. By letting the sensor buffer and transmit the full measurement history between events, the estimates of the CLMB policy coincide with those of the standard Kalman filter after each transmission from the sensor. In addition to performance benefits, this property also results in simpler analysis, with expressions for average transmission rate and MSE that are straightforward to compute.

In a simulation study, we compared the trade-off between MSE and transmission rate of the aforementioned policies with other policies from the literature. The results favored the CLMB policy in all examples, with varying margins of improvement over the other policies. Compared to other policies of similar complexity, clear performance improvement was also noted for the SSODP policy.

7

Conclusion

This chapter concludes the thesis with a summary and proposed directions for future research.

7.1 Summary

In this thesis, we have considered the design of trigger policies for event-based control and state estimation in an LQG setting. Based on the LQG-optimal sampled-data controller (referred to as Mirkin's LQG controller), we posed the problem of finding the corresponding optimal trigger policy for sampling. Solutions to this optimal sampling problem (OSP) were derived analytically for two new system types; spiral-type systems, and single-input and single-output systems of certain structure. The optimal event trigger was shown to be an ellipsoidal threshold for both system types. Tight bounds on optimal performance relative to periodic triggering were also derived for multivariate integrator systems. These bounds show that the well-known factor three improvement derived in [Åström and Bernhardsson, 1999] for a first-order integrator system constitutes an upper bound in the higher-order case. Moreover, the corresponding lower bound is decreasing with increasing system order, with equal performance of periodic and optimal event-based triggering in the limit.

Two numerical methods were proposed for solving the OSP for general linear systems; a simulation-based method and a linear complementarity method. The simulation-based method was derived using a time-dependent formulation of the Hamilton–Jacobi–Bellman equation for the OSP, suitable for standard PDE tools. The linear complementarity method was proposed as an alternative that avoids introducing time-dependence. Both methods were validated numerically and compared to a standard value iteration algorithm for reference. The simulation-based method showed a better trade-off between accuracy and solver time compared to value iteration, while the linear complementarity method outperformed both of the other methods

in terms of accuracy. However, both value iteration and the two proposed methods suffer from the curse of dimensionality, restricting their use to low-order systems.

The shape of the optimal event trigger threshold was investigated using the linear complementarity method on different classes of second-order systems. Findings showed that most second-order systems have an optimal threshold that is almost ellipsoidal. The notable exception is saddle-point systems, which have non-convex optimal thresholds. For this reason, we compared performance between triggering based on an ellipsoidal threshold and the optimal threshold for a saddle-point system. The performance results were practically indistinguishable, which further motivates the use of ellipsoidal trigger functions. A numerical validation demonstrated that suitable ellipse parameters for higher-order systems can be obtained via the Monte Carlo approach in [Becker et al., 2019].

Inspired by the LQG-optimal sampled-data controller, we also studied an "ideal" (in the LQG sense) sampled-data PID implementation. Based on this study, a range of design options of varying complexity for event-based PID control were proposed. A numerical study lead to the conclusion that so-called *PD triggering*—where events are triggering based on both the measured output and its derivative—is a useful option for a better trade-off between control performance and trigger rate.

Finally, we considered stochastic triggering in a prototypical remote estimation problem. With a stochastic trigger in the form of a scaled Gaussian function, the minimum mean square error (MMSE) estimator takes the form of a Kalman filter with event-dependent measurement update. For this MMSE estimator, we proposed two event triggers; stochastic send-on-delta with a simple prediction (SSODP) and closed loop with a measurement buffer (CLMB). The SSODP policy extends previously proposed trigger policies by incorporating a simple prediction based on a steady-state analysis of the process model. This simple prediction makes the SSODP policy capable of handling systems where other policies of similar complexity perform poorly, such as resonant systems. The CLMB policy was proposed as a high-performance alternative for cases when bidirectional transmissions between the sensor and remote estimator are allowed. By letting the sensor buffer and transmit the full measurement history between events, the estimates of the CLMB policy coincide with those of the standard Kalman filter at event times. In addition to performance benefits, this property also results in simpler analysis, with expressions for average transmission rate and estimator performance that are straightforward to evaluate.

7.2 Directions for Future Research

For future research on LQG-optimal event-based control, we propose three main directions:

- event triggering based on ellipsoidal thresholds,
- framework extensions aimed at sporadic disturbances, and
- incorporation of event-triggered learning.

Ellipsoidal Threshold Policies

We argue that ellipsoidal threshold policies are a useful and more practical alternative to the optimal one—especially for higher-order systems, where numerical methods for obtaining the optimal policy break down. While the Monte Carlo approach described in Section 4.6 can be used to find (almost) optimal ellipse parameters, we still believe that further improvements are possible. As seen in the derivation in [Becker et al., 2019], the approach was developed for a very general setting. However, the OSP comes with strong model assumptions due to the LQG setup. Most likely, these assumptions can be exploited in the Monte Carlo approach, either to evaluate some involved subexpressions exactly or to find suitable control variates for variance reduction.

Another direction that may lead to simpler optimization is to consider stochastic triggering of the form presented in Chapter 6. Stochastic triggering based on a Gaussian function has trigger probabilities with ellipsoidal level sets, and it can thus be interpreted as a "fuzzy" version of an ellipsoidal threshold policy. The benefit of this approach is that the state distribution remains Gaussian at all times, making it is possible to find exact expressions for the LQG cost and average sampling interval—similar to the analysis for the CLMB policy in Section 6.3. If optimization over these expressions can be performed efficiently, a deterministic threshold can then be recovered from the optimal stochastic policy by choosing a suitable trigger probability level set.

Sporadic Disturbances

Disturbances are assumed to continuously perturb the controlled process in the LQG framework considered in this thesis. Intuitively, however, we expect event-based strategies to be most advantageous for processes where disturbances only occur sporadically. An example of this was observed in the flexible servo study in Section 6.4. Due to the sporadic nature of the disturbances (two impulses), event-based triggering resulted in a closed-loop behavior very similar to that of periodic triggering, while triggering only 8 % as often.

One potential approach for incorporating sporadic disturbances is the framework proposed in [Antunes, 2013]. Disturbances are then modeled as impulses of random magnitude, arriving according to a Poisson process. The "sporadicity" of the disturbance process is then determined by the Poisson rate. This sporadic disturbance model is a generalization of the one considered in this thesis since both models coincide as the Poisson rate goes to infinity. Due to this link, it may be possible to extend the results presented in this thesis for sporadic disturbances using this model.

Another approach is the framework of linear-exponential-quadratic-Gaussian (LEQG) control, see [Jacobson, 1973; Duncan, 2013]. As the name implies, the framework considers the exponential of the standard LQG objective. The LEQG objective function also includes a scalar design variable that determines the exponential rate. Large values of the exponential rate variable result in a more *risk-averse* control design, with a controller more prone to rejecting large, sporadically occurring, disturbances from the tails of the Gaussian distribution. On the other hand, as the exponential rate variable approaches zero, the LEQG objective coincides with the standard, *risk-neutral*, LQG objective. This explicit connection to the standard LQG objective suggests that it may be possible to extend the work in this thesis to the LEQG case.

Event-Triggered Learning

It should be noted that the framework considered in this thesis is based on the availability of a representative process model. However, if model parameters are chosen poorly, there is no guarantee that actual performance will be close to that we expect from the model. A way to deal with this problem could be to include event-triggered learning in the closed-loop system, see [Solowjow et al., 2018; Baumann et al., 2019]. In this approach, expected behavior from the model is compared to data collected from the actual process. Whenever the model behavior differs significantly from the data, an identification experiment is triggered to update the model's parameters. If online redesign of the event-based controller can be achieved, then this approach could offer the benefits of both model-based and data-driven strategies. To achieve fast redesign of the event trigger, a suboptimal parametrization, such as the aforementioned ellipsoidal threshold, is most likely the best choice.

Bibliography

Anta, A. and P. Tabuada (2010). "To sample or not to sample: Self-triggered control for nonlinear systems". *IEEE Transactions on Automatic Control* **55**:9, pp. 2030–2042.

Antunes, D., W. P. M. H. Heemels, and P. Tabuada (2012). "Dynamic programming formulation of periodic event-triggered control: Performance guarantees and co-design". In: *Proceedings of the 51st IEEE Conference on Decision and Control*. Maui, HI, USA, pp. 7212–7217.

Antunes, D. J. and B. A. Khashooei (2016). "Consistent event-triggered methods for linear quadratic control". In: *Proceedings of the 55th IEEE Conference on Decision and Control*. Las Vegas, NV, USA, pp. 1358–1363.

Antunes, D. (2013). "Event-triggered control under poisson events: The role of sporadicity". *IFAC Proceedings Volumes* **46**:27, pp. 269–276.

Antunes, D. and W. Heemels (2014). "Rollout event-triggered control: Beyond periodic control performance". *IEEE Transactions on Automatic Control* **59**:12, pp. 3296–3311.

Aranda-Escolástico, E., M. Guinaldo, R. Heradio, J. Chacon, H. Vargas, J. Sánchez, and S. Dormido (2020). "Event-based control: A bibliometric analysis of twenty years of research". *IEEE Access* **8**, pp. 47188–47208.

Årzén, K.-E. (1999). "A simple event-based PID controller". *IFAC Proceedings Volumes* **32**:2, pp. 8687–8692.

Åström, K. J. and R. M. Murray (2008). *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press.

Åström, K. J. (2008). "Event based control". In: Astolfi, A. et al. (Eds.). *Analysis and Design of Nonlinear Control Systems*. Springer, pp. 127–147.

Åström, K. J. (1970). *Introduction to Stochastic Control Theory*. Vol. 70. Mathematics in Science and Engineering. Academic Press.

Åström, K. J. and B. Bernhardsson (1999). "Comparison of periodic and event based sampling for first-order stochastic systems". *IFAC Proceedings Volumes* **32**:2, pp. 5006–5011.

Åström, K. J. and T. Hägglund (2006). *Advanced PID Control*. The Instrumentation, Systems and Automation Society.

Åström, K. J. and B. Wittenmark (1997). *Computer-Controlled Systems*. 3rd ed. Prentice-Hall.

Baumann, D., J. Zhu, G. Martius, and S. Trimpe (2018). "Deep reinforcement learning for event-triggered control". In: *Proceedings of the 57th IEEE Conference on Decision and Control*. Miami, FL, USA, pp. 943–950.

Baumann, D., F. Solowjow, K. H. Johansson, and S. Trimpe (2019). "Event-triggered pulse control with model learning (if necessary)". In: *Proceedings of the 2019 American Control Conference*. Philadelphia, PA, USA, pp. 792–797.

Becker, S., P. Cheridito, and A. Jentzen (2019). "Deep optimal stopping". *Journal of Machine Learning Research* **20**, pp. 1–25.

Bertsekas, D. P. (2012). *Dynamic Programming and Optimal Control, Vol. II*. 4th ed. Athena Scientific.

Bezanson, J., A. Edelman, S. Karpinski, and V. B. Shah (2017). "Julia: A fresh approach to numerical computing". *SIAM Review* **59**:1, pp. 65–98.

Björk, T. (2019). *Arbitrage Theory in Continuous Time*. 4th ed. Oxford University Press.

Braksmayer, M. and L. Mirkin (2017). "H_2 optimization under intermittent sampling and its application to event-triggered control". In: *Proceedings of the 20th IFAC World Congress*. Toulouse, France, pp. 7869–7874.

Cervin, A. (2016). "LQG-optimal PI and PID control as benchmarks for event-based control". In: *Proceedings of the 2nd International Conference on Event-based Control, Communication, and Signal Processing*. Kraków, Poland, pp. 1–8.

Cervin, A. and M. Thelander Andrén (2020). "LQG-optimal versus simple event-based PID controllers". In: *Proceedings of the 2020 American Control Conference*. Denver, CO, USA, pp. 3678–3684.

Cervin, A. and T. Henningsson (2008). "Scheduling of event-triggered controllers on a shared network". In: *Proceedings of the 47th IEEE Conference on Decision and Control*. Cancún, Mexico, pp. 3601–3606.

Chen, T. and B. Francis (1994). *Optimal Sampled-Data Control Systems*. Communications and Control Engineering. Springer.

Cogill, R. (2009). “Event-based control using quadratic approximate value functions”. In: *Proceedings of the 48th IEEE Conference on Decision and Control*. Shanghai, China, pp. 5883–5888.

Cogill, R., S. Lall, and J. P. Hespanha (2007). “A constant factor approximation algorithm for event-based sampling”. In: *Proceedings of the 2007 American Control Conference*. New York, NY, USA, pp. 305–311.

Davar, S. and A. Mohammadi (2017). “Event-based particle filtering with point and set-valued measurements”. In: *Proceedings of the 25th European Signal Processing Conference*. Kos, Greece, pp. 211–215.

Dimarogonas, D. V., E. Frazzoli, and K. H. Johansson (2011). “Distributed event-triggered control for multi-agent systems”. *IEEE Transactions on Automatic Control* **57**:5, pp. 1291–1297.

Dorf, R., M. Farren, and C. Phillips (1962). “Adaptive sampling frequency for sampled-data control systems”. *IRE Transactions on Automatic Control* **7**:1, pp. 38–47.

Duncan, T. E. (2013). “Linear-exponential-quadratic gaussian control”. *IEEE Transactions on Automatic Control* **58**:11, pp. 2910–2911.

Durand, S. and N. Marchand (2009a). “Further results on event-based PID controller”. In: *Proceedings of the 2009 European Control Conference*. Budapest, Hungary, pp. 1979–1984.

Durand, S. and N. Marchand (2009b). “An event-based PID controller with low computational cost”. In: *Proceedings of the 8th International Conference on Sampling Theory and Applications*. Marseille, France.

Ellis, P. (1959). “Extension of phase plane analysis to quantized systems”. *IRE Transactions on Automatic Control* **4**:2, pp. 43–54.

Evans, L. C. (1998). *Partial Differential Equations*. Vol. 19. Graduate Studies in Mathematics. American Mathematical Society.

Fornberg, B. and N. Flyer (2015). “Solving PDEs with radial basis functions”. *Acta Numerica* **24**, pp. 215–258.

Friedman, A. (1982). *Variational Principles and Free-Boundary Problems*. Dover.

Gatsis, K., A. Ribeiro, and G. J. Pappas (2016). “State-based communication design for wireless control systems”. In: *Proceedings of the 55th IEEE Conference on Decision and Control*. Las Vegas, NV, USA, pp. 129–134.

Gatsis, K., A. Ribeiro, and G. J. Pappas (2014). “Optimal power management in wireless control systems”. *IEEE Transactions on Automatic Control* **59**:6, pp. 1495–1510.

Glad, T. and L. Ljung (2000). *Control Theory : Multivariable and Nonlinear Methods*. Taylor & Francis.

Goldenshluger, A. and L. Mirkin (2017). "On minimum-variance event-triggered control". *IEEE Control System Letters* **1**:1, pp. 32–37.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press.

Goodwin, G. C., J. I. Yuz, C. Agüero, and M. Cea (2010). "Sampling and sampled-data models". In: *Proceedings of the 2010 American Control Conference*. Baltimore, MD, USA, pp. 1–20.

Han, D., Y. Mo, J. Wu, S. Weerakkody, B. Sinopoli, and L. Shi (2015). "Stochastic event-triggered sensor schedule for remote state estimation". *IEEE Transactions on Automatic Control* **60**:10, pp. 2661–2675.

Heemels, W. P. M. H., M. C. F. Donkers, and A. R. Teel (2013). "Periodic event-triggered control for linear systems". *IEEE Transactions on Automatic Control* **58**:4, pp. 847–861.

Heemels, W. P. M. H., K. H. Johansson, and P. Tabuada (2012). "An introduction to event-triggered and self-triggered control". In: *Proceedings of the 51st IEEE Conference on Decision and Control*. Maui, HI, USA, pp. 3270–3285.

Henningsson, T. (2011). "Sporadic event-based control using path constraints and moments". In: *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*. Orlando, FL, USA, pp. 4723–4728.

Henningsson, T., E. Johannesson, and A. Cervin (2008). "Sporadic event-based control of first-order linear stochastic systems". *Automatica* **44**:11, pp. 2890–2895.

Henningsson, T. (2012). *Stochastic Event-Based Control and Estimation*. PhD thesis. Department of Automatic Control, Lund University, Sweden.

Henriksson, D., A. Cervin, and K.-E. Årzén (2002). "TrueTime: Simulation of control loops under shared computer resources". *IFAC Proceedings Volumes* **35**:1, pp. 417–422.

Hernández-Lerma, O. (1989). *Adaptive Markov Control Processes*. Vol. 79. Applied Mathematical Sciences. Springer.

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

Hsia, T. (1972). "Comparisons of adaptive sampling control laws". *IEEE Transactions on Automatic Control* **17**:6, pp. 830–831.

Hutchinson, I. H. (2015). *A Student's Guide to Numerical Methods*. Cambridge University Press.

Innes, M., E. Saba, K. Fischer, D. Gandhi, M. C. Rudilosso, N. M. Joy, T. Karmali, A. Pal, and V. Shah (2018). "Fashionable modelling with Flux". arXiv: 1811.01457.

Jacobson, D. (1973). "Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games". *IEEE Transactions on Automatic Control* **18**:2, pp. 124–131.

Kabamba, P. (1987). "Control of linear systems using generalized sampled-data hold functions". *IEEE Transactions on Automatic Control* **32**:9, pp. 772–783.

Kalman, R. E. (1960). "A new approach to linear filtering and prediction problems". *Transactions of the ASME–Journal of Basic Engineering* **82**:Series D, pp. 35–45.

Khashooei, B. A., D. J. Antunes, and W. P. M. H. Heemels (2017). "Output-based event-triggered control with performance guarantees". *IEEE Transactions on Automatic Control* **62**:7, pp. 3646–3652.

Kingma, D. P. and J. Ba (2014). "Adam: A method for stochastic optimization". arXiv: 1412.6980.

Li, L., M. Lemmon, and X. Wang (2010). "Event-triggered state estimation in vector linear processes". In: *Proceedings of the 2010 American Control Conference*. Baltimore, MD, USA, pp. 2138–2143.

Liu, Q., Z. Wang, X. He, and D. Zhou (2014a). "A survey of event-based strategies on control and estimation". *Systems Science & Control Engineering* **2**:1, pp. 90–97.

Liu, S.-C., T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas (2014b). *Event-Nased Neuromorphic Systems*. John Wiley & Sons.

Meng, X. and T. Chen (2012). "Optimal sampling and performance comparison of periodic and event based impulse control". *IEEE Transactions on Automatic Control* **57**:12, pp. 3252–3259.

Mirkin, L. (2017). "Intermittent redesign of analog controllers via the Youla parameter". *IEEE Transactions on Automatic Control* **62**:4, pp. 1838–1851.

Miskowicz, M. (2006). "Send-on-delta concept: an event-based data reporting strategy". *Sensors* **6**:1, pp. 49–63.

Miskowicz, M. (2015). *Event-Based Control and Signal Processing*. Embedded Systems. CRC Press.

Miskowicz, M. (2007). "Asymptotic effectiveness of the event-based sampling according to the integral criterion". *Sensors* **7**:1, pp. 16–37.

Molin, A. and S. Hirche (2009). "On LQG joint optimal scheduling and control under communication constraints". In: *Proceedings of the 48h IEEE Conference on Decision and Control*. Shanghai, China, pp. 5832–5838.

Molin, A. and S. Hirche (2010a). “Optimal event-triggered control under costly observations”. In: *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems*. Budapest, Hungary.

Molin, A. and S. Hirche (2010b). “Structural characterization of optimal event-based controllers for linear stochastic systems”. In: *Proceedings of the 49th IEEE Conference on Decision and Control*. Atlanta, GA, USA, pp. 3227–3233.

Molin, A., H. Tischer, and S. Hirche (2011). “Order reduction in optimal event-triggered control design for linear stochastic systems”. In: *Proceedings of the 2011 American Control Conference*. San Fransisco, CA, USA, pp. 2222–2227.

Molin, A. (2014). *Optimal Event-triggered Control with Communication Constraints*. PhD thesis. Chair of Information-Oriented Control, Technical University of Munich, Germany.

Murty, K. G. (1972). “On the number of solutions to the complementarity problem and spanning properties of complementary cones”. *Linear Algebra and its Applications* **5**:1, pp. 65–108.

Øksendal, B. (2000). *Stochastic Differential Equations—An Introduction with Applications*. 5th ed. Springer.

Øksendal, B. and K. Reikvam (1998). “Viscosity solutions of optimal stopping problems”. *Stochastics and Stochastic Reports* **62**:3, pp. 285–301.

Otanez, P. G., J. R. Moyne, and D. M. Tilbury (2002). “Using deadbands to reduce communication in networked control systems”. In: *Proceedings of the 2002 American Control Conference*. Anchorage, AK, USA, pp. 3015–3020.

Peskir, G. and A. Shiryaev (2006). *Optimal Stopping and Free-Boundary Problems*. Lectures in Mathematics, ETH Zürich. Birkhäuser.

Potter, J. E. (1964). “A guidance-navigation separation theorem”. In: *Proceedings of the 1964 Astrodynamics Guidance and Control Conference*. Los Angeles, CA, USA, pp. 1–24.

Rault, T., A. Bouabdallah, and Y. Challal (2014). “Energy efficiency in wireless sensor networks: A top-down survey”. *Computer Networks* **67**, pp. 104–122.

Robin, M. (1983). “Long-term average cost control problems for continuous time markov processes: A survey”. *Acta Applicandae Mathematica* **1**:3, pp. 281–299.

Rump, S. M. (2003). “On P-matrices”. *Linear Algebra and its Applications* **363**, pp. 237–250.

Ruuskanen, J. and A. Cervin (2019). “Event-based state estimation using the auxiliary particle filter”. In: *Proceedings of the 2019 European Control Conference*. Naples, Italy, pp. 1854–1860.

Ruuskanen, J. and A. Cervin (2020). “On innovation-based triggering for event-based nonlinear state estimation using the particle filter”. In: *Proceedings of the 2020 European Control Conference*. Saint Petersburg, Russia, pp. 1401–1408.

Sánchez, J., A. Visioli, and S. Dormido (2011). “A two-degree-of-freedom PI controller based on events”. *Journal of Process Control* **21**:4, pp. 639–651.

Schmitt, E. J., B. Noack, W. Krippner, and U. D. Hanebeck (2019). “Gaussianity-preserving event-based state estimation with an FIR-based stochastic trigger”. *IEEE Control System Letters* **3**:3, pp. 769–774.

Seyboth, G. S., D. V. Dimarogonas, and K. H. Johansson (2013). “Event-based broadcasting for multi-agent average consensus”. *Automatica* **49**:1, pp. 245–252.

Shi, D., L. Shi, and T. Chen (2016). *Event-Based State Estimation : A Stochastic Perspective*. Vol. 41. Studies in Systems, Decision and Control. Springer.

Sid, M. A. and S. Chitraganti (2016). “Nonlinear event-based state estimation using particle filtering approach”. In: *Proceedings of the 8th International Conference on Modelling, Identification and Control*. Algiers, Algeria, pp. 874–879.

Sijs, J. and M. Lazar (2012). “Event based state estimation with time synchronous updates”. *IEEE Transactions on Automatic Control* **57**:10, pp. 2650–2655.

Solowjow, F., D. Baumann, J. Garcke, and S. Trimpe (2018). “Event-triggered learning for resource-efficient networked control”. In: *Proceedings of the 2018 American Control Conference*. Milwuakee, WI, USA, pp. 6506–6512.

Song, J., A. K. Mok, D. Chen, M. S. Nixon, T. Blevins, and W. Wojsznis (2006). “Improving PID control with unreliable communications”. In: *Proceedings of the ISA EXPO Technical Conference*. Houston, TX, USA.

Stellato, B., G. Banjac, P. Goulart, A. Bemporad, and S. Boyd (2020). “OSQP: An operator splitting solver for quadratic programs”. *Mathematical Programming Computation*. URL: https://doi.org/10.1007/s12532-020-00179-2.

Thelander Andrén, M. (2020). “Remote state estimation using stochastic triggering with buffered measurements”. Manuscript in preparation.

Thelander Andrén, M., B. Bernhardsson, A. Cervin, and K. Soltesz (2017). "On event-based sampling for LQG-optimal control". In: *Proceedings of the 56th IEEE Conference on Decision and Control*. Melbourne, Australia, pp. 5438–5444.

Thelander Andrén, M. and A. Cervin (2016). "Event-based state-estimation using an improved stochastic send–on–delta scheme". In: *Proceedings of the 2nd International Conference on Event-based Control, Communication and Signal Processing*. Kraków, Poland, pp. 1–8.

Thelander Andrén, M. and A. Cervin (2020). "Optimal event-based sampling in LQG control: Exact solutions and numerical methods". *IEEE Transactions on Automatic Control*. Submitted.

Tiberi, U., J. Araújo, and K. H. Johansson (2012). "On event-based PI control of first-order processes". *IFAC Proceedings Volumes* **45**:3, pp. 448–453.

Trimpe, S. and R. D'Andrea (2014). "Event-based state estimation with variance-based triggering". *IEEE Transactions on Automatic Control* **59**:12, pp. 3266–3281.

Tsatsomeros, M. J. (2002). "Generating and detecting matrices with positive principal minors". *Asian Information-Science-Life* **1**:2, pp. 115–132.

Van Loan, C. F. (1978). "Computing integrals involving the matrix exponential". *IEEE Transactions on Automatic Control* **23**:3, pp. 395–404.

Vasyutynskyy, V. and K. Kabitzsch (2009). "First order observers in event-based PID controls". In: *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation*. Mallorca, Spain, pp. 1–8.

Vasyutynskyy, V. and K. Kabitzsch (2006). "Implementation of PID controller with send-on-delta sampling". In: *Proceedings of the UKACC International Control Conference*. Glasgow, UK, pp. 4423–4428.

Wilmott, P., S. Howison, and J. Dewynne (1995). *The Mathematics of Financial Derivatives*. Cambridge University Press.

Yamamoto, Y. (1994). "A function space approach to sampled data control systems and tracking problems". *IEEE Transactions on Automatic Control* **39**:4, pp. 703–713.

Zhang, D., P. Shi, Q.-G. Wang, and L. Yu (2017). "Analysis and synthesis of networked control systems: a survey of recent advances and challenges". *ISA Transactions* **66**, pp. 376–392.

Zhou, K., J. C. Doyle, and K. Glover (1996). *Robust and Optimal Control*. Prentice-Hall.

A

Discrete-Time Model for the Baseline LQG Design

This appendix presents a derivation of the equivalent discrete-time generalized plant (2.12) for the baseline LQG design Section 2.4, including the integrate–and–reset (IaR) pre-filter. The involved matrices in the discrete-time formulation are expressed in terms of the given parameters in the continuous-time model (2.1).

Consider the extended plant, with state vector $\bar{x} \coloneqq \begin{bmatrix} x^\intercal & y_f^\intercal \end{bmatrix}^\intercal$, where y_f is the output of the IaR pre-filter

$$y_f(t) \coloneqq \frac{1}{h}\int_{t-h}^{t} y(v)dv = \frac{1}{h}\int_{t-h}^{t} \big(C_y x(v) + D_{yw} w(v)\big)dv.$$

First, from (2.11) we see immediately that

$$C_{\bar{y}} = \begin{bmatrix} 0 & I \end{bmatrix}.$$

Next, we integrate the differential equation for x in (2.1) and note that u is constant over each sampling interval due to the assumption of zero-order hold. This yields [Åström, 1970, Chapter 3]

$$\forall t \in [\tau_k,\ \tau_{k+1}] : \begin{cases} \bar{x}(t) = \Phi(t-\tau_k)\bar{x}_k + \bar{w}_{\bar{x}}(t-\tau_k) + \Gamma(t-\tau_k)\bar{u}_k, \\ \bar{w}_{\bar{x}}(t-\tau_k) \sim \mathcal{N}\big(0,\, R_{\bar{x}}(t-\tau_k)\big), \end{cases} \tag{A.1}$$

with

$$\begin{aligned} \Phi(t) &= e^{t\bar{A}} - \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}, \\ \Gamma(t) &= \int_0^t e^{v\bar{A}} dv \begin{bmatrix} B_u \\ 0 \end{bmatrix}, \\ R_{\bar{x}}(t) &= \int_0^t e^{v\bar{A}} \begin{bmatrix} B_w & \frac{1}{h} D_{yw} \end{bmatrix} \begin{bmatrix} B_w & \frac{1}{h} D_{yw} \end{bmatrix}^\intercal e^{v\bar{A}^\intercal} dv, \end{aligned}$$

where

$$\bar{A} \coloneqq \begin{bmatrix} A & 0 \\ \frac{1}{h}C_y & 0 \end{bmatrix} \quad \Longrightarrow \quad e^{t\bar{A}} = \begin{bmatrix} e^{tA} & 0 \\ \frac{1}{h}C_y \int_0^t e^{vA} dv & I \end{bmatrix}. \tag{A.2}$$

The expression for the exponential matrix in (A.2) is due to Van Loan's formula [Van Loan, 1978, Theorem 1]. By considering $t = \tau_{k+1} = \tau_k + h$ in (A.1) we retrieve the difference equation for $\bar{x}$ in (2.12), with

$$A_{\bar{x}} = \Phi(h), \quad B_{\bar{u}} = \Gamma(h), \quad B_{\bar{w}} B_{\bar{w}}^{\mathsf{T}} = R_{\bar{x}}(h).$$

Finally, we consider the equation for the regulated output $\bar{z}$ in (2.12). First, split the cost J into a sum of integrals of length h

$$\begin{aligned} J &= \lim_{T\to\infty} \mathbb{E}\Big[\frac{1}{T}\int_0^T \|z(t)\|^2 dt\Big] \\ &= \lim_{N\to\infty} \mathbb{E}\Big[\frac{1}{Nh}\sum_{k=0}^{N-1}\int_{\tau_k}^{\tau_k+h} \|z(t)\|^2 dt\Big] \\ &= \lim_{N\to\infty} \frac{1}{Nh}\sum_{k=0}^{N-1}\int_{\tau_k}^{\tau_k+h} \mathbb{E}\big[\|z(t)\|^2\big] dt, \end{aligned} \tag{A.3}$$

where Tonelli's theorem was used in the third equality. Each integral in (A.3) can be written as

$$\int_{\tau_k}^{\tau_k+h} \mathbb{E}\big[\|z(t)\|^2\big] dt = \int_{\tau_k}^{\tau_k+h} \mathbb{E}\left[\left\| \begin{bmatrix} \bar{C}_z^{\mathsf{T}} \\ D_{zu}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \bar{x}(t) \\ u(t) \end{bmatrix} \right\|^2\right] dt, \tag{A.4}$$

where

$$\bar{C}_z \coloneqq \begin{bmatrix} C_z & 0 \end{bmatrix}.$$

Note that since the integral (A.4) covers the interval $t \in [\tau_k, \tau_k + h]$ we have $u(t) = \bar{u}_k$ and $\bar{x}(t)$ is given by (A.1). Since $\bar{w}_{\bar{x}}$ has zero mean and is uncorrelated with x_k and $\bar{u}_k$ for all $t > \tau_k$, the only stochastic term in (A.4) with non-zero expected value is the quadratic term in $\bar{w}_{\bar{x}}$. The expected value in (A.4) thus evaluates to

$$\begin{aligned} \mathbb{E}\big[\|z(t)\|^2\big] = \mathbb{E}\big[\|\bar{C}_z \bar{w}_{\bar{x}}(t-\tau_k)\|^2\big] \\ + \left\| \begin{bmatrix} \bar{C}_z^{\mathsf{T}} \\ D_{zu}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \Phi(t-\tau_k) & \Gamma(t-\tau_k) \\ 0 & I \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} \right\|^2. \end{aligned} \tag{A.5}$$

The first term on the right-hand side of (A.5) can be computed as [Åström and Wittenmark, 1997, (11.28)]

$$\mathbb{E}\big[\|\bar{C}_z \bar{w}_{\bar{x}}(t-\tau_k)\|^2\big] = \operatorname{tr}\big(\bar{C}_z^{\mathsf{T}} \bar{C}_z R_{\bar{x}}(t-\tau_k)\big).$$

The integral (A.4) is thus given by

$$\int_{\tau_k}^{\tau_k+h} \mathbb{E}\big[\|z(t)\|^2\big]dt = \operatorname{tr}\big(\bar{C}_z^\intercal \bar{C}_z \int_0^h R_{\bar{x}}(t)dt\big) + \begin{bmatrix}\bar{x}_k\\ \bar{u}_k\end{bmatrix}^\intercal \int_0^h \begin{bmatrix}\Phi(t) & \Gamma(t)\\ 0 & I\end{bmatrix}^\intercal \begin{bmatrix}\bar{C}_z^\intercal\\ D_{zu}^\intercal\end{bmatrix} \begin{bmatrix}\bar{C}_z^\intercal\\ D_{zu}^\intercal\end{bmatrix}^\intercal \begin{bmatrix}\Phi(t) & \Gamma(t)\\ 0 & I\end{bmatrix} dt \begin{bmatrix}\bar{x}_k\\ \bar{u}_k\end{bmatrix}.$$

The cost can therefore be equivalently written as

$$J = \gamma_w + \limsup_{N\to\infty} \mathbb{E}\left[\frac{1}{Nh}\sum_{k=0}^{N-1}\|\bar{z}_k\|^2\right],$$

with

$$\gamma_w \coloneqq \frac{1}{h}\operatorname{tr}\big(\bar{C}_z^\intercal \bar{C}_z \int_0^h R_{\bar{x}}(t)dt\big) \tag{A.6}$$

and

$$\bar{z}_k \coloneqq C_{\bar{z}} x_k + D_{\bar{z}\bar{u}}\bar{u}_k,$$

where matrices $C_{\bar{z}}$ and $D_{\bar{z}\bar{u}}$ are given by the relation

$$\begin{bmatrix}C_{\bar{z}}^\intercal\\ D_{\bar{z}\bar{u}}^\intercal\end{bmatrix}\begin{bmatrix}C_{\bar{z}}^\intercal\\ D_{\bar{z}\bar{u}}^\intercal\end{bmatrix}^\intercal = \int_0^h \begin{bmatrix}\Phi(t) & \Gamma(t)\\ 0 & I\end{bmatrix}^\intercal \begin{bmatrix}\bar{C}_z^\intercal\\ D_{zu}^\intercal\end{bmatrix}\begin{bmatrix}\bar{C}_z^\intercal\\ D_{zu}^\intercal\end{bmatrix}^\intercal \begin{bmatrix}\Phi(t) & \Gamma(t)\\ 0 & I\end{bmatrix} dt. \tag{A.7}$$

B

From PID Parameters to Benchmark LQG Problem

This appendix present the (non-unique) inverse relation between a given ideal PID controller on parallel form (cf. (5.11)) and the benchmark LQG design problem described in Section 5.1. That is, given a set of (positive) PID parameters K, T_i and T_d, we would like to find corresponding values of q_y, $q_{y\eta}$, r_d and k for the benchmark problem.

We recall from Section 5.1 that the parameters for the benchmark problem satisfy k, r_d, $q_y > 0$ and $q_y \geq q_{y\eta}^2$. In turn, these parameters are related to the optimal state feedback and estimator gains f_1, f_2, and l_d as

$$\begin{aligned} f_1 &= -\sqrt{2\big(\sqrt{q_y} - q_{y\eta}\big)}, \\ f_2 &= -\sqrt{q_y}, \\ l_d &= -\sqrt{r_d}. \end{aligned} \tag{B.1}$$

From these relations, we see that permissible values of parameters for the benchmark problem can only be found iff f_1, f_2, $l_d < 0$. Since the feedback and estimator gains are related to the PID parameters according to

$$\begin{aligned} K &= \frac{l_d f_1 - k f_2}{k}, \\ T_i &= \frac{l_d f_1 - k f_2}{l_d k f_2}, \\ T_d &= \frac{-f_1 k - l_d}{l_d k f_1 - k^2 f_2}, \end{aligned} \tag{B.2}$$

we see that this in turn is equivalent to K, T_i, $T_d > 0$, i.e., the PID parameters must be positive for an inverse relation to exist.

The next question is if (B.2) always can be solved for f_1, f_2, l_d with a given set of positive PID parameters. By first isolating l_d from (B.2), we

obtain a third-order equation

$$l_d^3 + l_d^2 k^2 K T_d + k^2 K l_d + \frac{k^2 K}{T_i} = 0.$$

By Descartes' rule of signs, this equation has either one or three negative real solutions, so we can always find a valid $l_d < 0$. Secondly, with a solution $l_d < 0$, we then obtain

$$f_2 = \frac{K}{T_i l_d} < 0,$$

which is also valid. Thirdly, from the relation

$$K T_d = \frac{-l_d - f_1 k}{k^2},$$

it follows that

$$f_1 = K T_d k - \frac{l_d}{k}. \tag{B.3}$$

This means that $f_1 < 0$ iff k is chosen such that

$$k > \sqrt{\frac{-l_d}{K T_d}}. \tag{B.4}$$

Since any k satisfying (B.4) is valid, the resulting benchmark LQG design problem is not unique. After having chosen a large enough value for k, we obtain f_1 via (B.3). Finally, values for q_y, $q_{y\eta}$ and r_d are obtained via (B.1).

C

Proofs

This appendix presents proofs for selected thesis results.

Proof of Theorem 3.5. First, note that assumptions **A1**, **A2** and $A = 0$ implies $R, Q \succ 0$. Now, recall the Riccati-like equation (3.29)

$$PRP + \frac{1}{2}\mathrm{tr}(RP)P = Q. \tag{C.1}$$

Multiplying (C.1) by $Q^{-\frac{1}{2}}$ from both sides and defining $S \coloneqq Q^{\frac{1}{2}} R Q^{\frac{1}{2}}$ yields

$$Q^{-\frac{1}{2}} P Q^{-\frac{1}{2}} S Q^{-\frac{1}{2}} P Q^{-\frac{1}{2}} + \frac{1}{2}\mathrm{tr}(SQ^{-\frac{1}{2}} P Q^{-\frac{1}{2}}) Q^{-\frac{1}{2}} P Q^{-\frac{1}{2}} = I. \tag{C.2}$$

With the eigendecomposition $S = U_S \Lambda_S U_S^\intercal$ and definition $\Lambda_P \coloneqq U_S^\intercal Q^{-\frac{1}{2}} P Q^{-\frac{1}{2}} U_S$ we can re-write (C.2) as

$$\Lambda_P \Lambda_S \Lambda_P + \frac{1}{2}\mathrm{tr}(\Lambda_S \Lambda_P)\Lambda_P = I. \tag{C.3}$$

With the ansatz that Λ_P is a diagonal matrix we can also express (C.3) as

$$\Lambda_{P,ii}^2 \Lambda_{S,ii} + \frac{1}{2}\sum_{j=1}^{n}(\Lambda_{S,jj}\Lambda_{P,jj})\Lambda_{P,ii} = 1, \quad i = 1, \ldots, n. \tag{C.4}$$

Note that if there exists a solution to (C.4), then it is the unique solution to (C.3) and ultimately (C.1). Define $\alpha \coloneqq \sum_{j=1}^{n}(\Lambda_{S,jj}\Lambda_{P,jj}) > 0$ and insert it into (C.4). We can then re-write (C.4) as a second-order equation in $\Lambda_{P,ii}$ as

$$\Lambda_{P,ii}^2 + \frac{\alpha}{2\Lambda_{S,ii}}\Lambda_{P,ii} - \frac{1}{\Lambda_{S,ii}} = 0, \quad i = 1, \ldots, n. \tag{C.5}$$

Note that $\Lambda_{P,ii}, \Lambda_{S,ii} > 0$ since $R, Q \succ 0$. Solving (C.5) for $\Lambda_{P,ii}$ yields

$$\Lambda_{P,ii} = -\frac{\alpha}{4\Lambda_{S,ii}} + \sqrt{\frac{\alpha^2}{16\Lambda_{S,ii}^2} + \frac{1}{\Lambda_{S,ii}}}, \quad i = 1, \ldots, n, \tag{C.6}$$

and thus

$$4\Lambda_{S,ii}\Lambda_{P,ii} = -\alpha + \sqrt{\alpha^2 + 16\Lambda_{S,ii}}, \quad i = 1, \ldots, n. \tag{C.7}$$

Summation over $i = 1, \ldots, n$ in (C.7) and some re-arranging yields

$$(n+4)\alpha - \sum_{i=1}^{n} \sqrt{\alpha^2 + 16\Lambda_{S,ii}} = 0, \quad \alpha > 0. \tag{C.8}$$

Solving (C.4) is thus equivalent to finding a root of a scalar function $l : \alpha \mapsto \mathbb{R}$, defined as the left hand side of (C.8). Since l is continuous and

$$\lim_{\alpha \to 0^+} l(\alpha) = -\sum_{i=1}^{n} 4\sqrt{\Lambda_{S,ii}} < 0,$$
$$\lim_{\alpha \to \infty} l(\alpha) = \infty > 0,$$

we conclude that there exists a solution to (C.8). Furthermore, the solution is unique since l is strictly increasing, with

$$\frac{dl}{d\alpha} > 4, \quad \forall \alpha > 0.$$

With a solution to (C.8), we then obtain Λ_P via (C.6), which gives the final solution

$$P = Q^{\frac{1}{2}} U_S \Lambda_P U_S^{\mathsf{T}} Q^{\frac{1}{2}}.$$

This concludes the proof. □

Proof of Theorem 3.6. First, note that assumptions **A1**, **A2** and $A = 0$ implies $R, Q \succ 0$. Now, from (3.32) and (3.33) we have

$$J_{\text{ratio}} =$$
$$2\frac{\operatorname{tr}(RQ)}{\big(\operatorname{tr}(RP)\big)^2} = 2\frac{\operatorname{tr}\big((RP)^2 + \frac{1}{2}\operatorname{tr}(RP)RP\big)}{\big(\operatorname{tr}(RP)\big)^2} = 1 + 2\frac{\operatorname{tr}\big((RP)^2\big)}{\big(\operatorname{tr}(RP)\big)^2}, \tag{C.9}$$

where (3.29) was used in the second equality. Now, let $\lambda \coloneqq \begin{bmatrix}\lambda_1 & \ldots & \lambda_n\end{bmatrix}^{\mathsf{T}}$ denote the vector of eigenvalues to the matrix RP. Since $R, P \succ 0$ we have $\lambda_i > 0$ for all $i = 1, \ldots, n$. Therefore;

$$\operatorname{tr}(RP) = \sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} |\lambda_i| = \|\lambda\|_1. \tag{C.10}$$

Furthermore, we have

$$\operatorname{tr}\big((RP)^2\big) = \sum_{i=1}^{n} \lambda_i^2 = \|\lambda\|^2. \tag{C.11}$$

Inserting (C.10) and (C.11) into (C.9) yields

$$J_{\text{ratio}} = 1 + 2\frac{\|\lambda\|^2}{\|\lambda\|_1^2}.$$

If $n = 1$ we have $\|\lambda\|^2 = \|\lambda\|_1^2$ and thus $J_{\text{ratio}} = 3$. If $n \geq 2$, the Cauchy–Schwarz inequality gives

$$\frac{1}{n} \leq \frac{\|\lambda\|^2}{\|\lambda\|_1^2} < 1. \tag{C.12}$$

From (C.12) it follows that

$$1 + \frac{2}{n} \leq J_{\text{ratio}} < 3, \quad n \geq 2. \tag{C.13}$$

When RP only have repeated eigenvalues, i.e., $\lambda_i \coloneqq \lambda_{\text{rp}}$ for all $i = 1, \ldots, n$, the lower bound in (C.13) is attained as

$$J_{\text{ratio}} = 1 + 2\frac{\|\lambda\|^2}{\|\lambda\|_1^2} = 1 + 2\frac{\lambda_{\text{rp}}^2 n}{(\lambda_{\text{rp}} n)^2} = 1 + \frac{2}{n}.$$

The upper bound in (C.13) is the limit of J_{ratio} when all but one eigenvalue of RP approach zero, as

$$J_{\text{ratio}} = 1 + 2\frac{\|\lambda\|^2}{\|\lambda\|_1^2} \to 1 + 2\frac{\lambda_i^2}{\lambda_i^2} = 3, \quad \text{as } \lambda_j \to 0,\ \forall j \neq i.$$

To show that the same conditions hold for the eigenvalues of RQ, we first multiply (3.29) with R from the left, which yields

$$(RP)^2 + \text{tr}(RP)RP = RQ. \tag{C.14}$$

Define the Jordan decomposition of RP as

$$RP \coloneqq S\mathcal{J}S^{-1},$$

where S is an invertible matrix and $\mathcal{J}$ is the Jordan canonical form of RP, which is a triangular matrix with λ along its diagonal. Inserting the Jordan decomposition into (C.14) yields

$$S\big(\mathcal{J}^2 + \frac{1}{2}\text{tr}(\mathcal{J})\mathcal{J}\big)S^{-1} = RQ,$$

which implies that RQ is similar to $\mathcal{J}^2 + \frac{1}{2}\text{tr}(\mathcal{J})\mathcal{J}$, and thus share the same eigenvalues. These eigenvalues are given by

$$\lambda_i^2 + 1/2\sum_{j=1}^{n} \lambda_j\lambda_i, \quad i = 1, \ldots, n. \tag{C.15}$$

Since $\lambda_i > 0$ for all $i = 1, \dots, n$, the eigenvalues (C.15) of RQ are all repeated iff the eigenvalues λ_i of RP are all repeated. Furthermore, the i^{th} eigenvalue of RQ tends to zero iff the eigenvalue λ_i of RP tends to zero. Thus the conditions on RQ are equivalent to those for RP, which concludes the proof. □

Proof of Theorem 3.7. Define $d \coloneqq \|\tilde{x}\|_Q$. Using the chain rule we have

$$\nabla_{\tilde{x}} V = \frac{1}{d}\frac{\partial V}{\partial d} Q\tilde{x},$$
$$\nabla^2_{\tilde{x}} V = \frac{1}{d}\frac{\partial V}{\partial d} Q + \frac{1}{d^2}\Big(\frac{\partial^2 V}{\partial d} - \frac{1}{d}\frac{\partial V}{\partial d}\Big) Q\tilde{x}\tilde{x}^\intercal Q.$$

From the assumption $RQ = \alpha I$ we then have

$$\frac{1}{2}\mathrm{tr}(R\nabla^2 V_{\tilde{x}}) =$$
$$\frac{\alpha}{2}\mathrm{tr}\Big(\frac{1}{d}\frac{\partial V}{\partial d} I_n + \frac{1}{d^2}\Big(\frac{\partial^2 V}{\partial^2 d} - \frac{1}{d}\frac{\partial V}{\partial d}\Big)\tilde{x}\tilde{x}^\intercal Q\Big) = \frac{\alpha}{2}\Big(\frac{\partial^2 V}{\partial^2 d} + \frac{n-1}{d}\frac{\partial V}{\partial d}\Big).$$

Since $A = aI + B$ and BR is skew-symmetric, we have

$$\tilde{x}^\intercal A^\intercal \nabla_{\tilde{x}} V = \frac{1}{d}\frac{\partial V}{\partial d}\tilde{x}A^\intercal Q\tilde{x} = \frac{1}{d}\frac{\partial V}{\partial d}\Big(a\tilde{x}^\intercal Q\tilde{x} + \tilde{x}^\intercal B^\intercal Q\tilde{x}\Big)$$
$$= \frac{1}{d}\frac{\partial V}{\partial d}\Big(ad^2 + \alpha\tilde{x}^\intercal B^\intercal R^{-1}\tilde{x}\Big) = \frac{\partial V}{\partial d} ad, \quad \text{(C.16)}$$

where in the third equality we used $RQ = \alpha I$, and in the fourth equality we used the relation

$$BR + RB^\intercal = 0 \iff R^{-1}B + B^\intercal R^{-1} = 0.$$

For spiral systems we thus have

$$\mathcal{A}V(\tilde{x}) + \|\tilde{x}\|_Q^2 = \frac{\alpha}{2}\frac{\partial^2 V(d)}{\partial^2 d} + \Big(ad + \frac{\alpha(n-1)}{2d}\Big)\frac{\partial V(d)}{\partial d} + d^2,$$

which means that (3.10) can be reduced to one dimension, and by Theorem 3.1 an optimal sampling sequence will be generated by a Markovian threshold policy which triggers when

$$V(d) = \rho + V(0), \quad \text{(C.17)}$$

holds. Since the level set defined by (C.17) corresponds to some fixed value $d = \Delta_\rho$, an optimal trigger function is given by

$$g(\tilde{x}) = \Delta_\rho - \|\tilde{x}\|_Q.$$

With the problem reduced to one dimension, it is straightforward to derive the expressions for J_u, $\bar{h}$ and V similarly to how the corresponding expressions in Theorem 3.3 were derived in [Henningsson et al., 2008], to which we refer for details. This concludes the proof. □

Proof of Theorem 3.8. We begin by proving the result for the single-input case. In this case the weight matrix Q is of rank 1, and thus has an eigendecomposition of the form

$$Q = \underbrace{\begin{bmatrix} U_\parallel & U_\perp \end{bmatrix}}_{:=U} \begin{bmatrix} \lambda_Q & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_\parallel^\mathsf{T} \\ U_\perp^\mathsf{T} \end{bmatrix}, \quad UU^\mathsf{T} = I,$$

where $\lambda_Q > 0$ is the single non-zero eigenvalue of Q, $U_\parallel \in \mathbb{R}^{n\times 1}$, $U_\perp \in \mathbb{R}^{n\times(n-1)}$ and $U_\parallel^\mathsf{T} U_\perp = 0$. Specifically, we have

$$U_\parallel = \frac{F^\mathsf{T}}{\|F\|}, \quad \lambda_Q = \|Q\|.$$

With the change of variables $\tilde{x}_{\text{new}} := U^\mathsf{T}\tilde{x}$ we get

$$\begin{aligned} \dot{\tilde{x}}_{\text{new}} &= \begin{bmatrix} \dot{\tilde{x}}_{\text{new},\parallel} \\ \dot{\tilde{x}}_{\text{new},\perp} \end{bmatrix} \\ &= U^\mathsf{T} A U \tilde{x}_{\text{new}} + U^\mathsf{T} v = \begin{bmatrix} \frac{FAF^\mathsf{T}}{\|F\|^2} & \frac{FAU_\perp}{\|F\|} \\ \frac{U_\perp^\mathsf{T} AF^\mathsf{T}}{\|F\|} & U_\perp^\mathsf{T} A U_\perp \end{bmatrix} \begin{bmatrix} \tilde{x}_{\text{new},\parallel} \\ \tilde{x}_{\text{new},\perp} \end{bmatrix} + U^\mathsf{T} v, \end{aligned} \tag{C.18}$$

where $\tilde{x}_{\text{new},\parallel} \in \mathbb{R}$ and $\tilde{x}_{\text{new},\perp} \in \mathbb{R}^{n-1}$. The LQG cost (3.1) only depends on the scalar $\tilde{x}_{\text{new},\parallel}$ since

$$\begin{aligned} \|\tilde{x}\|_Q^2 &= \tilde{x}^\mathsf{T} Q \tilde{x} \\ &= \tilde{x}_{\text{new}}^\mathsf{T} U^\mathsf{T} Q U \tilde{x}_{\text{new}} = \begin{bmatrix} \tilde{x}_{\text{new},\parallel} \\ \tilde{x}_{\text{new},\perp} \end{bmatrix}^\mathsf{T} \begin{bmatrix} \lambda_Q & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_{\text{new},\parallel} \\ \tilde{x}_{\text{new},\perp} \end{bmatrix} = ||Q|| \tilde{x}_{\text{new},\parallel}^2. \end{aligned} \tag{C.19}$$

Furthermore, the assumption $FAQ = FQA$ gives

$$\frac{FAU_\perp}{||F||} = \frac{FF^\mathsf{T} FAU_\perp}{||F||^3} = \frac{FQAU_\perp}{||F|| \cdot ||Q||} = \frac{FAQU_\perp}{||F|| \cdot ||Q||} = \frac{FAF^\mathsf{T} U_\parallel U_\perp}{||F|| \cdot ||Q||} = 0.$$

The same assumption also implies $U_\perp^\mathsf{T} AF^\mathsf{T}/\|F\| = 0$. The states $\tilde{x}_{\text{new},\parallel}$ and $\tilde{x}_{\text{new},\perp}$ are thus discoupled, and the OSP is reduced to a first-order problem in $\tilde{x}_{\text{new},\parallel} = F\tilde{x}/\|F\|$. The solution of the reduced-order problem is given by Theorem 3.3, with parameters retrieved from (C.18) and (C.19) as

$$a = \frac{FAF^\mathsf{T}}{\|F\|^2}, \quad r = \frac{FRF^\mathsf{T}}{\|F\|^2}, \quad q = ||Q||.$$

The single-output case is handled analogously to the single-input case using a linear transformation based on the eigendecomposition of R. Observing that the transformed innovations process is one-dimensional and that the assumption $L^\intercal AR = L^\intercal RA$ gives a discoupled transformed system then gives the final result. This concludes the proof. □

Proof of Theorem 4.2. First, we require the following definition:

Definition C.1—Row Diagonally Dominant
A matrix $B \in \mathbb{R}^{N\times N}$ is said to be *row diagonally dominant* (RDD) if

$$|B_{ii}| > \sum_{j\neq i} |B_{ij}|, \quad \forall i = 1, ..., N. \qquad \diamondsuit$$

With this definition we proceed with the proof by using the following lemma [Tsatsomeros, 2002, Proposition 4.6]:

Lemma C.1
The matrix $M \coloneqq -\Psi\Phi^{-1}$ is a P-matrix if $-\Psi$ and Φ are RDD matrices with positive diagonal elements.

See [Tsatsomeros, 2002, Proposition 4.6] for proof of this lemma. The statement of Theorem 4.2 is thus proven if we show that $-\Psi$ and Φ are both RDD with positive diagonal entries for all $c > \underline{c}$ for some finite $\underline{c} > 0$.

Consider first Φ, which has positive diagonal entries since

$$\Phi_{ii} = \phi_i(\tilde{x}_i) = \exp\big(-c\|\tilde{x}_i - \tilde{x}_i\|^2\big) = 1 > 0, \quad i = 1, \ldots, N.$$

For Φ to be RDD we require that

$$|\Phi_{ii}| = 1 > \sum_{j\neq i} |\Phi_{ij}| = \sum_{j\neq i} \exp\big(-c\|\tilde{x}_i - \tilde{x}_j\|^2\big), \quad \forall i = 1, \ldots, N. \qquad \text{(C.20)}$$

The assumption of unique collocation points implies $\|\tilde{x}_i - \tilde{x}_j\| \neq 0$. The sum in (C.20) is therefore guaranteed to be a strictly decreasing function in $c \in (0, \infty)$. Furthermore, we have

$$i = 1, \ldots, N : \begin{cases} \lim\limits_{c\to 0^+} \sum_{j\neq i} \exp\big(-c\|\tilde{x}_i - \tilde{x}_j\|^2\big) = N - 1, \\ \lim\limits_{c\to\infty} \sum_{j\neq i} \exp\big(-c\|\tilde{x}_i - \tilde{x}_j\|^2\big) = 0. \end{cases}$$

Therefore, the inequality (C.20) is always satisfied if $N \leq 2$, whereas if $N > 2$ there exist a unique set of values $\{\underline{c}_{\Phi,i} > 0\}$ such that

$$1 = \sum_{j\neq i} \exp\big(-\underline{c}_{\Phi,i}\|\tilde{x}_i - \tilde{x}_j\|^2\big), \quad \forall i = 1, \ldots, N.$$

The matrix Φ is thus RDD if we pick a c satisfying

$$c > \underline{c}_\Phi \coloneqq \max_i \underline{c}_{\Phi,i}.$$

Now consider the matrix $-\Psi$, which also has positive diagonal elements as

$$\begin{aligned}(-\Psi)_{ii} &= -\Lambda_i(\tilde{x}_i)\phi_i(\tilde{x}_i) = -\Lambda_i(\tilde{x}_i) \\ &= -c\big(2c\|\tilde{x}_i - \tilde{x}_i\|_R^2 - 2\tilde{x}_i^\intercal A^\intercal(\tilde{x}_i - \tilde{x}_i) - \mathrm{tr}(R)\big) = c\,\mathrm{tr}(R) > 0. \quad \text{(C.21)}\end{aligned}$$

The inequality in (C.21) holds since $R \succeq 0$ and $R \neq 0$. For $-\Psi$ to be RDD we require that

$$\begin{aligned}\mathrm{tr}(R) &> \frac{1}{c}\sum_{j\neq i}|(-\Psi)_{ij}| \\ &= \sum_{j\neq i}|\mathrm{tr}(R) - 2c\|\tilde{x}_i - \tilde{x}_j\|_R^2 + 2\tilde{x}_i^\intercal A^\intercal(\tilde{x}_i - \tilde{x}_j)|\phi_j(\tilde{x}_i), \quad \forall i = 1, ..., N.\end{aligned} \quad \text{(C.22)}$$

Using the triangle inequality, we note that the sum in (C.22) is bounded from above by a function $b_i : \mathbb{R} \mapsto \mathbb{R}$ given as

$$b_i(c) \coloneqq \sum_{j\neq i}\big(\mathrm{tr}(R) + 2c\|\tilde{x}_i - \tilde{x}_j\|_R^2 + 2|\tilde{x}_i^\intercal A^\intercal(\tilde{x}_i - \tilde{x}_j)|\big)\phi_j(\tilde{x}_i), \quad i = 1, \ldots, N. \quad \text{(C.23)}$$

Thus, the inequality

$$\mathrm{tr}(R) > b_i(c), \quad \forall i = 1, \ldots, N, \quad \text{(C.24)}$$

implies that (C.22) is satisfied and that $-\Psi$ is RDD. Each function $b_i(c)$ is continuous, with limits

$$i = 1, \ldots, N : \begin{cases} \lim\limits_{c\to 0+} b_i(c) = \sum_{j\neq i}(\mathrm{tr}(R) + 2|\tilde{x}_i^\intercal A^\intercal(\tilde{x}_i - \tilde{x}_j)|) > \mathrm{tr}(R), \\ \lim\limits_{c\to\infty} b_i(c) = 0. \end{cases} \quad \text{(C.25)}$$

Furthermore, the derivative of $b_i(c)$ is given by

$$\begin{aligned}&\frac{\partial b_i}{\partial c} = \\ &\sum_{j\neq i}\big(2\|\tilde{x}_i - \tilde{x}_j\|_R^2 - \|\tilde{x}_i - \tilde{x}_j\|^2(\mathrm{tr}(R) + 2c\|\tilde{x}_i - \tilde{x}_j\|_R^2 + 2|\tilde{x}_i^\intercal A^\intercal(\tilde{x}_i - \tilde{x}_j)|)\big)\phi_j(\tilde{x}_i).\end{aligned} \quad \text{(C.26)}$$

Since the collocation points are unique, each function $b_i(c)$ has a unique stationary point given by

$$c_i^\star = \frac{\sum_{j\neq i}\left(2\|\tilde{x}_i - \tilde{x}_j\|_R^2 - \|\tilde{x}_i - \tilde{x}_j\|^2(\mathrm{tr}(R) + 2|\tilde{x}_i^\mathsf{T} A^\intercal(\tilde{x}_i - \tilde{x}_j)|)\right)}{2\sum_{j\neq i}\|\tilde{x}_i - \tilde{x}_j\|_R^2\|\tilde{x}_i - \tilde{x}_j\|^2}.$$

By studying the sign of (C.26) around $c_i^\star$ we deduce that the stationary point for each function $b_i(c)$ is a local maxima, and each function is strictly decreasing on the interval $c \in (c_i^\star, \infty)$. Thus, there exists a $\underline{c}_{\Psi,i} \geq c_i^\star$ for each row i such that

$$\mathrm{tr}(R) > b_i(c), \quad \forall c > \underline{c}_{\Psi,i}, \quad \forall i = 1, \ldots, N.$$

The inequality (C.24) is therefore satisfied for all c satisfying

$$c > \underline{c}_\Psi := \max_i \underline{c}_{\Psi,i},$$

and $-\Psi$ is RDD for these values of c. Finally, both $-\Psi$ and Φ are guaranteed to be RDD for all c satisfying

$$c > \underline{c} := \max(\underline{c}_\Psi, \underline{c}_\Phi).$$

Thus, by Lemma C.1 we have showed that M is a P-matrix for all $c > \underline{c}$. This concludes the proof. □