Diss. ETH No. 27025

Control-Bounded Converters

A thesis submitted to attain the degree of
DOCTOR OF SCIENCE of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

Hampus Malmberg

M.Sc. ETH
born on July 21, 1988
citizen of Sweden

accepted on the recommendation of
Prof. Dr. Hans-Andrea Loeliger, examiner
Prof. Dr. Boris Murmann, co-examiner
Prof. Dr. Hanspeter Schmid, co-examiner

2020

Series in Signal and Information Processing **Vol. 33**

Editor: Hans-Andrea Loeliger

Bibliographic Information published by Die Deutsche Nationalbibliothek

Die Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the internet at http://dnb.d-nb.de.

First Edition 2021

HARTUNG-GORRE VERLAG KONSTANZ

ISSN 1616-671X
ISBN-10: 3-86628-697-X
ISBN-13: 978-3-86628-697-9

Acknowledgments

Firstly, I want to thank my supervisor, Prof. Hans-Andrea Loeliger, for giving me this grand opportunity; for that, I will be forever thankful. Additionally, if it would not have been for his never-ending stream of ideas, intuition, and enthusiasm, this work would have neither been possible nor as rewarding.

Secondly, I want to thank Prof. Boris Murmann and Prof. Hanspeter Schmid for agreeing to co-supervise this thesis and for providing constructive and valuable remarks and suggestions. I also want to thank Prof. Hanspeter Schmid for many insightful discussions during my time as the teaching assistant of his analog signal processing and filtering class.

I have also gained much from my fellow PhD colleagues at the signal and information processing laboratory (ISI): Boxiao Ma, Raphael Keusch, Patrick Murer, Federico Wadehn, Reto Wildhaber, Elizabeth Ren, Robert Graczyk, Nour Zalmai, Christoph Pfister, Tibor Keresztfalvi, Gian Marti, Guy Revasch, Annina Bracher, Lukas Bruderer, Sarah Neff, Christian Schürch, George Wilckens, and Jiun-Hung Yu. In addition to our social encounters, I have collaborated with most of you in one way or another. This has been an absolute joy and has, in many ways, benefited my work. In particular, I want to thank Boxaio, who endured sharing an office with me and always (I hope) had the time to discuss and interact on various topics. Raphael, with whom I collaborated on multiple projects and whose many indulging "side projects" certainly diversified life at ISI. Patrick, with whom I shared many teaching duties and whose razor-sharp feedback, I will always consider the truth. My factor graph sparring partner Federico who holds both the record for my longest ever meeting and the total number of meetings ever held with a single person. Reto, who enthusiastically included me in his exciting research and offered

many useful tricks and hacks in return. Elizabeth, with whom I have shared multiple student projects with and always supported me with feedback. Nour, who always challenged me and thereby pushed both my limits and capabilities.

I am most impressed by and thankful to my father, Pär Malmberg, who carefully reviewed this thesis. His perspective and feedback have been most helpful and ever so encouraging.

Furthermore, I want express my gratitude towards the contributions of Olafur Thoroddsen and Michael Purghart whose master theses contributions helped shape Chapters 8 and 9 and Chapter 7 respectively.

I am very thankful to Patrick Strebel who was an integral part of this thesis's hardware prototype, together with Jonas Biveroni, and has taught me a great deal about circuits.

In addition to my PhD colleagues, I want to thank Prof. Amos Lapidtoh, Prof. Stefan Moser, Simone Ammann, and Rita Hildebrand for contributing to an inspiring, enjoyable, and memorable environment at the ISI lab.

I am most grateful towards my family; my mother, Maria Malmberg, my father, Pär Malmberg, my two brothers, Gustav Malmberg and Petter Nilsson, whose unconditional support throughout my studies have been essential.

Finally, I want to thank my girlfriend, Stina Wargert, with whom I shared the many highs and lows that this PhD adventure has brought. If it were not for her support, both technical and otherwise, this thesis would, if at all, have been written at a much later time.

Abstract

The need for analog-to-digital (A/D) and digital-to-analog (D/A) conversion is a ubiquitous part of many of today's practical applications. The research fields of A/D and D/A conversion are multi-disciplinary, involving topics such as discrete- and continuous-time signal processing, circuit theory, and circuit design. State-of-the-art achievements have refined the practical aspects of traditional converter architectures to a point where performance is reaching its physical limits and progress is stagnating.

In this thesis, we present an alternative perspective of analog-to-digital and digital-to-analog conversion called control-bounded conversion. This new perspective utilizes standard circuit components to build up unconventional circuit architectures through a novel theoretical framework between analog and digital. Ultimately, this versatile design principle allows less constrained analog and digital circuit architectures at the expense of a digital post-processing step.

We demonstrate the control-bounded conversion principle by a selection of converter examples. First we consider the chain-of-integrators and the leapfrog analog-to-digital converters, which emphasize the division of the analog and digital parts of a control-bounded analog-to-digital converter. In particular, these examples reveal the global nature of the analog design task compared to the local digital part, which can be decomposed into independently operated, sub-circuits.

Next, the chain-of-oscillators analog-to-digital converter shows how the control-bounded converter can be adapted for the problem of converting non-baseband signals as is common in communication systems. Specifically, the modulation task (frequency shifting) is incorporated into the digital part of the circuit, removing the need for a pre-processing step.

To suppress the influence of circuit imperfections, we introduce the Hadamard analog-to-digital converter that separates the physical and the logical signal dimensions of a control-bounded converter. This separation enables circuit architectures where the sensitivity to component mismatch and thermal noise can be distributed equally throughout the circuit architecture components, thereby minimizing its impact on conversion performance.

The overcomplete digital control shows how the digital part's complexity can be increased, resulting in better conversion performance, without substantially increasing the sensitivity to circuit imperfections. This idea relates to using higher-order quantization but partitions the analog part of the circuit in a novel way.

We demonstrate that the control-bounded analog-to-digital conversion concept can provide improved conversion performance when converting multiple signals jointly as opposed to independent conversion.

Finally, we show how the control-bounded conversion principle can be adopted for digital-to-analog conversion.

Keywords: Analog-to-digital conversion; digital-to-analog conversion; control-bounded conversion; Delta-Sigma modulation; Gaussian message passing; Wiener filter.

Kurzfassung

Die Notwendigkeit einer Analog-Digital (A/D) und Digital-Analog (D/A) Konvertierung ist ein allgegenwärtiger Bestandteil vieler heutiger Praxisanwendungen. Forschungsgebiete der A/D- und D/A-Wandlung sind interdisziplinär und umfassen Themen wie zeitdiskrete und zeitkontinuierliche Signalverarbeitung, Schaltungstechnik und Netzwerkanalyse. Durch die neusten technischen Errungenschaften wurden die praktischen Aspekte traditioneller Umsetzerarchitekturen so weit verfeinert, dass dessen Leistungsfähigkeit an ihre physikalischen Grenzen stößt und der Fortschritt stagniert.

In dieser Arbeit wird eine alternative Perspektive der Analog-Digital- und Digital-Analog-Umwandlung vorgestellt, die als steuerungsbegrenzende Umwandlung (control-bounded conversion) bezeichnet wird. Diese neue Perspektive verwendet Standardschaltungskomponenten, um unkonventionelle Schaltungsarchitekturen anhand eines neuartigen theoretischen Rahmens zwischen der analogen und der digitalen Welt aufzubauen. Letztendlich ermöglicht dieses vielseitige Entwurfsprinzip weniger eingeschränkte analoge und digitale Schaltungsarchitekturen auf Kosten eines digitalen Nachbearbeitungsschritts.

Das Prinzip der steuerungsbegrenzenden Konvertierung wird anhand einer Auswahl von Konverterbeispielen demonstriert. Zuerst werden die Integratorkette und die Leapfrog-Analog-Digital-Wandler behandelt, welche die Aufteilung eines steuerungsbegrenzenden Analog-Digital-Wandlers in einen analogen und einen digitalen Teil hervorheben. Insbesondere zeigen diese Beispiele die globale Natur der analogen Entwurfsaufgabe im Vergleich zum lokalen digitalen Teil, der in unabhängig betriebene Teilschaltungen zerlegt werden kann.

Danach wird mit dem Oszillatorenkette-Analog-Digital-Wandler (chain-of-oscillator ADC) aufgezeigt, wie der steuerungsbegrenzende Wandler angepasst werden kann für das in Kommunikationssystemen übliche Problem der Konvertierung von nicht bandbegrenzten Signalen. Dabei wird die Modulationsaufgabe (Frequenzverschiebung) in den digitalen Teil der Schaltung integriert, wodurch die Notwendigkeit eines Vorverarbeitungsschritts entfällt.

Um den Einfluss von Unregelmässigkeiten von Schaltungskomponenten zu unterdrücken, wird der Hadamard-Analog-Digital-Wandler eingeführt, welcher die physikalischen und logischen Signal Dimensionen eines steuerungsbegrenzenden Wandlers voneinander trennt. Diese Trennung ermöglicht Schaltungsarchitekturen, bei denen die Empfindlichkeit gegenüber Komponentenfehlanpassung und thermischem Rauschen gleichmäßig auf die Komponenten der Schaltungsarchitektur verteilt wird, wodurch deren Auswirkungen auf die Konvertierungsleistung minimiert wird.

Die übervollständige digitale Steuerung zeigt, wie die Komplexität des digitalen Teils erhöht werden kann, was zu einer besseren Konvertierungsleistung führt, ohne dabei die Empfindlichkeit gegenüber Unregelmässigkeiten von Schaltungskomponenten wesentlich zu erhöhen. Diese Idee bezieht sich auf die Verwendung von Quantisierung höherer Ordnung, aber teilt jedoch den analogen Teil der Schaltung auf neuartige auf.

Es wird aufgezeigt, dass das steuerungsbegrenzende Analog-Digital-Wandlungskonzept eine verbesserte Konvertierungsleistung bieten kann, wenn mehrere Signale gemeinsam anstatt unabhängig konvertiert werden.

Abschließend wird gezeigt, wie das Prinzip der steuerungsbegrenzenden Wandlung für die Digital-Analog-Wandlung übernommen werden kann.

Stichworte: Analog-Digital-Umsetzer; Digital-Analog-Umsetzer; steuerungsbegrenzende Wandler; Delta-Sigma modulation; Gaussian message passing; Wiener filter.

Contents

List of Figures

List of Symbols

Matrix and Vector Operations

a	a scalar value
$\boldsymbol{a}$	a column vector $\begin{pmatrix} a_1 & \dots & a_N \end{pmatrix}^\mathsf{T} \in \mathbb{R}^N$
$\boldsymbol{A}$	a Matrix $\begin{pmatrix} a_{11} & \dots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{M1} & \dots & a_{MN} \end{pmatrix} \in \mathbb{R}^{M \times N}$
$()^\mathsf{T}$	transpose
$()^\mathsf{H}$	Hermitian transpose
$\lvert a \rvert$	absolute value
$\lVert \boldsymbol{b} \rVert_p$	p-norm $\left(\sum_i \lvert b_i \rvert^p\right)^{1/p}$
$\lVert \boldsymbol{c} \rVert_\infty$	max norm, equivalent to max $\begin{pmatrix} \lvert c_1 \rvert, & \dots, & \lvert c_N \rvert \end{pmatrix}$
$\langle \boldsymbol{u}, \boldsymbol{v} \rangle$	inner product
$\otimes$	Kronecker product
$\boldsymbol{I}_k$	a k-by-k matrix with ones on the main diagonal and all other elements being zero
$\mathbf{0}_{k \times \ell}$	a k-by-ℓ matrix with all zero elements
$\mathbf{1}_{k \times \ell}$	a k-by-ℓ matrix with all one elements
$\boldsymbol{R}(\phi)$	rotation matrix $\begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix}$
$\operatorname{rank}(\boldsymbol{A})$	rank of matrix $\boldsymbol{A}$
$\operatorname{diag}(\boldsymbol{a})$	diagonal matrix with the $\boldsymbol{a}$ vector on the main diagonal and all other elements being zero

Sets

$\mathbb{Z}$	the integers
$\mathbb{Z}^+$	the positive integers
$\mathbb{R}$	the real numbers
$\mathbb{C}$	the complex numbers

Miscellaneous

$\dot{\boldsymbol{x}}(t)$	elementwise time derivative $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t)$
$(\boldsymbol{a} * \boldsymbol{b})(t)$	elementwise convolution
$\mathcal{L}_2$	finite energy functions $\int_{\mathbb{R}} \lvert f(\tau)\rvert^2 \,\mathrm{d}\tau < \infty$
$X(\omega)$	continuous-time Fourier transform $\int x(t)e^{-i\omega t}\,\mathrm{d}t$
$X(e^{i\Omega})$	discrete-time Fourier transform $\sum_{k\in\mathbb{Z}} x[k]e^{-i\Omega k}$

Control-Bounded Conversion

T	control period
T_s	sample period
$\mathcal{B}$	frequency band of interest
$\boldsymbol{A}$	system matrix
$\boldsymbol{B}$	input matrix
$\boldsymbol{C}$	signal observation matrix
$\boldsymbol{\Gamma}$	control input matrix
$\tilde{\boldsymbol{\Gamma}}$	control observation matrix
$\boldsymbol{u}(t)$	input signal
$\boldsymbol{u}[k]$	input samples
$\hat{\boldsymbol{u}}(t)$	estimated input signal
$\tilde{\boldsymbol{u}}(t)$	fictional input signal
$\boldsymbol{x}(t)$	state vector
$\hat{\boldsymbol{x}}[k]$	estimated state vector
$\boldsymbol{s}(t)$	control contribution
$\boldsymbol{s}[k]$	control signal
$\tilde{\boldsymbol{s}}(t)$	control observation
$\boldsymbol{y}(t)$	signal observation
$\breve{\boldsymbol{y}}(t)$	fictional signal observation
$\boldsymbol{G}(\omega)$	analog transfer function (ATF) matrix
$\boldsymbol{g}(t)$	analog impulse response matrix

$\boldsymbol{H}(\omega)$	noise transfer function (NTF)
$\boldsymbol{T}(\omega)$	signal transfer function (STF)
$d(t)$	DAC waveform
$\boldsymbol{D}(t)$	DAC waveform matrix
$b_{\boldsymbol{u}}$	largest permissable elementwise $\boldsymbol{u}(t)$ value
$b_{\boldsymbol{x}}$	largest permissable elementwise $\boldsymbol{x}(t)$ value

Acronyms

A/D analog-to-digital

ADC analog-to-digital converter

AS analog system

ATF analog transfer function

BIFM backward information filter, forward marginal

bits binary digits

D/A digital-to-analog

DAC digital-to-analog converter

DC digital control

DE digital estimator

dB decibel

dBFS decibel full scale

ENOB effective number of bits

FFT fast Fourier transform

FIR finite impulse response

IIR infinite impulse response

IVP inital value problem

MASH multi-stage noise shaping

NTF noise transfer function

ODE ordinary differential equation

OSR oversampling ratio

PSD power spectral density

SDE stochastic differential equation

SFDR spurious-free dynamic range

SNDR signal-to-noise and distortion ratio

SNR signal-to-noise ratio

STF signal transfer function

Chapter 1

Introduction

We live in an analog world flooded by digital interactions. At the present state of technology, it is hard to comprehend the synergies between these two worlds. The interconnection is enabled by messages, called signals, that act as intermediaries between entities in either world. As both the digital and the analog world are of a vastly different nature, crossing from one to the other requires a conversion process. We call the process of converting an analog signal into a digital signal, analog-to-digital (A/D) conversion, and similarly, the reversed process digital-to-analog (D/A) conversion.

In this work, we will primarily focus on A/D conversion. However, most of the presented ideas also apply to D/A conversion, as will be covered in the last part of this thesis.

The field of A/D conversion is well studied, and most of today's analog-to-digital converters (ADCs) derive from the elegant mathematical framework known as sampling theory. Unfortunately, some of the involved operations are impossible to implement using electrical circuits. As a result, in an implementation the theory is skillfully approximated by sophisticated engineering that pushes the involved circuitry towards its physical limits.

However, the sampling theory perspective is not the only theoretical framework capable of describing the A/D conversion problem. In this thesis, we pursue another promising perspective that we refer to as control-

bounded A/D conversion. In essence, this theoretical framework defines a new conceptual interface between analog and digital that divides the A/D conversion task according to the strengths of each domain. This can be seen as an intermediate step between sampling theory and practice, where the actual conversion process is designed with the physical properties of the underlying circuits in mind and the end result is a digital object from which we can sample.

The control-bounded ADC can informally be thought of as an analog system (electronic circuit) that performs various fundamental analog operations (additions, subtractions, derivatives, and integrations) with the purpose of amplifying an input signal fed into the system. The analog operations are such that, when fed an input signal, the internal state of the analog system quickly becomes overloaded, or equivalently grows outside its permissible range of operation. The analog system is prevented from overloading, by a digital control. The digital control's operation is primitive as it only observes low-resolution partial snapshots of the internal analog system states at fixed points in time. Based on these crude observations, the digital control interacts with the analog system using control actions to counteract the internal analog-system state growth. In other words, the digital control stabilizes the analog system via one or potentially many control loops. The digital control might be primitive, but as it systematically offloads fixed-sized portions of the accumulated internal analog system states over time, its combined effect results in a sophisticated digital representation of the internal analog system state trajectory.

The beauty of this approach is that, while the focus is to stabilize an analog system using digital control, it implicitly amounts to an analog-to-digital conversion process. Specifically, an estimate of the input signal can be obtained by solving an inverse problem, i.e., estimating what the input signal must have been in order to have triggered the specific sequence of control actions for the given analog system. Combining complex analog systems with multiple primitive digital controls results in remarkably precise estimates of the input signal, as will be demonstrated repeatedly throughout the examples given in this thesis.

Ultimately, control-bounded A/D conversion enables a large and mostly unexplored design space where the actual circuit design can be optimized for new and more practical circuit criterions, (potentially) exceeding the performance limits of conventional methods. A major part of this

thesis is committed to developing and proposing new ADC architectures from the control-bounded perspective. These examples are intended not only to show actual implementations but also to demonstrate the control-bounded converter's rich design and feature space.

1.1 Outline of the Thesis

This thesis is organized in three parts. The first, Chapters 2-4, introduces the fundamental concept, background, and generalized tools for describing, simulating, and evaluating the control-bounded ADC. The second part, Chapters 5-10, presents a series of control-bounded ADC examples. These examples are chosen such that they demonstrate important features of the control-bounded design space. Finally, the third part, Chapter 11 shows how the control-bounded principle can be adopted for D/A conversion.

Part I - Generalized Control-Bounded A/D Conversion

- Chapter 2 - Describes the fundamental problem of A/D conversion. In particular, we try to highlight the role of sampling and motivate the potential of unconventional approaches to A/D conversion.
- Chapter 3 - Gives a brief overview of conventional approaches to A/D conversion and introduces standard notation and concepts when characterizing the conversion performance of an ADC.
- Chapter 4 - Introduces the generalized control-bounded A/D conversion concept.

Part II - Control-Bounded A/D Conversion Examples

- Chapter 5 - Covers the chain-of-integrators control-bounded ADC. This example nicely emphasizes the separation between the analog and digital part of the ADC. The chain-of-integrators serves as the default example from which the following examples extends.
- Chapter 6 - Shows the leapfrog ADC, which in turn demonstrates one way that the analog part of the ADC can be enhanced via pole and zero placement.

- Chapter 7 - Adopts the control-bounded A/D conversion concept for signals that do not reside at the baseband frequencies. Specifically, we show a similar structure as in Chapter 5 called the chain-of-oscillators ADC.
- Chapter 8 - Focuses on how the analog part of the converter can be made more robust against imperfections such as component mismatch, thermal noise and limit cycles. The proposed solution is referred to as the Hadamard ADC.
- Chapter 9 - Establishes a digital control principle that interacts with the analog part through many independent but overlapping digital control paths. This concept relates to A/D conversion with higher-order quantizers but partitions the analog part of the ADC in a new way. The proposed digital control principle is referred to as an overcomplete digital control.
- Chapter 10 - Demonstrates how multiple control-bounded ADCs can be combined such that multiple input channels can be converted jointly, resulting in better overall performance.

Part III - Control-Bounded D/A Conversion

- Chapter 11 - Extends the control-bounded A/D conversion concept to D/A conversion. Specifically, all previously presented examples can be repurposed into digital-to-analog converters (DACs).

1.2 Contributions

The main contribution of this thesis is the advancement of the control-bounded ADC conversion concept. In particular, we develop analytical tools, derive fundamental properties, and demonstrate several unique features to this A/D conversion principle. As a result, the control-bounded A/D converter has matured into an intuitive and capable design paradigm that resonates with fundamental circuit theory.

Among the many individual contributions throughout this thesis, we want to highlight particularly:

- The overcomplete digital control in Chapter 9 that divides the control task into multiple components such that its complexity can be scaled robustly. The overcomplete digital control is also a

fundamental building block towards non-traditional A/D conversion scenarios such as multi-channel A/D conversion.

- The chain-of-oscillator ADC in Chapter 7 that extends the control-bounded A/D conversion concept for non baseband applications. Specifically, it uses modulation in the digital control that opens up new ways of building and scaling A/D converters for single or multi-band A/D conversion.
- The Hadamard ADC from Chapter 8 shows how the analog signal representation can be distributed uniformly among the involved circuit components to achieve a robust circuit implementations.

1.3 Related Work

The control-bounded A/D conversion concept is based on the work from [2–5, 17, 38] and was first presented in its current form in [19, 20].

A substantial effort has been made to adopt conventional performance metrics to the control-bounded A/D conversion concept. As part of this process, we have relied on [8, 9, 25, 32] to make meaningful comparisons to state-of-the-art ADCs. Additional related work will follow in Chapter 3.

Chapter 2

A Representation Problem

This chapter has two goals. Firstly, to define the A/D conversion problem and thereby serve as background for those not familiar with the concept. Secondly, to highlight where the current theory and practice diverge and thereby motivate the benefits of alternative approaches.

Formally, A/D conversion is the process when an analog signal $u\colon \mathbb{R} \to \mathbb{R}$ is converted into a digital representation u_R, i.e., a finite collection of binary digits (bits). Ideally, u_R is such that u is uniquely described by it. However, finding such a representation is a fundamentally ill-posed problem since, in a general analog setting, u is one out of infinitely many signals and would require a digital representation consisting of an infinite number of bits.

Instead, the process of A/D conversion can be thought of as finding a digital representation from which we could construct an approximate signal $\hat{u} : \mathbb{R} \to \mathbb{R}$ that best resembles u with respect to some cost function. This is illustrated in Figure 2.1. The construction of an approximate

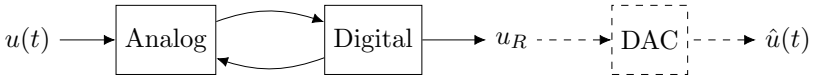


Figure 2.1: Conceptual A/D conversion.

signal $\hat{u}(t)$ is indicated in the figure by the dashed DAC block. This does not mean that a DAC accompanies every ADC. However, we think of the A/D conversion as being approximately reversible by some, at least conceptual, inverse mapping $\hat{u}(t) = \mathrm{DAC}(u_R)$.

Figure 2.1 also indicates that A/D conversion has an analog part and a digital part. In particular, the analog part applies a preconditioning operation to limit the possible input signals into a subset $\mathcal{U}$. The operation simplifies the A/D interface by ensuring certain properties of the signals in $\mathcal{U}$.

An example of a preconditioning operation would be to limit the input signals to the set of bandlimited signals using an anti-aliasing filter. In a more general view the preconditioning operation symbolizes the physical limitations of the underlying analog circuitry. Note that $\mathcal{U}$ might still contain an infinite number of signals.

2.1 Sampling Theory

After the preconditioning operation the next step in the A/D conversion process is the actual conversion. Here there are multiple approaches as the progression of A/D conversion field has inspired numerous competing techniques for finding u_R given u, some of which will be mentioned in Chapter 3. These ADCs have at least one thing in common, and that is that they all result in a digital representation u_R that is a sequence of samples $u[k]$, i.e., a series of fixed-point or floating-point numbers representing $u(t)$ evaluated at different times t. This representation is not a coincidence and brings us to the topic of sampling theory.

Classical sampling theory [15, 27, 37] is a closely related field to A/D conversion where one seeks the solution to the problem of perfectly reconstructing a function $x\colon \mathbb{R} \to \mathbb{R}$ using only a sequence of samples, i.e. x evaluated at some points in its domain. One of the most classical results is the Shannon-Nyquist sampling theorem [10] which states that:

Theorem 1. *(The Shannon-Nyquist theorem) For a function $x \in \mathcal{L}_2$ with a continuous-time Fourier transform $X(\omega)$, that is bandlimited by the frequency $1/T_{\mathcal{B}}$, i.e. $X(\omega) = 0$ for all $|\omega| \geq \pi/T_{\mathcal{B}}$, the function $x(t)$ can be uniquely described by the samples $x(\ell T_{\mathcal{B}})$ as*

$$x(t) = \sum_{\ell \in \mathbb{Z}} x(\ell T_{\mathcal{B}}) \operatorname{sinc}\left(\frac{t - \ell T_{\mathcal{B}}}{T_{\mathcal{B}}}\right) \tag{2.1}$$

where

$$\mathrm{sinc}(t) \triangleq \frac{\sin(\pi t)}{\pi t}. \tag{2.2}$$

Using samples to represent the analog signal $u(t)$ is a natural thing and is convenient for further digital processing.

Furthermore, the sampling concept is far more general than bandlimited signals and uniform samples. Some examples are the concept of wavelets, or more generally frames [21,31], that could efficiently represent other sets than bandlimited signals. Regardless which form, representing analog signal using samples (coefficients), corresponding to signal basis functions, is the natural output data type of an ADC.

Sample-Centric Analog-to-Digital Conversion

For good reasons, the sampling theorem has had an enormous impact on the A/D conversion community, which is evident as most of today's ADCs are preconditioned by an anti-aliasing filter and a sampling stage. In this view, the A/D conversion process can be thought of as a grid search, as illustrated in Figure 2.2. The figure shows a snapshot of a bandlimited analog signal (in red) as a function of time. The grid in the figure represents the discretization in both time and amplitude. The vertical lines of the grid correspond to sampling times, and therefore, only the analog signal evaluated at these lines (red crosses) are accessible to the actual conversion process. The horizontal lines correspond to the different digital representation, and thus the A/D conversion process amounts to assigning the signal, for each sample (red cross), to the closest grid point as indicated by the blue markers in the figure. This intuitive approach has its advantages as, among other things, the separation in time and amplitude allows us to solve the involved operations in sequential steps.

2.2 The Proposition

The inconvenient truth is that sampling theory, at least in the classical sense, does not entirely address the A/D conversion problem [14]. Specifically, individual samples still need an unrealistic digital representation with an infinite number of bits to represent the signal u perfectly. In other words, the samples $u_R[k]$ cannot be perfectly represented, which raises the question of whether sampling is a restriction, rather than an

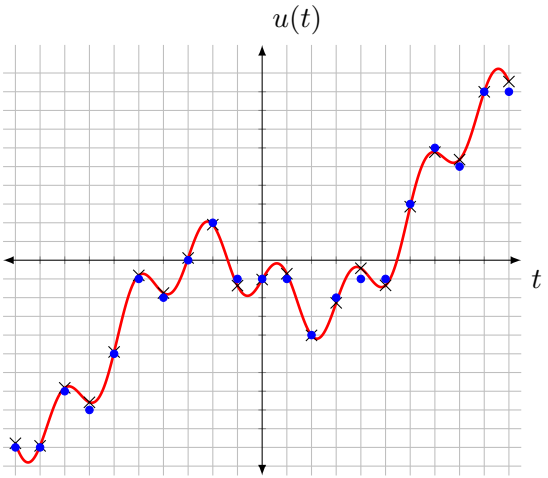


Figure 2.2: A sampling and quantization grid where the A/D conversion process, of the signal $u(t)$ (in red), amounts to: for every time step (black crosses) choosing the closets amplitude grid points (blue dots) of the signal.

enabler, as a part of the preconditioning step in the A/D conversion process. This proposition might seem outrageous since sampling reduced the A/D conversion problem from handling continuous-time analog signals to working with discrete-time analog samples.

However, there are arguments for alternative approaches such as the control-bounded ADC concept that is the main topic of this thesis. One argument for abandoning the sample-centric view is that sampling theory strives to represent as much information (signal) as possible using as few samples as possible. However, in the world of circuits, there is no such thing as a perfect measurement. Instead, the task of the ADC can be better described as trying to collect as much signal information as possible, using imperfect components of partly unknown values, in an environment that is naturally filled with heavy interferers. This means that the conversion task needs an A/D interface that is not only theoretically sound but also provides redundancy, and thereby robustness, towards the imperfections faced in a practical implementation.

One way of creating redundancy is to oversample, i.e., to take more

samples than what is strictly necessary according to the sampling theorem. This technique is employed by several conventional ADC architectures, see Section 3.2. However, there is no theoretical principle that proposes this to be an optimal, or even efficient, way of creating redundancy (seen from the overall number of bits used in the digital representation). Furthermore, this principle cannot be scaled unconditionally as for a large enough sampling frequency, the sampler's precision becomes the bottleneck of such an ADC architecture.

An alternative way of creating redundant digital representations is the concept of control-bounded ADC presented in Chapters 4-10. These are examples of non-sample-centric architectures where redundancy is naturally induced by considering the conversion process as stabilizing an analog system with a digital control rather than directly reading out sampled and quantized signal values. Specifically, since the digital control's goal is not to cancel the internal analog state but instead maintain bounded internal analog system states, each digital control interaction, most likely, results in a non-zero analog state remainder. This means that the impact of a digital control decision, at any given time, potentially impacts all future such decisions. Furthermore, as there is a substantial overlap between the effect of digital control decisions taken at different times, there might be many such control decision sequences that effectively suppress a given input signal. In other words, we recognize that merely controlling a state within a bound, as opposed to directly observing and or canceling signal contribution, implicitly creates highly redundant digital representations.

A consequence of using redundant digital representations is that several of the otherwise performance-critical aspects of a conventional converter, such as anti-aliasing filtering, sampling, and quantization, are relaxed or otherwise implicit. For instance, as the digital control makes its decisions based on the internal states of the analog system, and these states are the result of amplifying wanted signal characteristics of the sought signal, out-of-band components, that would otherwise need to be filtered out prior to conversion, are implicitly suppressed by the analog system itself. Furthermore, in the case of many independent digital controls, it is not the quantization and sampling specification of a single control that determines the system's performance. Instead, it is the cumulative control effort that relaxes the constraints on any single quantizer or sampler.

We believe that the control-bounded converter concept presents an op-

portunity that would benefit the circuit designer and provide a powerful post-processing and sampling platform. Specifically, the circuit design boils down to realizing a robust and performant analog design from continuous-time analog specifications using digital control circuits to stabilize the system. Furthermore, this design task would not require advanced digital signal processing, such as continuous-time to discrete-time conversions. Instead, the fundamental conversion performance is implicit as long as the system provides sufficient open-loop analog amplification and a digital control stabilizes the system. In a separate step, the sought signal representation is extracted from the highly redundant digital representation produced by the digital control(s), as we perform a post-processing digital filtering step. Here the redundant representation is converted into a signal representation (samples) that is best suited for the given application. We think of this process as first combining the redundant representation into a digital object, representing the continuous-time evolution of the input signal. From this digital object, we can then sample at arbitrary times and therefore change sampling rates with ease and produce complex sampling patterns if needed. It is worth pointing out that this post-processing step reduces to a linear filter for uniform samples and can therefore be implemented without excessive computational requirements.

In summary, the control-bounded A/D conversion concept provides a novel interface for the A/D conversion process, which gives much design freedom for both the converter's analog and digital parts, thus promoting each domain's strengths while only imposing minor restrictions on their interconnection.

Chapter 3

Conventional Analog-to-Digital Conversion

Analog-to-digital conversion is a well-researched field where not only one but multiple techniques co-exist among state-of-the-art approaches. Some examples are flash converters, sub-ranging converters, successive approximation converters, integrating converters, and $\Delta\Sigma$ modulators. They all have different advantages and shortcomings and therefore target different applications.

The content presented in this chapter is, for the most part, standard in the A/D community and is only repeated here to establish the necessary terminology and language to make meaningful comparisons to the control-bounded ADC presented in the succeeding chapters. Therefore, we will not go into specifics of particular ADCs mentioned above. The only exception is the continuous-time $\Delta\Sigma$ modulators, covered in Section 3.3. For in-depth description of conventional ADCs, the reader is directed to [25, 32] and references within.

3.1 Sample-per-Sample Converters

The classical, sample-centric, view on A/D conversion dictates a conversion chain as shown in Figure 3.1. This perspective divides the conversion task into three separate operations: preconditioning (anti-aliasing filtering), sampling, and quantizing.

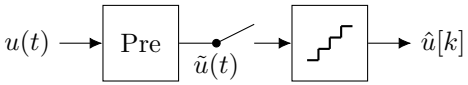


Figure 3.1: The sample-centric view on A/D conversion.

The preconditioning task is an analog filter. As this operation precedes the others, any imperfections or noise is directly destructive for the conversion process. Therefore, it is paramount to implement the analog filter with sufficient precision.

Next in the conversion chain is the sampler. The sampler converts the continuous-time signal into a discrete-time signal by extracting snapshots of $\tilde{u}(t)$ at specific times. The performance of the sampler depends on a stable clock as clock jitter causes jitter-induced noise. Furthermore, at high sampling rates, the energy consumption becomes substantial as a result of switching losses.

The last step in the conversion chain is the quantizer. The quantizer's task is to discretize the sampled signal's amplitude and thereby map the sampled signal to a digital representation. For sample-per-sample converters, much of the design effort goes into realizing the quantizer with sufficient resolution. This task is highly non-trivial. The quantization is also where most of the state of the art converters diverge in their approach.

For example, the flash converter realizes the quantizer by simultaneously comparing its input with predefined references and outputting a digital representation corresponding to the closest reference. Similarly, a subranging converter repeats the previously mentioned strategy in multiple rounds and additionally "zooms" by subtracting the determined reference and amplifies the resulting signal in between rounds. Notice that the subranging conversion process still is a sample-per-sample type conversion since only one sample at a time is processed even though the quantization process can stretch multiple clock periods.

3.2 Oversampling Converters

Another approach is to extend the quantization task to incorporate multiple samples. In other words, a form of vector quantization. This approach is referred to as oversampling and is popular for high-resolution applications.

The simplest version of an oversampling converter quantizes each sample individually, as before, and then combines the digitized samples by averaging, i.e., some sort of low-pass filter. The success of this approach is dependent on the fact that the converted signal resides in a sub-band of the Nyquist bandwidth, i.e., the sample rate is much higher than the Nyquist rate.

Better performance can be attained by not quantizing the samples one by one but instead introduce a feedback path from the quantizer output to its input, possibly inserting a loop filter $G(e^{i\Omega})$ in between as shown in Figure 3.2. Such a system is known as a $\Delta\Sigma$ modulator. The loop filter, together with the feedback path, suppresses the error in some frequency bands and thereby enhances the resolution in the same frequency bands. This concept is known as "noise shaping". The name comes from the fact that when the quantization error is modeled as an additive noise term, the feedback loop together with the loop filter effectively shapes the power spectral density (PSD) of the quantization noise seen at the output.

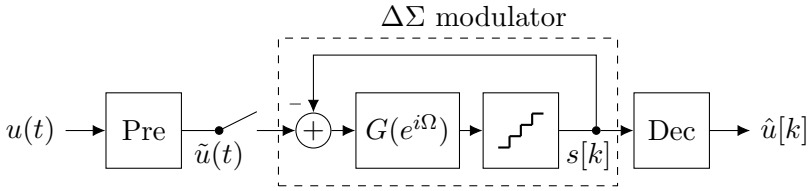


Figure 3.2: Discrete-time $\Delta\Sigma$ modulator including the decimation filter.

For frequencies not suppressed by the loop filter and feedback path, the conversion error is unaltered. This part of the estimate's frequency spectrum is referred to as out-of-band noise and typically contains a substantial part of the estimate's signal energy. Therefore, a $\Delta\Sigma$ modulator requires an additional post-processing step to suppress the out-of-band noise generated by the modulator. This additional post-processing step is referred to as decimation filter, marked (Dec) in Figure 3.2. In addition

to filtering out the quantization error, the decimation filter downsamples the signal to the Nyquist sample rate.

3.3 Continuous-Time Delta-Sigma Modulation

There exists a continuous-time version of the $\Delta\Sigma$ modulator which is shown in Figure 3.3 [22]. In this version, the preconditioning task, as well

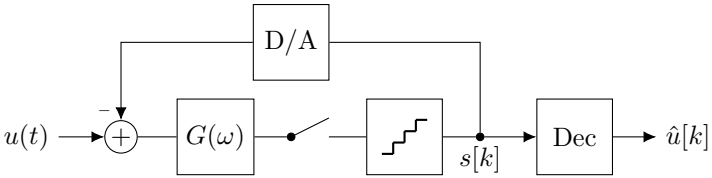


Figure 3.3: Continuous-time $\Delta\Sigma$ modulator.

as the sampling, are included in the feedback loop of the $\Delta\Sigma$ modulator. This approach is particularly interesting since the task of sampling and quantizing are joined via the feedback loop. Additionally, even though it is not the standard, the preconditioning task can be incorporated in the loop filter $G(\omega)$.

The continuous-time $\Delta\Sigma$ modulator is thus a hybrid between a continuous-time and discrete-time system. To analyze the $\Delta\Sigma$ modulator from Figure 3.3 the non-linear quantizer is replaced by an additive discrete-time error signal $z[k]$ denoted the quantization error signal. This is illustrated in Figure 3.4. From the linearized model, in Figure 3.4, we

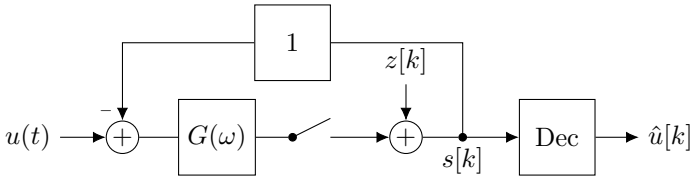


Figure 3.4: Linearized model of a continuous-time $\Delta\Sigma$ modulator.

recognize that the quantization error signal and input signal reach the estimate $\hat{u}[k]$ via different paths and therefore have different transfer

functions. The system's continuous/discrete-time nature complicates the analysis compared to a fully discrete-time system, as is the more typical $\Delta\Sigma$ modulator. The transfer function analysis for the given system is worked out in detail in Appendix B. For a bandlimited input signal $u(t)$ and loop filter $G(\omega)$, the analysis results in the noise transfer function (NTF) and signal transfer function (STF)

$$\mathrm{NTF}(e^{i\Omega}) = \frac{S(e^{i\Omega})}{Z(e^{i\Omega})} \tag{3.1}$$

$$= \frac{1}{1 + L_1(e^{i\Omega})} \tag{3.2}$$

and

$$\mathrm{STF}(e^{i\Omega}) = \frac{S(e^{i\Omega})}{\tilde{U}(e^{i\Omega})} \tag{3.3}$$

$$= \frac{L_0(e^{i\Omega})}{1 + L_1(e^{i\Omega})} \tag{3.4}$$

where $S(e^{i\Omega})$ is the discrete-time Fourier transform of $s[k]$, $Z(e^{i\Omega})$ is the discrete-time Fourier transform for $z[k]$, $\Omega = \omega T_s$. Note that the derivations assume an unity quantization gain and that the dynamical range of the quantizer scale with the choice of $G(\omega)$. Furthermore,

$$\tilde{U}(e^{i\Omega}) \triangleq \frac{1}{T_s} \sum_{k\in\mathbb{Z}} U\left(\frac{\Omega - 2\pi k}{T_s}\right), \tag{3.5}$$

$$L_0(e^{i\Omega}) \triangleq \sum_{k\in\mathbb{Z}} G\left(\frac{\Omega - 2\pi k}{T_s}\right), \tag{3.6}$$

$$L_1(e^{i\Omega}) \triangleq \frac{1}{T_s} \sum_{k\in\mathbb{Z}} G\left(\frac{\Omega - 2\pi k}{T_s}\right) D\left(\frac{\Omega - 2\pi k}{T_s}\right), \tag{3.7}$$

T_s is the sample period, $U(\omega)$ is the continuous-time Fourier transform of $u(t)$, and $D(\omega)$ is the continuous-time transfer function of the D/A converter's impulse response.

Notice that the transfer function does not include the decimation filter Dec. This is intentional as the decimation filter is a post-processing step that is typically addressed separately. The ideal effect of the decimation filter would be a brick wall filter, i.e., only passing through the frequency bands corresponding to the Nyquist rate of the input signal.

3.4 Performance Measures

One of the key performance measures of an ADC is the effective amplitude resolution of the samples. For an ideal sample-per-sample converter circuit, the effective resolution is determined by the number of bits used for each digital codeword in the quantizer. In reality, the effective resolution is not necessarily limited by the number of bits used but rather by imperfections such as thermal noise, component mismatch, sampling jitter, and non-linearities. Therefore, we use the term conversion error that amounts to the total effect of all errors seen at the output samples of the converter. One of the contributors to the conversion error is the previously described quantization error that will be covered in Section 3.4.3.

Furthermore, the effective resolution is described using the signal-to-noise ratio (SNR). This means that the resulting samples are decomposed into a signal component $\hat{u}[k]$ and a conversion error component $\epsilon[k]$. Furthermore, the mean squared value of both these discrete-time signals are computed, and the SNR is defined as

$$\mathrm{SNR} \triangleq \frac{P_{\hat{u}}}{P_{\epsilon}} \tag{3.8}$$

where $P_{\hat{u}}$ is the mean squared value of the estimated input signal, and P_{ϵ} is the mean squared value of the conversion error, respectively. The SNR is typically expressed in decibels which will be denoted $\mathrm{SNR}_{\mathrm{dB}}$.

For oversampling converters, as in Section 3.2, half the Nyquist frequency ($f_{\mathcal{B}}$), or equivalently the bandwidth of the input signal, is typically much smaller than the sampling frequency $f_s \triangleq 1/T_s$. This is typically described using the oversampling ratio (OSR)

$$\mathrm{OSR} \triangleq \frac{f_s}{2f_{\mathcal{B}}}. \tag{3.9}$$

We refer to the portion of the frequency band determined by the Nyquist rate as the frequency band of interest, and define it as

$$\mathcal{B} \triangleq \{2\pi f : |f| \leq f_{\mathcal{B}}\}. \tag{3.10}$$

Furthermore, the decimation filter's task is to suppress the out-of-band noise of the estimate. Therefore, when discussing the SNR of a $\Delta\Sigma$ modulator, we only consider the frequency band of interest since the

remaining spectrum is filtered out in a later stage. This can be directly translated to the mean squared value as

$$P_{\hat{u}} = \frac{1}{2\pi} \int_{\omega \in \mathcal{B}} S_{\hat{u}}(e^{i\omega T_s}) \, \mathrm{d}\omega \tag{3.11}$$

and

$$P_{\epsilon} = \frac{1}{2\pi} \int_{\omega \in \mathcal{B}} S_{\epsilon}(e^{i\omega T_s}) \, \mathrm{d}\omega \tag{3.12}$$

where $S_{\hat{u}}(e^{i\omega T_s})$ and $S_{\epsilon}(e^{i\omega T_s})$ are the PSD of the signal and conversion error, respectively.

3.4.1 Sinusoidal Test Signal

From the SNR expression in (3.8) it is clear that the performance depends on the input signal u. Therefore, it is customary to measure the SNR for a given test input signal. For oversampling converters, the standard test scenario is to excite the converter with a full-scale sinusoidal input signal. Subsequently, the SNR can be determined from numerical integration of the estimate's PSD. By a full-scale signal, we refer to a signal where the largest amplitude is the same as the largest permissible amplitude $u_{\max}$ of the system. An example spectrum is shown in Figure 3.5. From the figure a main peak is clearly visible at $\omega T_s/2\pi \approx 0.002$. To compute the SNR, we identify the main peak as well as any harmonics within the frequency band of interest as the signal component of the estimate and compute their combined mean squared value as $P_{\hat{u}}$. Similarly, the remaining frequency bins within the frequency band of interest are identified as noise; their mean squared value results in P_{ϵ}.

A related quantity is the signal-to-noise and distortion ratio (SNDR) which is computed almost identically but where the signal is only identified as the main peak of the spectrum and therefore the harmonics are added to the noise term.

Both these measures are exemplified in Figure 3.6. The figure demonstrates a clear linear relationship between the input signal strength and the SNR performance. Additionally, as the input amplitude comes close to full-scale, the performance drops rapidly. Note that Figure 3.6 is an artificial example and the given SNR and SNDR relationships are manufactured to demonstrate the expected behavior.

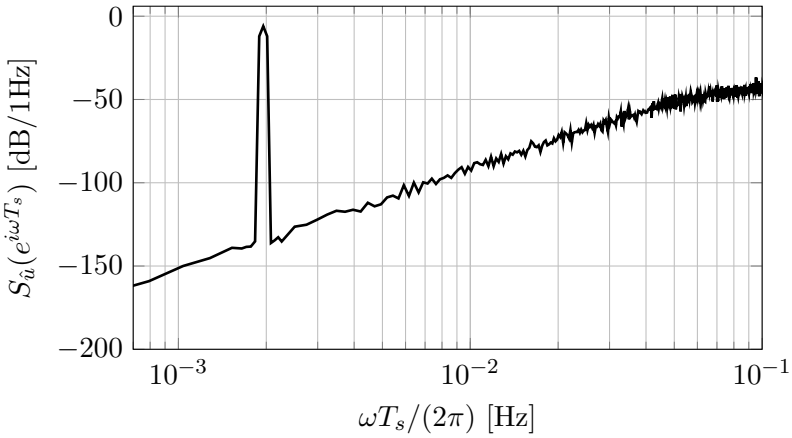


Figure 3.5: The PSD example plot of the estimate $\hat{u}[k]$ for a $\Delta\Sigma$ modulator.

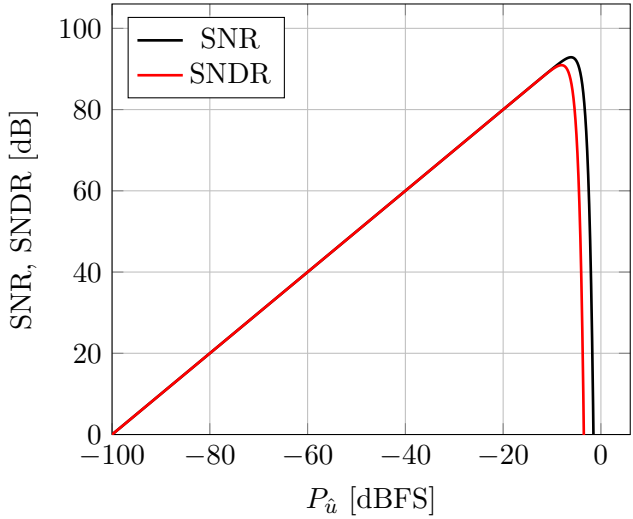


Figure 3.6: Demonstration of typical SNR, SNDR vs input signal power relationship.

3.4.2 Computing the Power Spectral Density

As the PSD is an essential part of evaluating the performance of a $\Delta\Sigma$ modulator, we will next describe this in detail. Furthermore, these computations will also apply to the control-bounded converters in the upcoming chapters, as the same principle determines their performance.

In this thesis, the PSD will only be computed for discrete-time signals with an underlying uniformly spaced sample grid. The discrete-time signals often contain a random component, which makes a directly computed fast Fourier transform (FFT) appear "noisy" and thereby visually troublesome to evaluate. Therefore, instead of directly computing the PSD using a FFT, we estimate the PSD by dividing the discrete-time signal into smaller segments and averaging their corresponding FFTs. Specifically, we use the Welch method as described in [36]. In this thesis, the Welch algorithm is parameterized equally for all PSDs computations using a segment length of $L = 2^{16}$, a Hann data window, and a 50% overlap between segments.

3.4.3 Quantization Error

The quantization error is determined by the distance between the references corresponding to two adjacent digital codewords in the quantizer. Typically, these references are uniformly distributed throughout the permissible input amplitude range of the quantizer, and then the largest possible error is determined as

$$2\Delta_{\max} = \frac{u_{\max} - u_{\min}}{2^b} \tag{3.13}$$

where b is the number of bits used in the codeword and $(u_{\max}, u_{\min})$ represents the largest and smallest permissible input amplitude to the quantizer, respectively. The magnitude of the quantization error at any given time depends on the input signal to the quantizer.

However, for the sake of a tractable analysis, it is standard to model the quantization noise samples as i.i.d. uniformly distributed random variables with a zero-mean and a variance

$$\sigma_q^2 = \frac{\Delta_{\max}^2}{12}. \tag{3.14}$$

The PSD of the quantization noise, under the stated assumptions, follows

as

$$S_q(e^{i\Omega}) = \frac{\Delta_{\max}^2}{24\pi f_s} \tag{3.15}$$

where the noise is assumed to be bandlimited, i.e., having zero energy for $|f| \geq f_s/2$.

3.4.4 Expected SNR of a Delta-Sigma Modulator

The SNR for the continuous-time $\Delta\Sigma$ modulator from Section 3.3 is computed as

$$\mathrm{SNR}_{\Delta\Sigma} = \frac{\int_{\omega\in\mathcal{B}} |\mathrm{STF}(e^{i\omega T_s})|^2 S_u(e^{i\omega T_s})\ \mathrm{d}\omega}{\int_{\omega\in\mathcal{B}} |\mathrm{NTF}(e^{i\omega T_s})|^2 S_\epsilon(e^{i\omega T_s})\ \mathrm{d}\omega}. \tag{3.16}$$

For a sinusoidal input of amplitude A, frequency f, and a quantization error modeled as in Section 3.4.3, the expected SNR can be approximated as

$$\mathrm{SNR}_{\Delta\Sigma} \approx A^2/2 \left(\int_{\omega\in\mathcal{B}} \frac{1}{1 + L_1(e^{i\omega T_s})}\ \mathrm{d}\omega \right)^{-1} 2\pi f_s \frac{12}{\Delta_{\max}^2} \tag{3.17}$$

with the assumption that f is within the frequency band of interest, i.e., assuming $|G(2\pi f)| \gg 1$.

The Taxonomy of a $\Delta\Sigma$ Modulator

As shown in [8], the SNR of a plain N-th order $\Delta\Sigma$ modulator, with a loop filter comprising chains of integrators, can be approximated by

$$\mathrm{SNR}_{\max} \approx \frac{3 \cdot 2^b (2N+1) \mathrm{OSR}^{2N+1}}{2\pi^{2N}} \tag{3.18}$$

where we have assumed a sinusoidal input with amplitude $\frac{u_{\max} - u_{\min}}{2}$ and the quantization error variance as in Section 3.4.3. As indicated by (3.18), the performance can be tuned by changing the number of bits in the quantizer b, the system order N, and the OSR. (3.18) is also often approximated directly in decibel (dB) as

$$\begin{aligned} \mathrm{SNR}_{\max_{\mathrm{dB}}} \approx\ & 6.02b + 1.76 \\ & + 10\log(2N+1) + 10(2N+1)\log(\mathrm{OSR}) \\ & - 20N\log(\pi). \end{aligned} \tag{3.19}$$

The SNR can also be expressed as effective number of bits (ENOB), i.e., the number of bits an ideal quantizer would require to achieve the same SNR. The expression can be derived from (3.18) and is commonly approximated as

$$\text{ENOB} \approx \frac{\text{SNR}_{\text{max}_{\text{dB}}} - 1.76}{6.02}. \tag{3.20}$$

3.4.5 Discrete-Time-to-Continuous-Time Transformation

As nicely covered in [22], the analog loop filter $G(\omega)$ of the continuous-time $\Delta\Sigma$ modulator from Section 3.3 is typically first designed using discrete-time analysis and then transformed into an approximated continuous-time form. This process is called discrete-time-to-continuous-time transformation, and multiple approaches exist. Typically this means that the discrete-time filter is approximated in the time domain, at the corresponding sample points, or in the frequency domain, at discrete frequencies, by a continuous-time filter. Such approximations are, in general, involved as many unknown parameters need to be determined. However, the topic of discrete-time-continuous-time transformations is well supported in the literature, see [22] and references within.

This additional design complexity is often considered a negative attribute of the continuous-time $\Delta\Sigma$ modulator and often outweighs the potential advantages of using continuous-time over discrete-time filters. As a result, the majority of $\Delta\Sigma$ modulators found in both academia and industry are predominantly discrete-time designs.

In contrast to designing a continuous-time $\Delta\Sigma$ modulator, for the control-bounded ADC of the following chapters, see Section 4.5, the analog part of the system is directly considered a continuous-time system. Therefore, we avoid transformations from continuous-time to discrete-time in the corresponding loop filter (analog system) design for the control-bounded perspective.

3.5 MASH Delta-Sigma Converter

Due to the non-linearity introduced by the quantizer and DAC in Figure 3.4 it is not straightforward to determine for what parameter settings a $\Delta\Sigma$ modulator is stable or not. This is especially true for systems with

higher-order loop filters $G(\omega)$ in combination with a low-order quantizer, and low-order DAC. Therefore, the design process requires long transient simulations and tuning to ensure stability. Furthermore, such an empirical validation results in no theoretical stability guarantees.

One approach to remedy this problem is the multi-stage noise shaping (MASH) $\Delta\Sigma$ converter. In this approach, the high-order loop filter is divided into small order filters where each such stage has its own dedicated A/D and D/A feedback loop. An example of a MASH $\Delta\Sigma$ converter is given in Figure 3.7. As shown in the figure, the MASH is described using

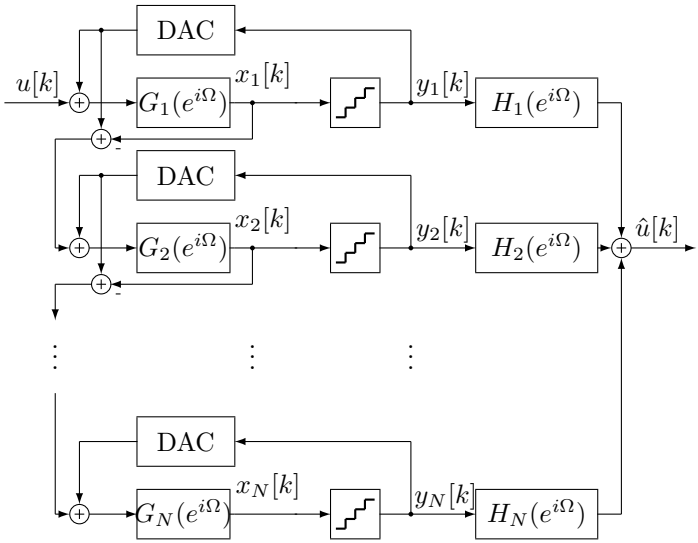


Figure 3.7: A continuous-time MASH $\Delta\Sigma$ modulator.

multiple discrete-time $\Delta\Sigma$ modulator in contrast to the continuous-time version from Section 3.3. The reason for changing the continuous-time to the discrete-time $\Delta\Sigma$ modulator, is that the digital cancellation principle is better illustrated in this domain. However, we do recognize that there exists a continuous-time version of the MASH $\Delta\Sigma$ modulator presented next.

The main concept behind dividing the conversion task into multiple $\Delta\Sigma$ modulators is that, starting from the second stage, the error signal of the previous conversion is the input signal to the next converter stage. This means that $y_1[k]$ is a sampled and quantized version of a filtered version

of $u(t)$ whereas $y_2[k], \ldots, y_N[k]$ are filtered version of the previous stage conversion error.

Digital Cancellation Logic

The multi-output nature of the MASH $\Delta\Sigma$ converter requires a reconstruction step in order to produce a final estimate $\hat{u}[k]$. This step is commonly referred to as the digital-cancellation logic, also represented in Figure 3.7 by the digital filters $H_1(e^{i\Omega}), \ldots, H_N(e^{i\Omega})$. The given structure of the MASH results in very characteristic cancellation logic filters, as we construct the linear system of equations

$$\begin{aligned}
\mathrm{STF}_1(e^{i\Omega})H_1(e^{i\Omega}) &= T(e^{i\Omega}) && (3.21)\\
\mathrm{NTF}_1(e^{i\Omega})H_1(e^{i\Omega}) + \mathrm{STF}_2(e^{i\Omega})H_2(e^{i\Omega}) &= 0 && (3.22)\\
&\vdots\\
\mathrm{NTF}_{N-1}(e^{i\Omega})H_{N-1}(e^{i\Omega}) + \mathrm{STF}_N(e^{i\Omega})H_N(e^{i\Omega}) &= 0 && (3.23)
\end{aligned}$$

where $\mathrm{STF}_1(e^{i\Omega}), \ldots, \mathrm{STF}_N(e^{i\Omega})$ and $\mathrm{NTF}_1(e^{i\Omega}), \ldots, \mathrm{NTF}_N(e^{i\Omega})$ are the STF and NTF of each sub system. The sub systems are constructed similarly as in (3.4) and (3.2) with the modification that

$$\begin{aligned}
\mathrm{STF}_1(e^{i\Omega}) &= \frac{Y_1(e^{i\Omega})}{\tilde{U}(e^{i\Omega})}, && (3.24)\\
\mathrm{STF}_2(e^{i\Omega}) &= \frac{Y_2(e^{i\Omega})}{Z_1(e^{i\Omega})}, && (3.25)\\
&\vdots\\
\mathrm{STF}_N(e^{i\Omega}) &= \frac{Y_N(e^{i\Omega})}{Z_{N-1}(e^{i\Omega})}, && (3.26)
\end{aligned}$$

and

$$\mathrm{NTF}_\ell(e^{i\Omega}) = \frac{Y_\ell(e^{i\Omega})}{Z_\ell(e^{i\Omega})}. \qquad (3.27)$$

Furthermore, $T(e^{i\Omega})$ represents the target transfer function for the input signal. This is typically a simple filter like a single sample delay. What the equation system describes is the cancellation of the previous conversion error. Note that a perfect cancellation is only possible in the discrete-time version of the MASH $\Delta\Sigma$ converter. For the continuous version, we can

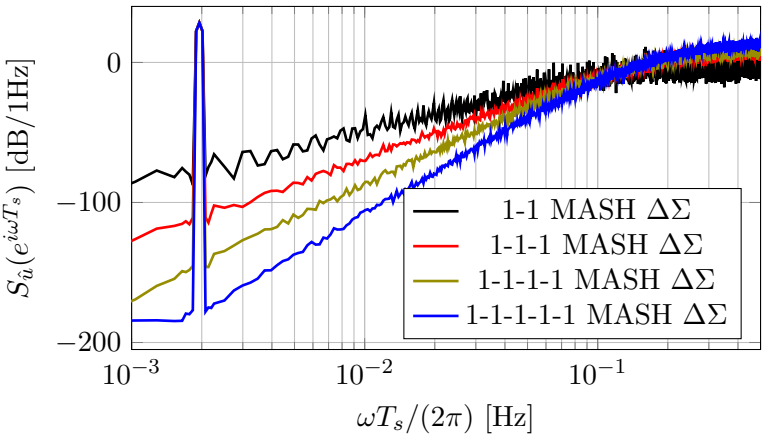


Figure 3.8: The PSD of a MASH ΔΣ Converter as in Figure 3.7 were the loop filters $G_1(\omega), \ldots, G_N(\omega)$ are first-order analog systems and we use one-bit quantizers. The notation (1-...-1) represents different MASH configurations where the number indicates the loop filter system order, and each number represents a node in the MASH structure.

only approximate cancellation at the sampling times. Regardless the specified system of equations results in

$$\hat{U}(e^{i\Omega}) = T(e^{i\Omega})\tilde{U}(e^{i\Omega}) + H_1(e^{i\Omega})\frac{\prod_{\ell=1}^{N}\mathrm{NTF}_\ell(e^{i\Omega})}{\prod_{m=2}^{N}\mathrm{STF}_m(e^{i\Omega})}Z_N(e^{i\Omega}) \quad (3.28)$$

as we solve the systems of equations for $H_1(e^{i\Omega}), \ldots, H_N(e^{i\Omega})$. From (3.28) we see that the estimate only has one error term, the conversion error from the last stage $Z_N(e^{i\Omega})$. Furthermore, this error term is shaped by the product of all previous NTFs, i.e., we retain the same performance as a higher-order system but ensure stability. The performance of a MASH ΔΣ converter is shown in Figure 3.8.

We will generalize the conventional digital-cancellation logic in Section 4.7 as we compare the MASH approach to that of the control-bounded converters.

Chapter 4

Control-Bounded Analog-to-Digital Conversion

The control-bounded analog-to-digital converter (ADC) approaches the A/D conversion task differently compared to conventional A/D converters from Chapter 3. Specifically, the A/D conversion task is broken down into an analog system (AS), a digital control (DC), and a digital estimator (DE) step, as illustrated in Figure 4.1. Before going into detail and motivating the overall ADC structure, we briefly summarize each step and describe their respective objective.

- The AS (preconditioning filter) is constructed such that it greatly amplifies, possibly in an unstable way, the sought signal characteristics of $\boldsymbol{u}(t)$.
- The DC observes a sampled and quantized version of the internal states of the AS and subsequently produces a control signal $\boldsymbol{s}[k] = \big(s_1[k], \ldots, s_M[k]\big)^{\mathsf{T}}$ to counteract the growth of these states. The control signal, which is a digital discrete-time signal, gets enforced by feeding back an analog continuous-time version $\boldsymbol{s}(t)$, denoted the control contribution, to the AS.

- The DE computes a continuous-time representation $\hat{\boldsymbol{u}}(t)$ by solving the inverse problem $(\boldsymbol{g} * \hat{\boldsymbol{u}})(t) = (\boldsymbol{g} * \boldsymbol{s})(t)$ where $\boldsymbol{g}(t)$ is the impulse response of the AS. Additionally, the representation is such that we can sample from it at arbitrary times t. These samples constitute the output of the control-bounded ADC.

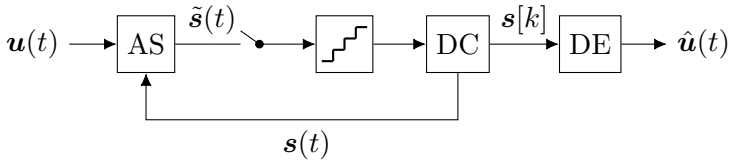


Figure 4.1: The control-bounded view on A/D conversion.

In general, each of the signals $\boldsymbol{u}(t)$, $\tilde{\boldsymbol{s}}(t)$, $\boldsymbol{s}[k]$, and $\boldsymbol{s}(t)$ are vector-valued.

In the special case when $\boldsymbol{u}(t)$, $\tilde{\boldsymbol{s}}(t)$, and $\boldsymbol{s}(t)$ are scalar-valued functions, the structure in Figure 4.1 does partially resemble that of a continuous-time $\Delta\Sigma$ converter from Figure 3.3. Indeed, replacing the decimation filter with the DE and recognizing the DAC as part of the DC, results in a structure similar to that of a control-bounded ADC. This comparison also reveals a fundamental conceptual difference: the interpretation of the intermediate quantity $\boldsymbol{s}[k]$. In the conventional view this is a sampled and quantized version of input signal $\boldsymbol{u}(t)$ seen at the output of the $\Delta\Sigma$ modulator, whereas in the control-bounded perspective this is a control signal that stabilized the system and is therefore only indirectly related to the input signal. These two perspectives result in two different ways of estimating $\hat{\boldsymbol{u}}(t)$.

However, the main contribution of the control-bounded ADC concept is not an alternative decimation filter to the current state-of-the-art continuous-time $\Delta\Sigma$ modulators. Instead, the control-bounded conversion perspective defines a new interface between analog and digital domain that enables ASs and DCs combinations which were previously unimaginable. We will come back and elaborate on this new design paradigm in Section 4.5. To fully grasp the design aspects, we must first develop the necessary mathematical foundation to describe the functionality and interactions of each of the three control-bounded A/D conversion building blocks. This will be done in the succeeding three sections (Section 4.1, Section 4.2, and Section 4.3).

The goal of this chapter is to describe the fundamental aspects of a generalized control-bounded ADC structure. Several examples of control-bounded ADCs will follow in the Chapters 5, 6, 7, 8, and 10.

4.1 Analog System

The preconditioning filter, from here on referred to as the **the analog system (AS)**, is an analog continuous-time filter. The role of the AS is to enhance specific signal attributes of $\boldsymbol{u}(t)$ while simultaneously suppressing unwanted attributes. Typically, this means amplifying one or multiple frequency bands of $\boldsymbol{u}(t)$ while suppressing others. In general, larger amplification and sharp transitions between the passband and stopband require more complex, higher-order ASs. The overall performance of a control-bounded converter will be inherently linked to the AS's ability to amplify the wanted signal characteristics, as we will return to in Section 4.5.

Furthermore, since the AS will be controlled via the DC, it can be thought of as an open-loop system. This means that stability is not necessary for the AS to operate as stability will be enforced via the DC loop.

4.1.1 State Space Model

To describe the dynamics of the AS, we use a state space model notation, illustrated in Figure 4.2. Specifically, the relation between a multi-channel input signal

$$\boldsymbol{u}(t) \triangleq \big(u_1(t), \ldots, u_L(t)\big)^{\mathsf{T}} \in \mathbb{R}^L, \tag{4.1}$$

the AS state vector

$$\boldsymbol{x}(t) \triangleq \big(x_1(t), \ldots, x_N(t)\big)^{\mathsf{T}} \in \mathbb{R}^N, \tag{4.2}$$

and the control contribution

$$\boldsymbol{s}(t) \triangleq \big(s_1(t), \ldots, s_M(t)\big)^{\mathsf{T}} \in \mathbb{R}^M \tag{4.3}$$

are given by the system of ordinary differential equations (ODEs)

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{\Gamma}\boldsymbol{s}(t). \tag{4.4}$$

We say that such a system has L inputs, M controls, and N states. Furthermore, the system matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, input matrix $\boldsymbol{B} \in \mathbb{R}^{N\times L}$, and control input matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{N\times M}$ are all real-valued matrices.

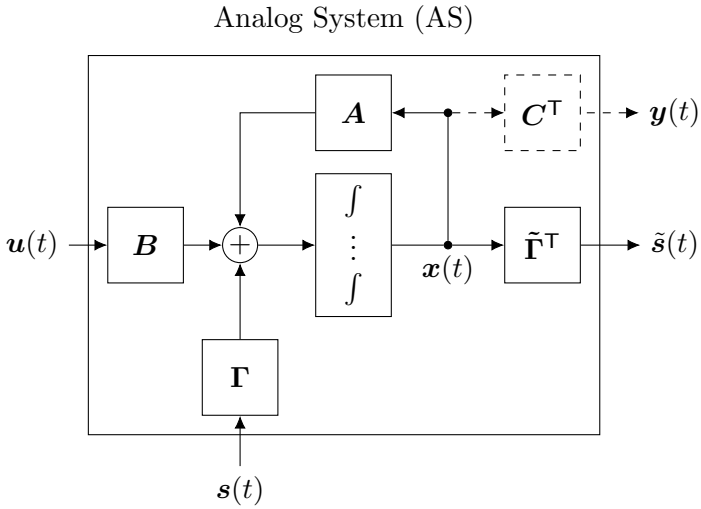


Figure 4.2: State space model of the AS.

The AS additionally has two outputs: the control observation $\tilde{\boldsymbol{s}}(t)$ and the signal observation $\boldsymbol{y}(t)$. The former is an actual physical signal that the DC uses when determining the control signal. The latter is a conceptual signal used by the DE when forming the estimate $\hat{\boldsymbol{u}}(t)$. Specifically, the DC observes the control observation $\tilde{\boldsymbol{s}}(t)$, which is a linear mapping of the state vector via the control observation matrix $\tilde{\boldsymbol{\Gamma}}^\mathsf{T} \in \mathbb{R}^{\tilde{M}\times N}$ as

$$\tilde{\boldsymbol{s}}(t) \triangleq \tilde{\boldsymbol{\Gamma}}^\mathsf{T}\boldsymbol{x}(t) \in \mathbb{R}^{\tilde{M}}. \tag{4.5}$$

Similarly, the DE uses $\tilde{N}$ state linear mappings as

$$\boldsymbol{y}(t) \triangleq \boldsymbol{C}^\mathsf{T}\boldsymbol{x}(t) \tag{4.6}$$

where $\boldsymbol{C}^\mathsf{T} \in \mathbb{R}^{\tilde{N}\times N}$ is the signal observation matrix. Note that since both $\boldsymbol{y}(t)$ and $\boldsymbol{C}$ are purely conceptual quantities, they have no part in the actual physical design of the AS. This is indicated in Figure 4.2 by the dashed markings.

Furthermore, the number of controls M and the number of control observations $\tilde{M}$ are not required to be the same. Also, note that we have assumed the control contribution $\boldsymbol{s}(t)$ to enter the AS in an additive way.

4.1.2 Transfer Function & Impulse Response Matrix

The given system of ODEs result in a transfer function matrix

$$\boldsymbol{G}(\omega) = \boldsymbol{C}^{\mathsf{T}} \left(i\omega \boldsymbol{I}_N - \boldsymbol{A}\right)^{-1} \boldsymbol{B} \in \mathbb{C}^{\tilde{N} \times L} \tag{4.7}$$

and impulse response matrix

$$\boldsymbol{g}(t) = \boldsymbol{C}^{\mathsf{T}} \exp\left(\boldsymbol{A}t\right) \boldsymbol{B} \in \mathbb{R}^{\tilde{N} \times L}, \tag{4.8}$$

both describing the relation from $\boldsymbol{u}(t)$ to $\boldsymbol{y}(t)$, where (4.7) is denoted the analog transfer function (ATF) matrix and similarly (4.8) as the analog impulse response matrix. In the expression above $\exp(\cdot)$ denotes the matrix exponential function.

4.1.3 Anti-Aliasing Filter

The control-bounded converter does not require an explicit anti-aliasing filter to precede it. Furthermore, as we do not consider DC signals to be samples, aliasing effects do not directly affect the control-bounded conversion process, as is the case for conventional converters. Specifically, in the control-bounded converter, aliasing effects manifest themselves as we sample the control contributions $\tilde{\boldsymbol{s}}(t)$. It is the AS's task to suppress the out-of-band signals possibly originating from either the input signal or the control contributions. In this view, the AS also functions as an anti-aliasing filter for the control observations. However, as suppressing out-of-band signals while amplifying the in-band signals is part of the fundamental design criteria of AS; additional anti-aliasing related circuitry is typically not necessary.

4.2 Digital Control

In contrast to the AS, the DC operates in a discrete-time setting where every operation is synchronized with a global clock, having a clock period denoted T. We will often refer to this clock period as the control period T. Furthermore, with the exception of the control contribution, see Section 4.2.1, all internal variables are discrete-time digital signals.

It is the task of the DC to maintain bounded AS states. To this end, the DC observes a sampled and quantized version of the control observations (4.5) and produces a control contribution in response. In other words, the DC internally includes both conventional ADCs, to observe the

control observation $\tilde{\boldsymbol{s}}(t)$, and DACs to produce the control contribution $\boldsymbol{s}(t)$, which is a continuous-time analog signal. In all the examples of this thesis, we only consider 1-bit quantizers. However, it is possible, but not recommended, to use higher-order quantization as discussed in Section 4.2.3. Clearly, for any properly designed control-bounded converter the effective resolution of the estimated samples $\hat{\boldsymbol{u}}(t)$, generated by the DE, greatly exceed those of the internal quatizers of the DC.

As previously stated, the task of the DC is to counteract the growth of the AS states caused by the input signal and previous control contributions. Its success is determined by the magnitude of the elements of the resulting state vector $\boldsymbol{x}(t)$. A DC that can maintain a bounded AS state for a bounded input signal is called effective. Effective controls will be the topic of Section 4.2.2.

4.2.1 Control Contribution

The control contribution $\boldsymbol{s}(t)$ was introduced as an analog continuous-time version of the control signal $\boldsymbol{s}[k]$. Specifically, the relation is determined by the DC's corresponding DAC waveforms [22] as

$$\boldsymbol{s}(t) = \boldsymbol{D}(t - kT)\boldsymbol{s}[k] \tag{4.9}$$

where $\boldsymbol{D}(t)$ is a diagonal matrix as

$$\boldsymbol{D}(t) \triangleq \operatorname{diag}\Big(\big(d_1(t), \ldots, d_M(t)\big)^{\mathsf{T}}\Big) \tag{4.10}$$

Here $d_\ell(t)$ is the continuous-time DAC waveform associated with control signal's ℓ-th element $s_\ell[k]$.

In this work, we will mostly use the square DAC waveform, i.e.,

$$d_\ell(t) \triangleq \begin{cases} 1 & \text{if } t \in [0, T) \\ 0 & \text{otherwise.} \end{cases} \tag{4.11}$$

However, it is straightforward to extend the control-bounded ADC for other DAC waveforms. This is the case for the switched capacitor DC in Section 5.3.2 and the chain-of-oscillators DC in Section 7.4.

4.2.2 Effective Control

The performance of a control-bounded converter is inherently linked to the ability of bounding the AS state vector, or equivalently, the conversion

error signal seen at the AS's fictional output, i.e., $-\boldsymbol{y}(t)$, as will be covered in Section 4.3.1. This is done by the DC which observes a sampled and quantized version of the control observation (4.5), and produces a control contribution $\boldsymbol{s}(t)$ in response.

Prior knowledge of the input signal set can be incorporated into the DC design. As an example, the input signals might be bandlimited, or their amplitude are bounded by a known value. Throughout this thesis, all input signals are assumed bounded, i.e.

$$\boldsymbol{u}(t) \in \mathcal{U} \triangleq \{\boldsymbol{v}(t) : \|\boldsymbol{v}(t)\|_\infty \leq b_{\boldsymbol{u}}, \forall t\} \tag{4.12}$$

for a $b_{\boldsymbol{u}} > 0$. Furthermore, we call a DC effective if it guarantees bounded state vectors, i.e.

$$\boldsymbol{x}(t) \in \mathcal{X} \triangleq \{\tilde{\boldsymbol{x}}(t) : \|\tilde{\boldsymbol{x}}(t)\|_\infty \leq b_{\boldsymbol{x}}, \forall t\} \tag{4.13}$$

for a $b_{\boldsymbol{x}} > 0$ and a bounded input signal.

Note that, in general, $\Delta\Sigma$ modulators as those in Section 3.3, have no stability guarantees for larger order filters and bounded input signals. By stability, we mean that the magnitude of one or multiple elements of the state vector does not exceed some bound resulting from physical limits of the AS, which in turn leads to that the $\Delta\Sigma$ modulators hang and or requires a reset during operation.

We can also formulate an effective control using the dynamical system from (4.4). Specifically, by evalutating the solution to the system of ODEs, the conditions for an effective control can be written as

$$\max_{\boldsymbol{x}(t)\in\mathcal{X},\boldsymbol{u}\in\mathcal{U},t\in[0,T)} \|\tilde{\boldsymbol{g}}(t)\cdot\boldsymbol{x}(0) + (\tilde{\boldsymbol{g}} * \boldsymbol{s})(t) + (\tilde{\boldsymbol{g}} * \boldsymbol{u})(t)\|_\infty \leq b_{\boldsymbol{x}} \tag{4.14}$$

where

$$\tilde{\boldsymbol{g}}(t) \triangleq \exp(\boldsymbol{A}t) \tag{4.15}$$

$$(\tilde{\boldsymbol{g}} * \boldsymbol{s})(t) = \int_0^t \tilde{\boldsymbol{g}}(t-\tau)\boldsymbol{\Gamma}\boldsymbol{s}(\tau)\,\mathrm{d}\tau \tag{4.16}$$

$$(\tilde{\boldsymbol{g}} * \boldsymbol{u})(t) = \int_0^t \tilde{\boldsymbol{g}}(t-\tau)\boldsymbol{B}\boldsymbol{u}(\tau)\,\mathrm{d}\tau. \tag{4.17}$$

Note that $\boldsymbol{s}(t)$ is a function of $\boldsymbol{x}(0)$ as the DC observes a quantized version of the control observation $\tilde{\boldsymbol{s}}(0) = \tilde{\boldsymbol{\Gamma}}\boldsymbol{x}(0)$, and based on this observation,

produces its control signal and control contribution. This comes from the assumption, without loss of generality, of a control period $t \in [0, T)$. A natural consequence of a bounded state vector $\|\boldsymbol{x}(t)\|_\infty \leq b_{\boldsymbol{x}}$ is that the fictional signal observation also will be bounded as

$$\begin{aligned} \|\boldsymbol{C}^\mathsf{T}\boldsymbol{x}(t)\|_\infty &= \|\boldsymbol{y}(t)\|_\infty && (4.18)\\ &\leq b_{\boldsymbol{y}} && (4.19)\\ &= \alpha \cdot b_{\boldsymbol{x}} && (4.20) \end{aligned}$$

for a constant $\alpha > 0$. This is of great importance since the bounded signal observation will be a key component of the DE in Section 4.3.

Typically, for a specific AS paired with a specific DC, (4.14) can be greatly simplified. An example will be given in Section 5.3 called the local DC.

Remainder and Growth Term

The left-hand side of condition (4.14) can be upper bounded, using the triangle inequality, as

$$\begin{aligned} &\max_{\boldsymbol{x}(t)\in\mathcal{X},\boldsymbol{u}\in\mathcal{U},t\in[0,T)} \|\tilde{\boldsymbol{g}}(t)\cdot\boldsymbol{x}(0) + (\tilde{\boldsymbol{g}} * \boldsymbol{s})(t) + (\tilde{\boldsymbol{g}} * \boldsymbol{u})(t)\|_\infty \leq \\ &\underbrace{\max_{\boldsymbol{x}(t)\in\mathcal{X},t\in[0,T)} \|\tilde{\boldsymbol{g}}(t)\cdot\boldsymbol{x}(0) + (\tilde{\boldsymbol{g}} * \boldsymbol{s})(t)\|_\infty}_{\max_{t\in[0,T)} R(t)} + \underbrace{\max_{u\in\mathcal{U},t\in[0,T)} \|(\tilde{\boldsymbol{g}} * u)(t)\|_\infty}_{\max_{t\in[0,T)} G(t)}. \end{aligned} \tag{4.21}$$

From the expression in (4.21) we identify two terms namely the remainder term $R(t)$ and the growth term $G(t)$. The remainder term can be thought of as the difference between the applied control signal and the state trajectory. It is the goal of the DC to make this term as small as possible for all $t \in [0, T)$ but in particular at the end of the control period T.

The growth term represents how much input signal can be fed into the dynamical system during one control period without exceeding the bound. In contrast to the remainder term, the growth term cannot be minimized directly by the DC. Instead, only after the control period T can the DC observe the growth terms effect and start to "digest" the newly introduced signal contribution. As this is an upper bound,

$$\max_{t\in[0,T)} (R(t) + G(t)) \leq b_{\boldsymbol{x}} \tag{4.22}$$

implies that the DC is effective.

4.2.3 Higher-Order Quantizers

In the $\Delta\Sigma$ literature, it is customary to use higher-order quantizers in the feedback loop at the expense of a more complex ADC and DAC in the signal path. This results in better performance and can, to some degree, mitigate artifacts such as limit cycles.

Using higher-order quantizers in the control-bounded converter achieves the same effect since the DC can maintain a smaller state-bound $b_{\boldsymbol{x}}$ for the same class of input signals. However, this is not the preferred way to increase the complexity of the DC. Instead, we advocate the concept of overcomplete control, described in Chapter 9, which achieves the same effect but with additional robustness advantages and low-order quantizers and DACs.

4.2.4 Independent Digital Controls

For the control-bounded ADC examples that will follow, the DC will distribute the control task such that multiple controls operate independently of each other. We call this independent DC. This design choice results in less complicated hardware implementations of the DC since the different control observations do not need to be coordinated. Constructing control signals based on multiple control observations could render superior performance but will not be pursued further in this thesis.

A general control-bounded ADC where the DC uses independent DC is shown in Figure 4.3.

4.3 Digital Estimator

The third part of the control-bounded conversion process is the digital estimation (DE) step. Here, the DE forms an estimate $\hat{\boldsymbol{u}}(t)$ of $\boldsymbol{u}(t)$ based on the control signal $\boldsymbol{s}[k]$, its corresponding control contribution $\boldsymbol{s}(t)$, and the knowledge of the AS system parameters.

4.3.1 Statistical Estimation Problem

The first step of the digital estimation is to forget that the control signal $\boldsymbol{s}[k]$ also represents a sampled and quantized version of some

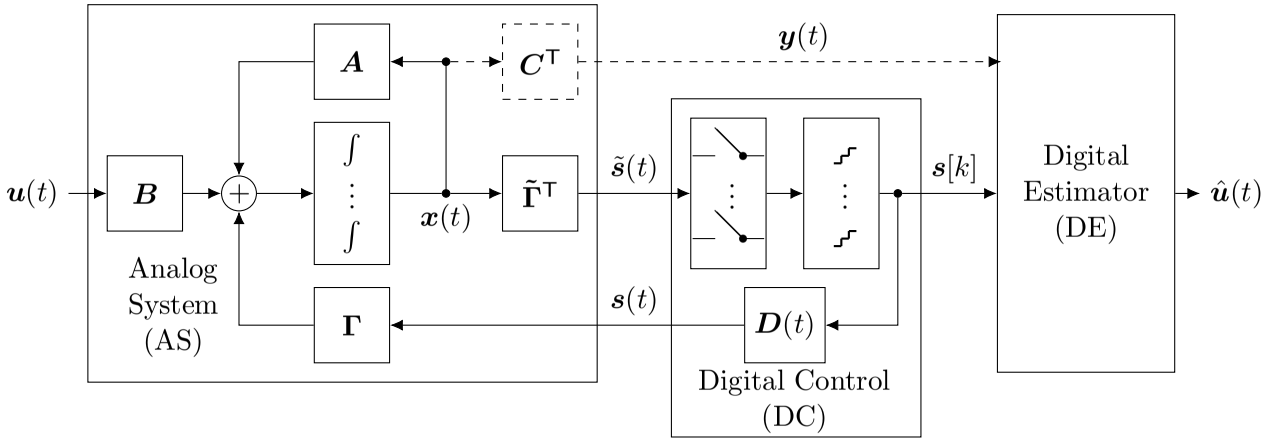


Figure 4.3: A control-bounded ADC using independent DCs. Note that the dashed markings referrers to conceptual quantities that are not part of any hardware design but belongs to the DE and is further explained in Section 4.3.

linear mapping of the state vector. Instead, we focus on the control contribution $\boldsymbol{s}(t) : \mathbb{R} \to \mathbb{R}^M$ that we know results in an effective DC, i.e., its contribution to the state vector $\boldsymbol{x}(t)$ must resemble a mirrored version of the input signal's contribution to the same vector. It follows that the same can be said for the fictional signal observation $\boldsymbol{y}(t)$ as this is simply a linear mapping of the state vector. To formalize this approach imagine the fictional signal

$$\breve{\boldsymbol{y}}(t) \triangleq (\boldsymbol{g} * \boldsymbol{u})(t) \in \mathbb{R}^{\tilde{N}}, \tag{4.23}$$

i.e., the signal observation that would have resulted in the absence of a DC. The actual signal observation can then be written as

$$\boldsymbol{y}(t) = \breve{\boldsymbol{y}}(t) - \boldsymbol{q}(t) \tag{4.24}$$

where $\boldsymbol{q}(t)\colon \mathbb{R} \to \mathbb{R}^{\tilde{N}}$ is the control contribution seen at the signal observation. Note that $\boldsymbol{q}(t)$ is fully determined by the control signal $\boldsymbol{s}[k]$ and is therefore known to the DE.

In contrast to $\boldsymbol{y}(t)$, both $\breve{\boldsymbol{y}}(t)$ and $\boldsymbol{q}(t)$ are not bounded by $b_{\boldsymbol{y}}$. On the contrary, these two quantities magnitudes might, at times, be substantial. In fact, allowing $\|\breve{\boldsymbol{y}}(t)\|_\infty \gg b_{\boldsymbol{y}}$ while $\|\boldsymbol{y}(t)\|_\infty \leq b_{\boldsymbol{y}}$ will be synonymous with small conversion errors, as will explained further below.

For the sake of tractable analysis, we will now assume that the system dynamics (4.4) are invariant and stable. This assumption only applies to the analysis in this section. In particular, the actual digital estimation filter Section 4.3.2 will not be limited by these assumptions.

We formulate an estimate $\hat{\boldsymbol{u}}(t)$ of $\boldsymbol{u}(t)$ of the form

$$\hat{\boldsymbol{u}}(t) = (\boldsymbol{h} * \boldsymbol{q})(t) \in \mathbb{R}^L \tag{4.25}$$

where $\boldsymbol{h}(t)\colon \mathbb{R} \to \mathbb{R}^{L \times \tilde{N}}$ is the impulse response matrix of the estimation filter. Furthermore, by (4.24), it follows that the estimate can be written as

$$\begin{aligned} \hat{\boldsymbol{u}}(t) &= (\boldsymbol{h} * \breve{\boldsymbol{y}})(t) - (\boldsymbol{h} * \boldsymbol{y})(t) && (4.26)\\ &\approx (\boldsymbol{h} * \breve{\boldsymbol{y}})(t) && (4.27)\\ &= (\boldsymbol{h} * \boldsymbol{g} * \boldsymbol{u})(t) && (4.28) \end{aligned}$$

where the quality of the approximation in (4.27) relies on that the elements of $\breve{\boldsymbol{y}}(t)$ have a much larger magnitude than those of $\boldsymbol{y}(t)$. Furthermore,

we recognize $-\boldsymbol{y}(t)$ as the conversion error signal seen at the signal observation output of the AS. The whole estimation process is summarized in Figure 4.4.

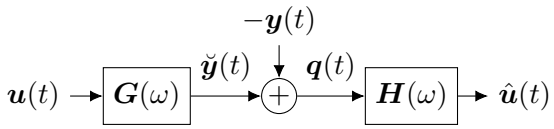


Figure 4.4: The estimation problem of a control-bounded converter, where $\boldsymbol{q}(t)$ is known to the DE.

It remains to determine $\boldsymbol{h}(t)$. To this end, we construct a statistical estimation problem where $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$ are both assumed to be independent, multivariate, centered, and wide-sense stationary stochastic processes. Furthermore, the objective is to find the filter with impulse response matrix $\boldsymbol{h}(t)$ such that

$$\boldsymbol{h}(t) = \underset{\bar{\boldsymbol{h}}(t)}{\operatorname{argmin}} \, \mathrm{E}\big[(\hat{\boldsymbol{u}}(t) - \boldsymbol{u}(t))^2\big] \tag{4.29}$$

$$= \underset{\bar{\boldsymbol{h}}(t)}{\operatorname{argmin}} \, \mathrm{E}\big[((\bar{\boldsymbol{h}} * \boldsymbol{q})(t) - \boldsymbol{u}(t))^2\big] \,. \tag{4.30}$$

Note that t has no apparent meaning in (4.30) since both $\boldsymbol{q}(t)$ and $\boldsymbol{u}(t)$ are assumed weakly stationary. In fact, we know $\boldsymbol{q}(t)$ to be cyclostationary with period T. However, assuming it to be weakly stationary turns out to give satisfactory results. The optimization problem in (4.30) has an analytical solution which, via the orthogonality principle, is determined by the conditions

$$\mathrm{E}\big[((\boldsymbol{h} * \boldsymbol{q})(t) - \boldsymbol{u}(t)) \, \boldsymbol{q}(t+\tau)^{\mathsf{T}}\big] = \boldsymbol{0}_{L \times \tilde{N}} \tag{4.31}$$

for any $\tau \in \mathbb{R}$. (4.31) can also be written as

$$\mathrm{E}\big[(\boldsymbol{h} * \boldsymbol{q})(t)\boldsymbol{q}(t-\tau)^{\mathsf{T}}\big] = \mathrm{E}\big[\boldsymbol{u}(t)\boldsymbol{q}(t-\tau)^{\mathsf{T}}\big] \tag{4.32}$$

$$(\boldsymbol{h} * \boldsymbol{R}_{\boldsymbol{q}\boldsymbol{q}^{\mathsf{T}}})(\tau) = \boldsymbol{R}_{\boldsymbol{u}\boldsymbol{q}^{\mathsf{T}}}(-\tau) \tag{4.33}$$

where

$$\boldsymbol{R}_{\boldsymbol{q}\boldsymbol{q}^{\mathsf{T}}}(\tau) \triangleq \mathrm{E}\big[\boldsymbol{q}(t)\boldsymbol{q}(t+\tau)^{\mathsf{T}}\big] \tag{4.34}$$

$$\boldsymbol{R}_{\boldsymbol{u}\boldsymbol{q}^{\mathsf{T}}}(\tau) \triangleq \mathrm{E}\big[\boldsymbol{u}(t)\boldsymbol{q}(t+\tau)^{\mathsf{T}}\big] \tag{4.35}$$

are the autocovariance and cross-covariance respectively. The Equation (4.33) is commonly known as the Wiener-Hopf equation.

By taking the Fourier transform on both sides of (4.33) we obtain

$$\boldsymbol{H}(\omega)\left(\boldsymbol{G}(\omega)\boldsymbol{S}_{\boldsymbol{u}\boldsymbol{u}^\intercal}(\omega)\boldsymbol{G}(\omega)^{\mathsf{H}}+\boldsymbol{S}_{\boldsymbol{y}\boldsymbol{y}^\intercal}(\omega)\right)=\boldsymbol{S}_{\boldsymbol{u}\boldsymbol{u}^\intercal}(\omega)\boldsymbol{G}(\omega)^{\mathsf{H}} \quad (4.36)$$

where $\boldsymbol{H}(\omega)$ is the element-wise Fourier transform of $\boldsymbol{h}(t)$ and the PSDs

$$\boldsymbol{S}_{\boldsymbol{u}\boldsymbol{u}^\intercal}(\omega)\triangleq\int_{-\infty}^{\infty}\mathrm{E}\big[\boldsymbol{u}(t)\boldsymbol{u}(t+\tau)^{\mathsf{T}}\big]\,e^{-i\omega\tau}\,\mathrm{d}\tau \quad (4.37)$$

$$\boldsymbol{S}_{\boldsymbol{y}\boldsymbol{y}^\intercal}(\omega)\triangleq\int_{-\infty}^{\infty}\mathrm{E}\big[\boldsymbol{y}(t)\boldsymbol{y}(t+\tau)^{\mathsf{T}}\big]\,e^{-i\omega\tau}\,\mathrm{d}\tau. \quad (4.38)$$

In the case both $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$ are assumed i.i.d. multivariate Gaussian stochastic processes, their spectral densities can be written as

$$\boldsymbol{S}_{\boldsymbol{u}\boldsymbol{u}^\intercal}(\omega)=\sigma_{\boldsymbol{u}}^2\boldsymbol{I}_L \quad (4.39)$$

$$\boldsymbol{S}_{\boldsymbol{y}\boldsymbol{y}^\intercal}(\omega)=\sigma_{\boldsymbol{y}}^2\boldsymbol{I}_N. \quad (4.40)$$

Consequently, the condition from (4.36) can be rearranged such that the reconstruction filter follows as

$$\boldsymbol{H}(\omega)=\boldsymbol{G}(\omega)^{\mathsf{H}}\left(\boldsymbol{G}(\omega)\boldsymbol{G}(\omega)^{\mathsf{H}}+\eta^2\boldsymbol{I}_N\right)^{-1} \quad (4.41)$$

where $\eta^2\triangleq\sigma_{\boldsymbol{y}}^2/\sigma_{\boldsymbol{u}}^2$. (4.41) is also known as a Wiener filter [1, 13]. The steps from (4.32) to (4.33) and in particular (4.36) are covered in detail in Appendix A.

Revisiting the approximation in (4.27) we can write the conversion error signal as

$$\boldsymbol{\epsilon}(t)\triangleq\hat{\boldsymbol{u}}(t)-(\boldsymbol{h}*\boldsymbol{g}*\boldsymbol{u})(t) \quad (4.42)$$

$$=-(\boldsymbol{h}*\boldsymbol{y})(t). \quad (4.43)$$

From (4.43) it is clear that the error is "shaped" by $\boldsymbol{H}(\omega)$. With this in mind, we will refer to the reconstruction filter $\boldsymbol{H}(\omega)$ as the noise transfer function (NTF) matrix. By rewriting the estimate from (4.26) in the Fourier domain as

$$\hat{\boldsymbol{U}}(\omega)=\underbrace{\boldsymbol{H}(\omega)\boldsymbol{G}(\omega)}_{\mathrm{STF}(\omega)}U(\omega)-\underbrace{\boldsymbol{H}(\omega)}_{\mathrm{NTF}(\omega)}\boldsymbol{Y}(\omega) \quad (4.44)$$

we can also identify the signal transfer function (STF) matrix as

$$\boldsymbol{T}(\omega) \triangleq \mathrm{STF}(\omega) \tag{4.45}$$
$$= \boldsymbol{H}(\omega)\boldsymbol{G}(\omega). \tag{4.46}$$

The Scalar Input Case

In the scalar input signal case the ATF matrix (4.7) is a column vector. Therefore, the matrix inverse in (4.41) can be reduced to a scalar division using the matrix inversion lemma. Subsequently, the NTF and STF from (4.44) can be written in a simplified form as

$$\boldsymbol{H}(\omega) = \mathrm{NTF}(\omega) = \frac{\boldsymbol{G}(\omega)^{\mathsf{H}}}{\|\boldsymbol{G}(\omega)\|_2^2 + \eta^2} \in \mathbb{C}^{1\times\tilde{N}} \tag{4.47}$$

and

$$\mathrm{STF}(\omega) = \frac{\|\boldsymbol{G}(\omega)\|_2^2}{\|\boldsymbol{G}(\omega)\|_2^2 + \eta^2} \in \mathbb{R}. \tag{4.48}$$

Setting the Bandwidth

The bandwidth of the DE filter is regulated using the parameter $\eta > 0$. In the scalar input case, assuming a monotonically decreasing $\|\boldsymbol{G}(\omega)\|_\infty$ in ω, the bandwidth of the DE can be determined as

$$\|\boldsymbol{G}(\omega_{\mathrm{crit}})\|_2^2 = \eta^2 \tag{4.49}$$

where the frequency band of interest is determined by $0 \leq |\omega| \leq \omega_{\mathrm{crit}}$

Additionally, at the critical frequency the ratio of the STF and NTF

$$\frac{\mathrm{STF}(\omega_{\mathrm{crit}})}{\|\boldsymbol{H}(\omega_{\mathrm{crit}})\|_2} = \frac{\|\boldsymbol{G}(\omega_{\mathrm{crit}})\|_2^2}{\|\boldsymbol{G}(\omega_{\mathrm{crit}})\|_2^2 + \eta^2} \left(\frac{\|\boldsymbol{G}(\omega_{\mathrm{crit}})\|_2}{\|\boldsymbol{G}(\omega_{\mathrm{crit}})\|_2^2 + \eta^2}\right)^{-1} \tag{4.50}$$
$$= \|\boldsymbol{G}(\omega_{\mathrm{crit}})\|_2 \tag{4.51}$$
$$= \eta. \tag{4.52}$$

4.3.2 Digital Estimation Filter

The estimate in (4.25) is not straightforward since it involves continuous-time convolution and possibly unbounded quantities. Fortunately, there is

a non-standard Kalman smoothing algorithm [19] that, when computing samples of $\hat{\boldsymbol{u}}(t)$, converges to the estimate in (4.25) as the considered time window extends towards infinity. Furthermore, the algorithm is indifferent to the stable AS assumption that was made in the analysis in the previous section.

The algorithm is derived using factor graphs [4, 5, 18]. However, in-depth knowledge of factor graphs is not necessary for applying the algorithm as it reduces to a linear filter. Therefore, we will proceed by presenting the resulting filter. A derivation of this algorithm can be found in Appendix D.1 or alternatively in [20].

In principle, this algorithm can be adjusted for any set of samples $\{t_1, t_2, \dots\}$. However, the expressions below are computed for regular sampling grid, i.e., uniformly spaced samples $\{\dots, (k-1)T, kT, \dots\}$ where T is the control period from the DC.

The algorithm reduces to three steps: firstly, a forward recursion

$$\overrightarrow{\boldsymbol{m}}_{k+1} \triangleq \boldsymbol{A}_f \overrightarrow{\boldsymbol{m}}_k + \boldsymbol{B}_f \boldsymbol{s}[k], \tag{4.53}$$

a backward recursion

$$\overleftarrow{\boldsymbol{m}}_{k-1} \triangleq \boldsymbol{A}_b \overleftarrow{\boldsymbol{m}}_k + \boldsymbol{B}_b \boldsymbol{s}[k-1], \tag{4.54}$$

and finally the estimate

$$\hat{\boldsymbol{u}}(t_k) \triangleq \boldsymbol{W}^{\mathsf{T}} \left(\overleftarrow{\boldsymbol{m}}_k - \overrightarrow{\boldsymbol{m}}_k \right). \tag{4.55}$$

An offline and online implementation of these recursions, and their corresponding complexity, will be further discussed in Section 4.3.4 and Section 4.3.5.

To compute the quantities $\boldsymbol{A}_f$, $\boldsymbol{B}_f$, $\boldsymbol{A}_b$, $\boldsymbol{B}_b$, and $\boldsymbol{W}$ we first need to compute the symmetric steady-state forward and backward covariance matrices $\overrightarrow{\boldsymbol{V}} \in \mathbb{R}^{N \times N}$ and $\overleftarrow{\boldsymbol{V}} \in \mathbb{R}^{N \times N}$ which are of the same dimensions as $\boldsymbol{A}$. The forward steady-state covariance matrix can be computed by finding the limit

$$\overrightarrow{\boldsymbol{V}} \triangleq \lim_{\tau \to 0} \lim_{\ell \to \infty} \overrightarrow{\boldsymbol{V}}_\ell \tag{4.56}$$

where

$$\overrightarrow{\boldsymbol{V}}_{\ell+1} \triangleq \overrightarrow{\boldsymbol{V}}_\ell + \tau \left(\boldsymbol{A} \overrightarrow{\boldsymbol{V}}_\ell + \left(\boldsymbol{A} \overrightarrow{\boldsymbol{V}}_\ell \right)^{\mathsf{T}} + \boldsymbol{B}\boldsymbol{B}^{\mathsf{T}} - \frac{1}{\eta^2} \overrightarrow{\boldsymbol{V}}_\ell \boldsymbol{C}\boldsymbol{C}^{\mathsf{T}} \overrightarrow{\boldsymbol{V}}_\ell \right) \tag{4.57}$$

or equivalently by solving the continuous-time algebraic Riccati (CARE) equation

$$\boldsymbol{A}\overrightarrow{\boldsymbol{V}}_\ell + \left(\boldsymbol{A}\overrightarrow{\boldsymbol{V}}_\ell\right)^\mathsf{T} + \boldsymbol{B}\boldsymbol{B}^\mathsf{T} - \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}_\ell \boldsymbol{C}\boldsymbol{C}^\mathsf{T}\overrightarrow{\boldsymbol{V}}_\ell = \boldsymbol{0}_{N\times N}. \tag{4.58}$$

The backward steady state covariance matrix is computed almost identically but with a sign change resulting in the continuous-time algebraic Riccati equation

$$\boldsymbol{A}\overleftarrow{\boldsymbol{V}}_\ell + \left(\boldsymbol{A}\overleftarrow{\boldsymbol{V}}_\ell\right)^\mathsf{T} - \boldsymbol{B}\boldsymbol{B}^\mathsf{T} + \frac{1}{\eta^2}\overleftarrow{\boldsymbol{V}}_\ell \boldsymbol{C}\boldsymbol{C}^\mathsf{T}\overleftarrow{\boldsymbol{V}}_\ell = \boldsymbol{0}_{N\times N}. \tag{4.59}$$

The matrix $\boldsymbol{W} \in \mathbb{R}^{L\times N}$ follows from solving the linear equation system

$$\left(\overrightarrow{\boldsymbol{V}} + \overleftarrow{\boldsymbol{V}}\right)\boldsymbol{W} = \boldsymbol{B} \tag{4.60}$$

with respect to $\boldsymbol{W}$. The $\boldsymbol{A}_f \in \mathbb{R}^{N\times N}$ matrix is defined as

$$\boldsymbol{A}_f \triangleq \exp\left(\left(\boldsymbol{A} - \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}\boldsymbol{C}\boldsymbol{C}^\mathsf{T}\right)T\right) \tag{4.61}$$

and similarly the $\boldsymbol{A}_b \in \mathbb{R}^{N\times N}$ matrix is defined as

$$\boldsymbol{A}_b \triangleq \exp\left(-\left(\boldsymbol{A} + \frac{1}{\eta^2}\overleftarrow{\boldsymbol{V}}\boldsymbol{C}\boldsymbol{C}^\mathsf{T}\right)T\right). \tag{4.62}$$

Furthermore, the $\boldsymbol{B}_f \in \mathbb{R}^{N\times M}$ matrix is defined as

$$\boldsymbol{B}_f \triangleq \int_0^T \exp\left(\left(\boldsymbol{A} - \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}\boldsymbol{C}\boldsymbol{C}^\mathsf{T}\right)(T-\tau)\right)\boldsymbol{\Gamma}\boldsymbol{D}(\tau)\,\mathrm{d}\tau \tag{4.63}$$

and the $\boldsymbol{B}_b \in \mathbb{R}^{N\times N}$ matrix is defined as

$$\boldsymbol{B}_b \triangleq -\int_0^T \exp\left(-\left(\boldsymbol{A} + \frac{1}{\eta^2}\overleftarrow{\boldsymbol{V}}\boldsymbol{C}\boldsymbol{C}^\mathsf{T}\right)(T-\tau)\right)\boldsymbol{\Gamma}\boldsymbol{D}(T-\tau)\,\mathrm{d}\tau. \tag{4.64}$$

4.3.3 Parallel Digital Estimation Filter

Equations (4.53), (4.54) and (4.55) can also be casted as a fully parallel version where

$$\overrightarrow{\tilde{m}}_{k+1,n} \triangleq \overrightarrow{\tilde{\lambda}}_n \overrightarrow{\tilde{m}}_{k,n} + \overrightarrow{\tilde{f}}_n(\boldsymbol{s}[k]) \tag{4.65}$$

$$\overleftarrow{\tilde{m}}_{k-1,n} \triangleq \overleftarrow{\tilde{\lambda}}_n \overleftarrow{\tilde{m}}_{k,n} + \overleftarrow{\tilde{f}}_n(\boldsymbol{s}[k-1]) \tag{4.66}$$

and

$$\hat{u}_\ell[k] = \sum_{n=1}^{N} \overrightarrow{w}_{n,\ell} \overrightarrow{m}_{k,n} + \overleftarrow{w}_{n,\ell} \overleftarrow{m}_{k,n} \tag{4.67}$$

Note that (4.65), (4.66) and (4.67) are all scalar expressions. The index n in (4.65)–(4.67) and the index ℓ in (4.67) refer to the respective vector components. The coefficients $\overrightarrow{\lambda}_n$ and $\overleftarrow{\lambda}_n$ are obtained from the eigenvalue decomposition

$$\boldsymbol{A_f} = \boldsymbol{Q_f} \overrightarrow{\boldsymbol{\Lambda}} \boldsymbol{Q_f}^{-1} \tag{4.68}$$

$$\boldsymbol{A_b} = \boldsymbol{Q_b} \overleftarrow{\boldsymbol{\Lambda}} \boldsymbol{Q_b}^{-1} \tag{4.69}$$

where $\overrightarrow{\boldsymbol{\Lambda}} = \operatorname{diag}(\overrightarrow{\lambda}_1, \ldots, \overrightarrow{\lambda}_N)$ and $\overleftarrow{\boldsymbol{\Lambda}} = \operatorname{diag}(\overleftarrow{\lambda}_1, \ldots, \overleftarrow{\lambda}_N)$ are the eigenvalues of $\boldsymbol{A_f}$ and $\boldsymbol{A_b}$ respectively. The scalar functions $\overrightarrow{f}_n(\cdot)$ and $\overleftarrow{f}_n(\cdot)$ are the n-th elements of the vectorized functions

$$\overrightarrow{\boldsymbol{f}}(\boldsymbol{s}[k]) \triangleq \boldsymbol{Q_f}^{-1} \boldsymbol{B_f} \boldsymbol{s}[k] \tag{4.70}$$

$$\overleftarrow{\boldsymbol{f}}(\boldsymbol{s}[k]) \triangleq \boldsymbol{Q_b}^{-1} \boldsymbol{B_b} \boldsymbol{s}[k], \tag{4.71}$$

and $\overrightarrow{w}_{n,\ell}$ and $\overleftarrow{w}_{n,\ell}$ are the (n,ℓ)-th elements of the matrices

$$\overrightarrow{\boldsymbol{W}} \triangleq -\boldsymbol{Q_f}^{\mathsf{T}} \boldsymbol{W} \tag{4.72}$$

$$\overleftarrow{\boldsymbol{W}} \triangleq \boldsymbol{Q_b}^{\mathsf{T}} \boldsymbol{W}. \tag{4.73}$$

4.3.4 Offline Batch Estimation

The DE filter, as specified by (4.53), (4.54), and (4.55), result in an offline version of the DE filter. Specifically, we can estimate the input signal $\hat{\boldsymbol{u}}(kT)$ for a batch of samples $k \in [K_0, K_0 + K_1 - 1]$ where $K_1, K_0 \in \mathbb{Z}^+$. Computing the batch requires the control signals $\{\boldsymbol{s}[K_0], \ldots, \boldsymbol{s}[K_0 + K_1 - 1]\}$ as well as the precomputed filter coefficients from (4.61), (4.62), (4.63), (4.64), and (4.60).

A pseudo code implementation of the described recursions are given in Algorithm 2 that is found in Appendix E.1. In summary, the algorithm first does a forward recursion followed by a backward recursion and estimation step.

Similarly, the parallel recursions from (4.65), (4.66) and (4.67) result in an offline algorithm as given in Algorithm 3 that is found in Appendix E.1.2. In the pseudo code we have highlighted the possibility of executing code in parallel by the *do in parallel* statement. The main difference between Algorithm 2 and Algorithm 3 is the number of required multiplications scale linearly in the latter compared to quadratic in the former. Note that this has nothing to do with previously mentioned parallelism. The linear scaling of number of multiplications is of great value from a computational effort viewpoint but comes with an additional caveat. Specifically, the parallel version requires complex arithmetics as the eigenvalue decomposition, from (4.68) and (4.69), typically results in complex eigenvalues and eigenvectors.

Computational Complexity for Offline Version

Before discussing the computational complexity of these offline batch algorithms we remind ourselves that, as also given in Table 4.1,

- K_1 is the number of samples in the batch,
- L is the number of input channels,
- M is the number of independent scalar controls used by the DCs,
- and N is the number of states in our AS.

The computational effort of Algorithm 2 and Algorithm 3 are summarized in Table 4.2 and Table 4.3 respectively where we have assumed a naive implementation of the matrix vector operations. Specifically, these results follow from the general notion that a $L \times N$ matrix and N vector product requires LN multiplications and $L(N-1)$ additions. Furthermore, assuming a binary control signal $\boldsymbol{s}[k]$ the product $\boldsymbol{B}_f\boldsymbol{s}[k]$, $\boldsymbol{B}_b\boldsymbol{s}[k]$, $\boldsymbol{Q}_{\boldsymbol{f}}^{-1}\boldsymbol{B}_{\boldsymbol{f}}\boldsymbol{s}[k]$, and $\boldsymbol{Q}_{\boldsymbol{b}}^{-1}\boldsymbol{B}_{\boldsymbol{b}}\boldsymbol{s}[k]$ each reduces to $M(N-1)$ additions.

Alternatively, due to the digital nature of $\boldsymbol{s}[k]$, these computations could be precomputed and implemented using a lookup table. Specifically, such a lookup table would map 2^M unique control signal sample combinations to twice as many N-dimensional scalar vectors. This approach would dramatically decrease the total number of additions, especially for Algorithm 3, as shown in Table 4.2 and Table 4.3. However, as the memory requirements grows exponentially in M such an approach is only feasible for a relatively low number of independent controls M.

K_1	L	M	N
batch size	#inputs	#DCs	#AS states

Table 4.1: The involved estimator parameters for the offline batch DEs.

	multiplications	additions if $\boldsymbol{s} \in \{+1,-1\}^M$	additions if lookup table
Forward recursion (4.53)	N^2K_1	$(N^2+MN-N)K_1$	N^2K_1
Backward recursion (4.54)	N^2K_1	$(N^2+MN)K_1$	$N(N+1)K_1$
Estimate (4.55)	LNK_1	$L(N-1)K_1$	$L(N-1)K_1$
Total	$(2N^2+LN)K_1$	$(2(N^2+MN)+LN-L-N)K_1$	$(N\,(L+2N+1)-L)\,K_1$

Table 4.2: Computational effort for offline batch DE as in Algorithm 2.

	multiplications	additions if $\boldsymbol{s} \in \{+1,-1\}^M$	additions if lookup table
Forward recursion (4.65)	$4NK_1$	$2MNK_1$	$2NK_1$
Backward recursion (4.66)	$4NK_1$	$2MNK_1$	$2NK_1$
Estimate (4.67)	$8LNK_1$	$2L(2N-1)K_1$	$2L(2N-1)K_1$
Total	$8(L+1)NK_1$	$4\,(MN+L(N-1/2))\,K_1$	$4(L+1)NK_1-2LK_1$

Table 4.3: Computational effort for parallel offline batch DE as in Algorithm 3.

For Algorithm 2 the binary control signal results in $K_1(N(N-1)+N(M-1)+N)$ additions in the forward recursion (row 8 in Algorithm 2), $K_1(N(N-1)+N(M-1)+2N)$ additions in the backward recursion (row 12 in Algorithm 2), and $K_1L(N-1)$ additions in the estimate (row 13 in Algorithm 2).

For Algorithm 3 we account for the complex arithmetics by the notion that a complex multiplication results in 4 real arithmetic multiplications and a complex addition requires 2 real additions. As the computations in Algorithm 3 are all scalar valued, the total number of multiplications and additions follows directly from the expressions.

In terms of memory allocation, the algorithm requires the storage of $\{\boldsymbol{s}[K_0],\dots,\boldsymbol{s}[K_0+K_1-1]\}$ control signals which amounts to K_1M bits. Furthermore, the mean matrix and estimation matrix, from rows 4 and 5 in Algorithm 2, requires storing another $N(K_1+1)+LK_1$ real-valued scalar values. In Algorithm 3, this number is $4N(K_1+1)+2LK_1$ real-valued scalar values. Finally, the filter coefficients amounts to $2N^2+2MN+LN$ scalar values.

In the batch computation above we are not taking the effect of windowing into account. Therefore, to achieve a target performance, the batch must include additional samples on both sides, such that the estimates in the middle of the batch are sufficiently unaffected from windowing effects to reach a target performance criteria.

Computational Effort per Estimated Sample

Based on the results presented in Table 4.2 and Table 4.3 we can express the computational effort per estimated scalar sample, i.e. we divide the batch's computational effort by K_1L, the number of estimated scalar values per batch. Furthermore, we use the big $\mathcal{O}$ notation to describe the general complexity scaling.

The default offline batch estimator from Algorithm 2 in Appendix E consumes

- $\mathcal{O}\left(\frac{N^2}{L}\right)$ real-valued scalar multiplications,
- $\mathcal{O}\left(\frac{N^2+MN}{L}\right)$ real-valued scalar additions,
- and requires $\frac{M}{L}$ bits and $\frac{N}{L}+1$ real-valued scalar values to be kept in memory

per estimated scalar sample. In comparison the parallelized version, as in Algorithm 3, consumes

- $\mathcal{O}(N)$ real-valued scalar multiplications,
- $\mathcal{O}\left(\frac{N(L+M)}{L}\right)$ real-valued scalar additions, or alternatively $\mathcal{O}(N)$ when using a lookup table,
- and requires $\frac{M}{L}$ bits and $4\frac{N}{L}+2$ real-valued scalar values to be kept in memory

per estimated scalar sample.

4.3.5 Online Filter Estimator

The algorithm from the previous section described an offline algorithm for computing a batch of estimates. In this section we will generalize the batch algorithm for the purpose of computing sequences of consecutive batches. In essence, this will involve specifying a time horizons and a batch size to balance fundamental sample delay, computational complexity, and memory allocation.

Fundamentally, this is achieved by adopting a sliding window type filter as illustrated in Figure 4.5. For this approach, computational steps outlined in Section 4.3.4 remains. Additionally, we now considering the effects of windowing by explicitly including a lookahead computation involving K_2 control signal samples that succeed the last estimated sample in the batch. As both the proposed batch estimators are recursive it is only in the, forward time direction that we require a lookahead computation. For looking back in time, the previous filter trajectory can simply be incorporated by passing a single initial mean vector corresponding to the last batch state vector from the preceding batch computation. The online version of Algorithm 2 and Algorithm 3 are given in Algorithm 4 and Algorithm 7 respectively. Further implementation details can be found in Appendix E.2.

The principle difference between these online versions and their corresponding offline versions is the K_2 lookahead computations. In particular, K_2 needs to be chosen large enough such that the effect of windowing is smaller than the sought resolution for the last estimate $\hat{\boldsymbol{u}}[(\ell+1)K_1-1]$ of the ℓ-th batch. Additionally, K_1+K_2 makes up, the worst case, sample delay between a control signal sample and a input estimate sample involved in the same batch.

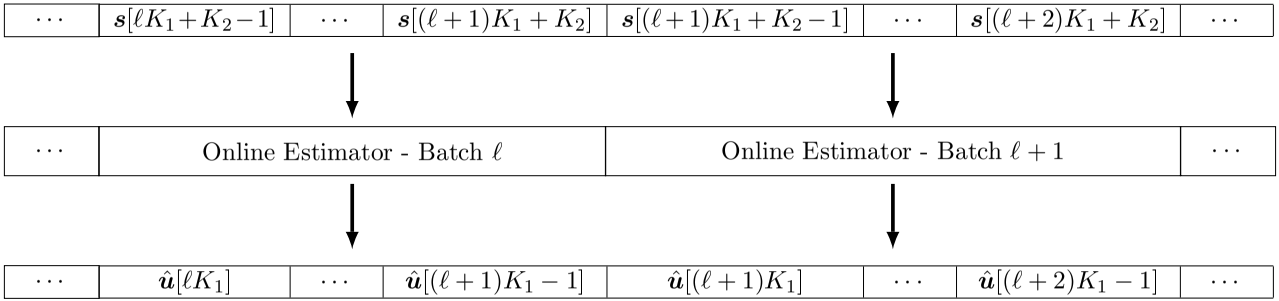


Figure 4.5: The sliding window filter version demonstrating how batches of estimates gets sequentially computed. Note that only K_1 new control signal samples are inserted for each batch. However, each batch computation internally uses $K_1 + K_2$ control signal samples where the additional samples are stored in the preceding batch computation.

Computational Complexity for Online Filter Estimator

As for the offline version we summarize the computational effort of both these online estimators in Table 4.5 and Table 4.6.

When considering the computational effort per scalar estimate Algorithm 4 results in

- $\mathcal{O}\left(\frac{N^2}{L}(1+\frac{K_2}{K_1})\right)$ real-valued scalar multiplications,
- $\mathcal{O}\left(\frac{N^2+MN}{L}(2+\frac{K_2}{K_1})\right)$ real-valued scalar additions,
- and requires $\frac{(1+\frac{K_2+1}{K_1})M}{L}$ bits and $\frac{N+1}{L}+1$ real-valued scalar values to be kept in memory

per estimated scalar sample. Alternatively, the parallel version in Algorithm 7 scales as

- $\mathcal{O}\left(N(1+\frac{K_2}{K_1 L})\right)$ real-valued scalar multiplications,
- $\mathcal{O}\left(N\left(1+\frac{M}{L}(1+\frac{K_2}{K_1})\right)\right)$ real-valued scalar additions,
- and requires $\frac{(1+\frac{K_2+1}{K_1})M}{L}$ bits and $4\frac{N+1}{L}$ real-valued scalar values to be kept in memory

per estimated scalar.

We will next summarize the involved tradeoffs in terms of the DE online version parameters K_1, the estimated samples per batch, and K_2 the lookahead per batch.

Starting with K_2. This parameter scales with the ultimate targeted resolution per sample as a finer resolution requires a longer lookahead to suppress the impact of windowing. This value can be determined by simulation and ideally should be chosen no larger than required.

Secondly, by selecting a large batch size K_1 the relative computational cost associated with the lookahead diminishes as can be seen from the scalings above. The fundamental drawback of a large batch size is that the fundamental sample delay $K_1 + K_2$ increases which might be a severe disadvantage for a real time application. Another aspect with increasing the batch size is that, even though the relative memory allocation per estimated sample does not increase, the total memory requirement grows linearly in K_1.

K_1	K_2	L	M	N
batch size	lookahead	#inputs	#DCs	#AS states

Table 4.4: Description of the involved estimator parameters for the online batch estimators.

	multiplications	additions if $\boldsymbol{s} \in \{+1, -1\}^M$
Lookahead	$N^2 K_2$	$(N^2 + MN - N)K_2$
Batch	$(2N^2 + LN)K_1$	$(2(N^2 + MN) + LN - L)K_1$
Total	$N^2(2K_1 + K_2) + LNK_1$	$(N^2 + MN)(2K_1 + K_2) + (LN - 1)K_1 - NK_2$

Table 4.5: Computational effort for the online estimator, as in Algorithm 4, when computing a single batch of K_1 samples.

	multiplications	additions if $\boldsymbol{s} \in \{+1, -1\}^M$
Lookahead	$4NK_2$	$2MNK_2$
Batch	$8(L+1)NK_1$	$4(-L + N(L+M))K_1$
Total	$8(L+1)NK_1 + 4NK_2$	$4LNK_1 + 2MN(2K_1 + K_2) - 4LK_1$

Table 4.6: Computational effort for parallel online estimator, as in Algorithm 7, when computing a single batch of K_1 samples. Note that the computational complexity of the equivalent offline version is obtained by choosing $K_2 = 0$.

4.3.6 Sub-Sampling

In Section 4.3.2, the digital filter was derived for samples spaced uniformly with the control period T. Another common scenario is to sample less densely. This could, for instance, be the Nyquist rate of the frequency band of interest as would be the case for a decimation filter commonly used in combination with a $\Delta\Sigma$ modulator, see Section 3.2. For a sample period $T_u = \xi T$ where ξ is an integer, we modify (4.53) and (4.54) as

$$\overrightarrow{\boldsymbol{m}}_{k+1} \triangleq \tilde{\boldsymbol{A}}_f \overrightarrow{\boldsymbol{m}}_k + \sum_{\ell=0}^{\xi-1} \tilde{\boldsymbol{B}}_f^{\ell} \boldsymbol{s}[k\xi + \ell], \tag{4.74}$$

$$\overleftarrow{\boldsymbol{m}}_{k-1} \triangleq \tilde{\boldsymbol{A}}_b \overleftarrow{\boldsymbol{m}}_k + \sum_{\ell=0}^{\xi-1} \tilde{\boldsymbol{B}}_b^{\ell} \boldsymbol{s}[k\xi - \ell - 1], \tag{4.75}$$

where $\tilde{\boldsymbol{A}}_f$, $\tilde{\boldsymbol{A}}_b$ are computed as in (4.61) and (4.62) but with T replaced by T_u,

$$\tilde{\boldsymbol{B}}_f^{\ell} \triangleq \int_0^{T_u} \exp\left(\left(\boldsymbol{A} - \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}}\right)(T - \ell T_u - \tau)\right)\boldsymbol{\Gamma}\boldsymbol{D}(\tau)\,\mathrm{d}\tau, \tag{4.76}$$

and

$$\tilde{\boldsymbol{B}}_b^{\ell} \triangleq -\int_0^{T_u} \exp\left(-\left(\boldsymbol{A} + \frac{1}{\eta^2}\overleftarrow{\boldsymbol{V}}\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}}\right)(T - \ell T_u - \tau)\right)\boldsymbol{\Gamma}\boldsymbol{D}(T_u - \tau)\,\mathrm{d}\tau. \tag{4.77}$$

Note that the final estimate computation from (4.55), as well as the computation of the steady-state covariances, remains unchanged.

4.3.7 Digital Estimator as an Impulse Response

The recursions described in (4.53), (4.54), and (4.55) can be organized as a mixed infinite impulse response (IIR) and finite impulse response (FIR) filter or alternatively as a FIR filter. In the following derivations we use the control period T to determine the spacing between samples. It is possible to adjust these expressions for other sample spacings as, for example, as in Section 4.3.6.

For the mixed IIR/FIR filter version this is achieved by writing out the estimate from (4.55) as

$$\hat{\boldsymbol{u}}(t_k) = -\boldsymbol{W}^{\mathsf{T}}\overrightarrow{\boldsymbol{m}}_k + \sum_{\ell=0}^{K-1} \tilde{\boldsymbol{h}}_\ell \boldsymbol{s}[k+\ell] \tag{4.78}$$

with

$$\tilde{\boldsymbol{h}}_\ell \triangleq \boldsymbol{W}^{\mathsf{T}} \boldsymbol{A}_b^\ell \boldsymbol{B}_b \in \mathbb{R}^{L\times M} \tag{4.79}$$

and $\overrightarrow{\boldsymbol{m}}_k$ recursively computed as in (4.53). We recognize $K > 0$ as the lookahead and this needs to be chosen large enough such that the filter's window size is not the limiting precision bottleneck, see Section 4.3.5.

Similarly, the forward recursion can also be expanded resulting in FIR filter version

$$\hat{\boldsymbol{u}}(t_k) = \sum_{\ell=-K_1}^{K_2-1} \breve{\boldsymbol{h}}_\ell \boldsymbol{s}[k+\ell] \tag{4.80}$$

where $\breve{\boldsymbol{h}}_\ell \in \mathbb{R}^{L\times M}$ follows as

$$\breve{\boldsymbol{h}}_\ell \triangleq \begin{cases} \boldsymbol{W}^{\mathsf{T}} \boldsymbol{A}_b^\ell \boldsymbol{B}_b & \text{if } \ell \geq 0 \\ -\boldsymbol{W}^{\mathsf{T}} \boldsymbol{A}_f^{-\ell+1} \boldsymbol{B}_f & \text{else.} \end{cases} \tag{4.81}$$

In this version the resulting FIR filter has two window length parameters. Namely, $K_2 > 0$ which is the lookahead as in the FIR/IIR filter version and $K_1 > 0$ which similarly represents how the time window reaches back in time. Note that (4.80) in comparison to (4.78) needs an additional $K_1 - 1$ filter taps to ensure sufficient resolution.

Note that the IIR and FIR filter coefficients above are, for a general control-bounded converter, matrices and not scalars which differs from how IIR/FIR filters are typically described. Clearly, this does not effect the FIR/IIR implementation other than scalar multiplications and additions become their matrix/vector equivalent.

4.3.8 The Digital Estimator as a Quadratic Program

Alternatively, to all the recursive estimators outlined in the previous sections, the digital estimation problem can be seen as the constrained

quadratic program

$$\begin{aligned}\hat{\boldsymbol{u}}(t) = \; & \underset{\tilde{\boldsymbol{u}}(t)}{\operatorname{argmin}} \int_{-\Delta}^{\Delta} \left(\|\boldsymbol{y}(\tau)\|_2^2 + \eta^2 \|\tilde{\boldsymbol{u}}(\tau)\|_2^2\right) \mathrm{d}\tau \\ & \text{such that} \\ & \dot{\boldsymbol{x}}(t) = \; \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\tilde{\boldsymbol{u}}(t) + \boldsymbol{\Gamma}\boldsymbol{s}(t), \;\; t \in [-\Delta, \Delta] \\ & \boldsymbol{y}(t) = \; \boldsymbol{C}^{\mathsf{T}}\boldsymbol{x}(t)\end{aligned} \tag{4.82}$$

where $\Delta > 0$ determines a time window. (4.82) can also been seen as a variation of Kalman smoothing [13, 18] where the estimate $\hat{u}(t)$ is known to converge to that of (4.25) with $\boldsymbol{h}(t)$ as the Wiener filter (4.41) for $\Delta \to \infty$.

4.4 Performance Measure

To formulate the design principle of a control-bounded ADC, we first need to determine a performance measure. To this end, we will adapt the standard SNR measure that is commonly used in the $\Delta\Sigma$ community, see Section 3.4. Additionally, due to our estimate and conversion error's continuous-time nature, the standard SNR definition needs to be slightly adjusted.

To make the following analysis more transparent, we restrict the input signal $\boldsymbol{u}(t)$ to the scalar case. The steps here can be fully extended to the multivariate case. However, this would require a generalized definition of SNR.

In the scalar input case, the mean squared values of the signal and conversion error can be written as

$$\boldsymbol{P}_{\boldsymbol{u}} \triangleq \mathrm{E}\big[u(t)^2\big] \tag{4.83}$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} |\boldsymbol{T}(\omega)|^2 \boldsymbol{S}_{\boldsymbol{u}\boldsymbol{u}^{\mathsf{T}}}(\omega)\, \mathrm{d}\omega \tag{4.84}$$

and

$$\boldsymbol{P}_{\boldsymbol{\epsilon}} \triangleq \mathrm{E}\big[\boldsymbol{\epsilon}(t)^2\big] \tag{4.85}$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} \boldsymbol{H}(\omega) \boldsymbol{S}_{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}}(\omega) \boldsymbol{H}(\omega)^{\mathsf{H}}\, \mathrm{d}\omega \tag{4.86}$$

where $\boldsymbol{y}(t)$ is modeled as a stationary stochastic process with PSD matrix

$$\boldsymbol{S}_{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}}(\omega) \triangleq \int_{-\infty}^{\infty} \mathrm{E}\big[\boldsymbol{y}(t)\boldsymbol{y}(t+\tau)^{\mathsf{T}}\big]\, e^{-i\omega\tau}\, d\tau. \tag{4.87}$$

Note that $\boldsymbol{y}(t)$ being stationary is a statistical assumption that we know to not be true for many input signals of interest. Additionally, we assume the spectrum of the signal observation vector $\boldsymbol{y}(t)$ to be bandlimited i.e.,

$$\boldsymbol{P}_{\boldsymbol{\epsilon}|\mathcal{B}} = \frac{1}{2\pi}\int_{\omega\in\mathcal{B}} \boldsymbol{H}(\omega)\boldsymbol{S}_{\boldsymbol{y}\boldsymbol{y}^\mathsf{T}}\boldsymbol{H}(\omega)^\mathsf{H}\,\mathrm{d}\omega \tag{4.88}$$

$$= \frac{\sigma^2_{\boldsymbol{y}|\mathcal{B}}}{2\pi}\cdot\mathcal{B} \tag{4.89}$$

$$= \boldsymbol{P}_{\boldsymbol{\epsilon}} \tag{4.90}$$

where $\mathcal{B}$ is defined as in (3.10). This cannot be literally true. Regardless, both these assumptions provide a useful model for the analysis as

$$\boldsymbol{S}_{\boldsymbol{y}\boldsymbol{y}^\mathsf{T}}(\omega) \approx \sigma^2_{\boldsymbol{y}|\mathcal{B}}\boldsymbol{I}_M. \tag{4.91}$$

Consequently, we can write

$$\boldsymbol{P}_{\epsilon|\mathcal{B}} \approx \frac{\sigma^2_{\boldsymbol{y}|\mathcal{B}}}{2\pi}\int_{\omega\in\mathcal{B}} \boldsymbol{H}(\omega)\boldsymbol{H}(\omega)^\mathsf{H}\,\mathrm{d}\omega \tag{4.92}$$

$$= \frac{\sigma^2_{\boldsymbol{y}|\mathcal{B}}}{2\pi}\int_{\omega\in\mathcal{B}} \frac{\|\boldsymbol{G}(\omega)\|_2^2}{\left(\|\boldsymbol{G}(\omega)\|_2^2+\eta^2\right)^2}\,\mathrm{d}\omega \tag{4.93}$$

$$\approx \frac{\sigma^2_{\boldsymbol{y}|\mathcal{B}}}{2\pi}\int_{\omega\in\mathcal{B}} \frac{1}{\|\boldsymbol{G}(\omega)\|_2^2}\,\mathrm{d}\omega \tag{4.94}$$

4.5 Design Principle

When designing a control-bounded ADC, the design task naturally splits into two subsequent steps. Firstly, to design the AS such that sufficient gain is provided in the signal band of interest. Secondly, an effective DC is determined that ensures a bounded state vector.

In the first step, the AS is optimized towards a target $\|\boldsymbol{G}(\omega)\|_2^2$ specified over the signal band of interest. In this step, we do not concern ourselves with stability constraints meaning this is a purely analog, continuous-time, and possibly unstable design step. Note that since it is the norm of the transfer function that is optimized, i.e., $\boldsymbol{G}(\omega)^\mathsf{H}\boldsymbol{G}(\omega)$, increasing the amplification by positive feedback (having unstable poles) does not necessarily increase the overall system amplification.

In the second step, when designing the DC, the goal is to bound the AS state vector. This means that the DC can be designed without concern

of any overall transfer function or AS amplification. As an example, the local control from Section 5.3 divides the control task into local DC tasks for every individual state of the AS.

These two design goals are nicely emphasized in the conversion error approximation from (4.94). Namely, the SNR or alternatively, the conversion error suppression, as in (3.8), is determined by the AS's $\|\boldsymbol{G}(\omega)\|_2^2$ norm integrated over the signal band of interest and the noise variance $\sigma^2_{\boldsymbol{y}|\mathcal{B}}$. The magnitude of $\sigma^2_{\boldsymbol{y}|\mathcal{B}}$ follows from the digital controls ability to bound the state vector. In other words, a large AS amplification, in the band of interest, together with a tight control bound, results in a large SNR.

It is possible to consider these two steps independently. However, the DC does depend on the choice of AS. Therefore, there is some value to also optimizing the AS such that it can be more easily controlled for a given DC strategy. An example would be to make a more complicated AS that amplifies the signal band of interest and also suppresses out-of-band signals. For such a system, the DC does not need to have a wideband DAC waveform, which could enable tighter state bounds.

Based on the AS, the DC, and a fixed bandwidth parameter η^2, the DE can be computed as in Section 4.3.2. Note that the DE does not pose any additional constraints on the design of the AS or the DC.

4.6 Non-Idealities

The performance measure of Section 4.4 can be extended to also account for thermal noise and component mismatch. For simplicity, we restrict ourselves to the case where $\boldsymbol{u}(t)$ is scalar as in Section 4.4.

4.6.1 Thermal Noise

Let $z(t)$ be a single thermal noise signal entering at some point in the AS and let $\boldsymbol{g}_z(t)$ be the vector of impulse responses from this noise source to $\boldsymbol{y}(t)$. Under the stated assumptions we can rewrite the signal observation from (4.24) as

$$\boldsymbol{y}(t) = (\boldsymbol{g} * \boldsymbol{u})(t) + (\boldsymbol{g}_z * z)(t) - \boldsymbol{q}(t). \tag{4.95}$$

Subsequently, the estimate from (4.25) follows as

$$\hat{\boldsymbol{u}}(t) = (\boldsymbol{h} * \boldsymbol{q})(t) \tag{4.96}$$
$$= (\boldsymbol{h} * \boldsymbol{g} * \boldsymbol{u})(t) + (\boldsymbol{h} * \boldsymbol{g}_z * z)(t) - (\boldsymbol{h} * \boldsymbol{y})(t), \tag{4.97}$$

where the term

$$\epsilon_z(t) \triangleq (\boldsymbol{h} * \boldsymbol{g}_z * z)(t) \tag{4.98}$$

is the additional error due to $z(t)$.

Assume that, within the frequency band of interest $\mathcal{B}$, $z(t)$ is a stationary stochastic process with a flat PSD

$$S_z(\omega) = \sigma^2_{z|\mathcal{B}}. \tag{4.99}$$

The contribution of (4.98) to the noise power (4.85) is then easily determined to be

$$P_{\epsilon_z} = \frac{\sigma^2_{z|\mathcal{B}}}{2\pi} \int_{\mathcal{B}} \boldsymbol{H}(\omega)\boldsymbol{G}_z(\omega)\boldsymbol{G}_z(\omega)^{\mathsf{H}}\boldsymbol{H}(\omega)^{\mathsf{H}}\, d\omega \tag{4.100}$$
$$= \frac{\sigma^2_{z|\mathcal{B}}}{2\pi} \int_{\mathcal{B}} \frac{|\boldsymbol{G}(\omega)^{\mathsf{H}}\boldsymbol{G}_z(\omega)|}{\left(\|\boldsymbol{G}(\omega)\|_2^2 + \eta^2\right)^2}\, d\omega, \tag{4.101}$$

where $\boldsymbol{G}_z(\omega)$ is the elementwise Fourier transform of $\boldsymbol{g}_z(t)$.

Finally, the total contribution, of multiple such thermal noise sources $z_1(t)$, $z_2(t)$, ... to the noise power (4.85), follows as $P_{\epsilon_{z_1}} + P_{\epsilon_{z_2}} + \ldots$.

4.6.2 Mismatch

Let $\tilde{\boldsymbol{g}}$, $\tilde{\boldsymbol{q}}$, and $\tilde{\boldsymbol{h}}$ be the nominal (i.e., assumed by the DE) values of the actual quantities $\boldsymbol{g}$, $\boldsymbol{q}$, and $\boldsymbol{h}$, respectively. We still have

$$\boldsymbol{y}(t) = (\boldsymbol{g} * u)(t) - \boldsymbol{q}(t), \tag{4.102}$$

but as the DE does not know the nominal AS parameters the estimate from (4.25) follows as

$$\hat{\boldsymbol{u}}(t) = (\tilde{\boldsymbol{h}} * \tilde{\boldsymbol{q}})(t) \tag{4.103}$$
$$= (\tilde{\boldsymbol{h}} * (\tilde{\boldsymbol{q}} - \boldsymbol{q}))(t) + (\tilde{\boldsymbol{h}} * \boldsymbol{q})(t) \tag{4.104}$$
$$= (\tilde{\boldsymbol{h}} * (\tilde{\boldsymbol{q}} - \boldsymbol{q}))(t) + (\tilde{\boldsymbol{h}} * \boldsymbol{g} * \boldsymbol{u})(t) - (\tilde{\boldsymbol{h}} * \boldsymbol{y})(t). \tag{4.105}$$

The total conversion error can then be written as

$$\epsilon(t) \triangleq \hat{\boldsymbol{u}}(t) - (\tilde{\boldsymbol{h}} * \tilde{\boldsymbol{g}} * \boldsymbol{u})(t) \tag{4.106}$$

$$= (\tilde{\boldsymbol{h}} * (\boldsymbol{g} - \tilde{\boldsymbol{g}}) * \boldsymbol{u})(t) + (\tilde{\boldsymbol{h}} * (\tilde{\boldsymbol{q}} - \boldsymbol{q}))(t) - (\tilde{\boldsymbol{h}} * \boldsymbol{y})(t). \tag{4.107}$$

The three terms in (4.107) are of a very different nature. The last term, $-(\tilde{\boldsymbol{h}} * \boldsymbol{y})(t)$, is the nominal conversion error as in (4.43), to which the analysis in Section 4.4 applies essentially unchanged. In other words, the contribution of this term to the in-band noise power (4.85) is essentially unaffected by the mismatch.

The first term in (4.107),

$$\epsilon_{\tilde{\boldsymbol{g}}}(t) \triangleq (\tilde{\boldsymbol{h}} * (\boldsymbol{g} - \tilde{\boldsymbol{g}}) * \boldsymbol{u})(t), \tag{4.108}$$

accounts for a modification of the STF. In principle, this term can be neutralized by calibrating post-filtering. Furthermore, if this term is considered as noise, its magnitude depends on the signal $\boldsymbol{u}(t)$.

If we assume $\boldsymbol{u}(t)$ to be white noise within the band of interest $\mathcal{B}$, the contribution of (4.108) to the in-band noise power can be expressed by an obvious modification of (4.100).

The second term in (4.107) is more troublesome

$$\epsilon_{\tilde{\boldsymbol{q}}}(t) \triangleq (\tilde{\boldsymbol{h}} * (\tilde{\boldsymbol{q}} - \boldsymbol{q}))(t) \tag{4.109}$$

$$= \left(\tilde{\boldsymbol{h}} * \sum_{\ell=1}^{M} (\tilde{\boldsymbol{g}}_{q_\ell} - \boldsymbol{g}_{q_\ell}) * s_\ell\right)(t), \tag{4.110}$$

where $\tilde{\boldsymbol{g}}_{q_\ell}$ and $\boldsymbol{g}_{q_\ell}$ are the nominal and the actual transfer functions, respectively, from $s_\ell(t)$ to $\boldsymbol{y}(t)$.

If we boldly assume $s_1(t), \ldots, s_M(t)$ to be stochastic processes with a flat PSD, within the band $\mathcal{B}$ of interest, the contribution of (4.110) to the in-band noise power can also be expressed by an obvious modification of (4.100). However, depending on the DC, the white-noise assumption may be too bold, cf. Figure 5.20 from the chain-of-integrators ADC example. In any case, the PSD of $\boldsymbol{s}(t)$ (for a specific input signal $\boldsymbol{u}(t)$) can be determined by simulations.

4.7 Relation to Delta-Sigma Modulators

We already acknowledged that for the special case of a single state, single input, and single control contribution, the control-bounded ADC and the $\Delta\Sigma$ modulator result in an identical AS and DC. We also established that the DE, alternatively the digital cancellation and decimation filter in the conventional view, were fundamentally different approaches in terms of how the signal is represented and post-filtered. We will next investigate the two different filters for the same AS and DC. Furthermore, in Section 4.7.2 we will show how the MASH $\Delta\Sigma$ modulator can be described as a control-bounded ADC. Subsequently, in Section 4.7.3 we generalize the concept of digital-cancellation logic such that this principle can be compared, at the same level of abstraction, to the general DE of the control-bounded ADC.

4.7.1 Transfer Function Comparison

To demonstrate the difference between the DE from Section 4.3 and the standard $\Delta\Sigma$ modulator approach we compare the STF and NTF for each case as well as the SNR. In other words we are comparing (3.2) and (3.4), from the $\Delta\Sigma$ modulator case, with (4.41) and (4.44) from the control-bounded ADC case. As these expressions depend on the AS, or alternatively the loop filter, we set

$$G_{\mathrm{CI}_1}(\omega) = \frac{\beta}{i\omega} \tag{4.111}$$

i.e., a first-order integrator system. Furthermore, we set the control period and the sample period equally as

$$T = T_s \tag{4.112}$$

$$= 1/(2\beta). \tag{4.113}$$

This parameter choice will ensure an effective controller and is further described in Section 5.3. The four different transfer functions are shown in Figure 4.6.

We know that the definition of SNR depends on the integral of the signal's and error's PSD over the frequency band of interest. However, from Figure 4.6, it is not immediately clear how this compares for the two approaches. Therefore, we conceptually apply a full scale sinusoidal test signal with a frequency of f. Furthermore, we assume a bandlimited flat

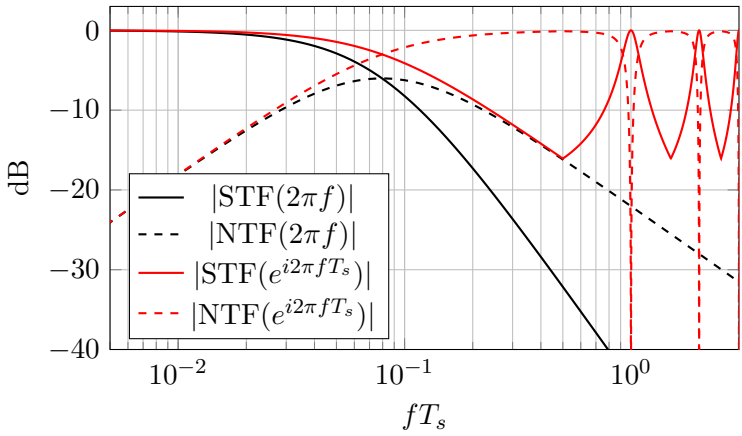


Figure 4.6: Comparison of NTF and STF for a first order integrator system. The black lines represents the control-bounded ADC as in (4.44). The red lines represent the continuous-time $\Delta\Sigma$ modulator system as in (3.2) and (3.4), i.e. the STF and NTF of the corresponding $\Delta\Sigma$ modulator from input to bitstream without subsequent filtering. Note that the control-bounded DE filter implicitly applies a low pass filter which is not the case for the $\Delta\Sigma$ modulator.

spectrum PSD conversion error/quantization error of the same expected square magnitude per unit time in both cases. The resulting SNR is shown in Figure 4.7. From the figure, it is clear that for signals with the frequencies well below the unit gain of the AS, the two different approaches perform essentially equivalent.

The proposed analysis could additionally be applied for a higher-order system $\boldsymbol{G}(\omega)$. However, note that in this case, additional assumptions need to be made. Specifically, the control-bounded ADC has N outputs, and as many NTFs, for a N-th order system, whereas the $\Delta\Sigma$ modulator only has one.

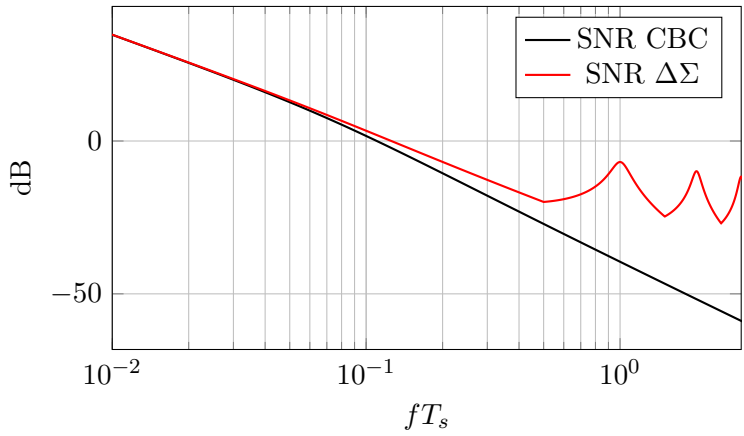


Figure 4.7: SNR plot for a sinusoidal input signal of frequency f and assuming the same PSD for both the conversion error as in (4.40) and the quantization error as in (3.15). Similarly as in Figure 4.6 the black line corresponds to the control-bounded ADC case and the red to the $\Delta\Sigma$ modulator.

4.7.2 MASH State Space Representation

A continuous-time version of the MASH $\Delta\Sigma$ converter from Section 3.5 can also be rearranged in the same form as in Figure 4.1 with a state space representation

$$\boldsymbol{A}_{\text{MASH}} = \begin{pmatrix} \boldsymbol{A}_1 & & & \\ -\boldsymbol{B}_2\boldsymbol{C}_1^{\mathsf{T}} & \boldsymbol{A}_2 & & \\ & \ddots & \ddots & \\ & & -\boldsymbol{B}_N\boldsymbol{C}_{N-1} & \boldsymbol{A}_N \end{pmatrix}, \tag{4.114}$$

$$\boldsymbol{B}_{\text{MASH}} = \begin{pmatrix} \boldsymbol{B}_1 \\ \boldsymbol{0}_{N\times L} \\ \vdots \end{pmatrix}, \tag{4.115}$$

$$\boldsymbol{C}_{\text{MASH}}^{\mathsf{T}} = \begin{pmatrix} \boldsymbol{C}_1^{\mathsf{T}} & & \\ & \boldsymbol{C}_2^{\mathsf{T}} & \\ & & \ddots \end{pmatrix}, \tag{4.116}$$

$$\boldsymbol{\Gamma}_{\text{MASH}}(\omega) = \begin{pmatrix} D_1(\omega) & & & \\ D_1(\omega) & D_2(\omega) & & \\ & D_2(\omega) & D_3(\omega) & \\ & & \ddots & \ddots \end{pmatrix}, \tag{4.117}$$

and $\tilde{\boldsymbol{\Gamma}}_{\text{MASH}} = \boldsymbol{C}_{\text{MASH}}^{\mathsf{T}}$. Furthermore, in the given representation each subsystem $G_1(\omega)$, $G_2(\omega)$, …, $G_N(\omega)$ are additionally written in state space form such that

$$G_\ell(\omega) = \boldsymbol{C}_\ell^{\mathsf{T}} \left(i\omega \boldsymbol{I}_{\breve{N}_\ell} - \boldsymbol{A}_\ell \right)^{-1} \boldsymbol{B}_\ell \tag{4.118}$$

where $\breve{N}_\ell$ is the system order of the ℓ-th system and $D_\ell(\omega)$ represents the frequency response of the ℓ-th DAC.

4.7.3 Generalized Digital Cancellation Logic

The MASH $\Delta\Sigma$ converter relies on a digital cancellation filter to produce its estimate. This concept can be extended for many of the control-bounded ADCs presented in this thesis. We will generalize this concept such that the MASH digital-cancellation logic can be compared at the same level of abstraction as the digital filter of the control-bounded ADC approach. The goal of this generalization is to highlight the difference in the approaches and finally to see why the control-bounded ADC allows more general ADC systems. Note that the state space representation used here is the general one from Section 4.1.1 and need not be the one presented in the context of the MASH $\Delta\Sigma$.

Firstly, following the steps in Section 3.3 by modeling the quantizers as additive white noise results in the approximated model as in Figure 4.8. The model from Figure 4.8 does not directly translate to a traditional state space form. To emphasize the quantization error seen as an input to the system, we use the given model's linearity to redraw the figure as in Figure 4.9. Subsequently, the frequency response of the output $\boldsymbol{y}[k]$ follows as

$$\boldsymbol{Y}(e^{i\Omega}) = \left(\boldsymbol{I}_{\tilde{N}} + \tilde{\boldsymbol{G}}_{\boldsymbol{\Gamma}}(e^{i\Omega}) \right) \boldsymbol{Z}(e^{i\Omega}) + \tilde{\boldsymbol{G}}_{\boldsymbol{B}}(e^{i\Omega}) \tilde{\boldsymbol{U}}(e^{i\Omega}) \tag{4.119}$$

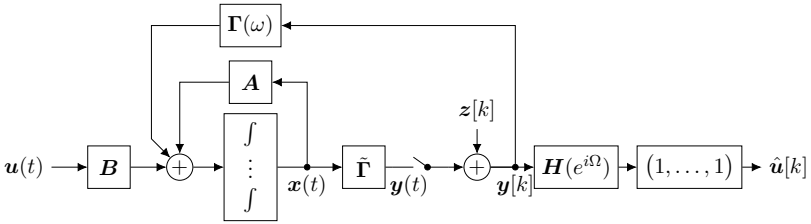


Figure 4.8: The linearized model as in Figure 3.4 for the MASH $\Delta\Sigma$ converter.

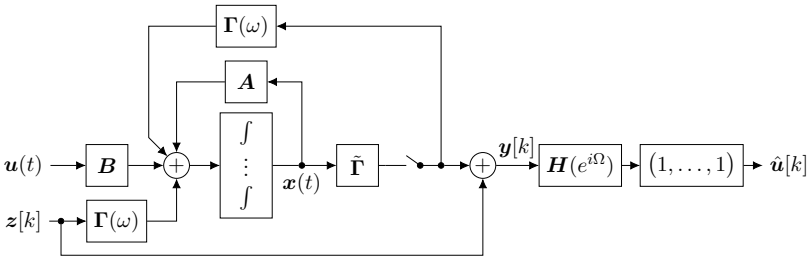


Figure 4.9: A MASH $\Delta\Sigma$ converter represented using a state space model where the quantization error is modeled as an input signal.

where

$$\tilde{\boldsymbol{G}}_{\boldsymbol{B}}(e^{i\Omega}) = \sum_{k\in\mathbb{Z}} \tilde{\boldsymbol{\Gamma}}\tilde{\boldsymbol{G}}\left(\frac{\Omega - 2\pi k}{T_s}\right)\boldsymbol{B} \in \mathbb{C}^{\tilde{N}\times L}, \tag{4.120}$$

$$\tilde{\boldsymbol{G}}_{\boldsymbol{\Gamma}}(e^{i\Omega}) = \frac{1}{T_s}\sum_{k\in\mathbb{Z}} \tilde{\boldsymbol{\Gamma}}\tilde{\boldsymbol{G}}\left(\frac{\Omega - 2\pi k}{T_s}\right)\boldsymbol{\Gamma}\left(\frac{\Omega - 2\pi k}{T_s}\right) \in \mathbb{C}^{\tilde{N}\times M}, \tag{4.121}$$

$$\tilde{\boldsymbol{G}}(\omega) = \left(i\omega\boldsymbol{I}_N - \boldsymbol{A} - \frac{1}{T_s}\sum_{\ell\in\mathbb{Z}} \tilde{\boldsymbol{\Gamma}}\boldsymbol{\Gamma}(\omega - 2\pi\ell/T_s)\right)^{-1} \in \mathbb{C}^{N\times N}, \tag{4.122}$$

$\Omega = \omega T_s$, and both $\tilde{\boldsymbol{U}}(e^{i\Omega}) \in \mathbb{R}^L$ and $\tilde{\boldsymbol{Z}}(e^{i\Omega}) \in \mathbb{R}^M$ are as in Section 3.3 but extended to the multidimensional setting.

From (4.119) we recognize the multi-dimensional NTF and STF of the

system as

$$\mathrm{NTF}(e^{i\Omega}) = \boldsymbol{I}_{\tilde{N}} + \tilde{\boldsymbol{G}}_{\boldsymbol{\Gamma}}(e^{i\Omega}) \in \mathbb{C}^{\tilde{N}\times M} \tag{4.123}$$

$$\mathrm{STF}(e^{i\Omega}) = \tilde{\boldsymbol{G}}_{\boldsymbol{B}}(e^{i\Omega}) \in \mathbb{C}^{\tilde{N}\times L} \tag{4.124}$$

The digital-cancellation logic filters $\boldsymbol{H}(e^{i\Omega}) = \big(H_1(e^{i\Omega}), \dots, H_{\tilde{N}}(e^{i\Omega})\big)^{\mathsf{T}}$ would ideally be determined by solving the linear equation system

$$\begin{pmatrix} \mathrm{STF}(e^{i\Omega})^{\mathsf{T}} \\ \mathrm{NTF}(e^{i\Omega})^{\mathsf{T}} \end{pmatrix} \boldsymbol{H}(e^{i\Omega})^{\mathsf{T}} = \begin{pmatrix} \boldsymbol{T}(e^{i\Omega}) \\ \boldsymbol{0}_{M\times 1} \end{pmatrix} \tag{4.125}$$

as this would result in the estimate

$$\hat{\boldsymbol{U}}(e^{i\Omega}) = \big(1, \dots, 1\big)\, \boldsymbol{H}(e^{i\Omega}) \left(\mathrm{NTF}(e^{i\Omega})\boldsymbol{Z}(e^{i\Omega}) + \mathrm{STF}(e^{i\Omega})\tilde{\boldsymbol{U}}(e^{i\Omega})\right) \tag{4.126}$$

$$= \boldsymbol{T}(e^{i\Omega})\tilde{\boldsymbol{U}}(e^{i\Omega}) \tag{4.127}$$

where $\boldsymbol{T}(e^{i\Omega}) \in \mathbb{C}^{L}$ is the multidimensional target transfer function.

However, as in the MASH $\Delta\Sigma$ case, this system of equations is overdetermined, having $M + L > \tilde{N}$ equations for $\tilde{N}$ variables, meaning that we cannot cancel all M quantization error contributions.

In the MASH $\Delta\Sigma$ converter from Section 3.5, there is a clear hierarchy between the scalar quantization errors in $\boldsymbol{Z}(e^{i\Omega})$, such that suppressing all but the last dimension is the optimal strategy, i.e.

$$\begin{pmatrix} \boldsymbol{I}_{M-1} & \boldsymbol{0}_{M\times 1} \end{pmatrix} \begin{pmatrix} \mathrm{STF}(e^{i\Omega})^{\mathsf{T}} \\ \mathrm{NTF}(e^{i\Omega})^{\mathsf{T}} \end{pmatrix} \boldsymbol{H}(e^{i\Omega})^{\mathsf{T}} = \begin{pmatrix} \boldsymbol{T}(e^{i\Omega}) \\ \boldsymbol{0}_{(M-1)\times 1} \end{pmatrix}. \tag{4.128}$$

This system of equations, in contrast to (4.125), could potentially be fully determined and if so results in $\tilde{N}$ digital cancellation filters. However, for a more general control-bounded converter, there might not be a clear smallest quantization error candidate and requires additional assumptions to formulate a cancellation strategy. As an alternative, it would be possible to find the least-squares solution of (4.125). We will not spend more time addressing these issues as they are only relevant in case we wanted to use conventional tools for the new generalized control-bounded ADC presented in this thesis.

However, an important insight from the given analysis is that, regardless of how the cancellation conditions are chosen, there are choices of $\boldsymbol{B}$ and

$\boldsymbol{\Gamma}$ that prohibits this generalized digital-cancellation logic. An example would be when $\boldsymbol{\Gamma}$ contains $\boldsymbol{B}$ within its columns. In that case, the quantization error corresponding to those columns cannot be canceled since that would contradict the input signal to be part of the estimate. A clear example is the chain-of-integrators example in Chapter 5.

On the contrary, for the DE of Section 4.3, there are no conditions to be specified or that could be violated. In fact, as was the topic of Section 4.5, for the control-bounded ADC the AS design and the DC can essentially be done separate steps and the DE follows from their parametrization.

In summary, we recognize that the control-bounded ADC can be viewed as a generalization of the MASH $\Delta\Sigma$ concept with a less restrictive cancellation logic (DE) that enables a greater AS and DC design space.

4.8 Simulating a Control-Bounded Analog-to-Digital Converter

By simulating a control-bounded ADC we refer to the interaction between the AS and DC as the system is excited by an analog input signal $\boldsymbol{u}(t)$, see Figure 4.1. In other words, we are simulating the underlying analog circuitry and this should not be confused with the purely digital operations of the DE. The operation and implementation of the DE is covered separately in Section 4.3.4, Section 4.3.5, and Appendix E.

We established in Section 4.1 that AS is modeled by a system of ODEs

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{\Gamma}\boldsymbol{s}(t). \tag{4.129}$$

Due to the linear relationship between the input, control signal, and prior state vector in the system of ODEs (4.129), a general solution, at time t, will be of the form

$$\begin{aligned}\boldsymbol{x}(t) = &\exp\left(\boldsymbol{A}(t-t_0)\right)\boldsymbol{x}(t_0)\\ &+\int_{t_0}^{t}\exp\left(\boldsymbol{A}(t-\tau)\right)\boldsymbol{B}\boldsymbol{u}(\tau)\,\mathrm{d}\tau\\ &+\int_{t_0}^{t}\exp\left(\boldsymbol{A}(t-\tau)\right)\boldsymbol{\Gamma}\boldsymbol{s}(\tau)\,\mathrm{d}\tau\end{aligned} \tag{4.130}$$

where we remind the reader that $\exp(\cdot)$ refers to a matrix exponential and $t_0 \in \mathbb{R}$ represents some initial starting time.

What is not immediately clear, from the used notation, is that the general solution in (4.130) is a function of a quantized version of the AS state vector $\boldsymbol{x}(t)$ as the control signal $\boldsymbol{s}(t)$ is updated by the DC at regular time intervals. Specifically, as the DC interaction is synchronous with a global clock operating with a clock period T, the control signal at times $(k-1)T \leq t < kT$ is determined based on the state vector at time $\boldsymbol{x}((k-1)T)$. Therefore, simulating the analog part of the control-bounded ADC amounts to evaluating the AS state vector at uniformly spaced times $\{t_0, t_0 + T, \ldots, t_0 + kT, \ldots\}$ where the previous state evaluation determines the control signal for the next state evaluation. The described procedure reduces to solving a sequence of inital value problems (IVPs). Specifically, the k-th step involves

1. Determining the control signal $\boldsymbol{s}(t)$ for the times $t \in [(k-1)T, kT)$ by evaluating a quantized version of the $\tilde{\boldsymbol{\Gamma}}\boldsymbol{x}((k-1)T)$. This corresponds to the control update done by the DC.

2. Solving the IVP, i.e. computing $\boldsymbol{x}(kT)$, given the previous solution $\boldsymbol{x}((k-1)T)$, and evaluating the control contribution $\boldsymbol{s}(t)$ and input signal $\boldsymbol{u}(t)$ for the times $t \in [(k-1)T, kT)$. This corresponds to the state evolution seen in the AS.

Solving IVPs is a standard problem in many engineering disciplines and therefore there exists a multitude of numerical techniques and software tools that can be used when simulating the control-bounded ADC. Typically, this means using numerical analysis techniques such as the family of Runge-Kutta methods.

4.8.1 Precomputed Control Contributions

For the proposed iterative scheme of solving IVPs the part of the solution involving the control-contribution, from here on denoted $\boldsymbol{x}_{\boldsymbol{s}}(t)$, is time-invariant except for the control signal, i.e.

$$\begin{aligned}
\boldsymbol{x}_{\boldsymbol{s}}(kT) &= \int_{(k-1)T}^{kT} \exp\left(\boldsymbol{A}(kT-\tau)\right) \boldsymbol{\Gamma}\boldsymbol{s}(\tau)\,\mathrm{d}\tau \\
&= \int_{(k-1)T}^{kT} \exp\left(\boldsymbol{A}(kT-\tau)\right) \boldsymbol{\Gamma}\boldsymbol{D}(\tau-(k-1)T)\boldsymbol{s}[k-1]\,\mathrm{d}\tau \\
&= \boldsymbol{\Gamma}_{\boldsymbol{x}}\boldsymbol{s}[k-1] \qquad (4.131)
\end{aligned}$$

where we used the control contribution definition from (4.9) resulting in the precomputed control contribution definition

$$\boldsymbol{\Gamma}_{\boldsymbol{x}} \triangleq \int_0^T \exp\left(\boldsymbol{A}(T-\tau)\right) \boldsymbol{\Gamma} \boldsymbol{D}(\tau)\, \mathrm{d}\tau. \tag{4.132}$$

We recognize the mentioned time-invariance as $\boldsymbol{x}_{\boldsymbol{s}}(t)$ only depends on the control signal $\boldsymbol{s}[k]$ for $t \in \{\dots, (k-1)T, kT, (k+1)T, \dots\}$.

As (4.132) only needs to be computed once this can be done at higher precision thereby improving the overall quality of the simulation. Furthermore, using precomputed control contributions simplifies the IVP as (4.129) reduces to

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{4.133}$$

and thereby the general solution follows by

$$\begin{aligned} \boldsymbol{x}(kT) = &\exp\left(\boldsymbol{A}(T)\right) \boldsymbol{x}\left((k-1)T\right) \\ &+ \int_{t_0}^{t} \exp\left(\boldsymbol{A}(t-\tau)\right) \boldsymbol{B}\boldsymbol{u}(\tau)\, \mathrm{d}\tau \\ &+ \boldsymbol{\Gamma}_{\boldsymbol{x}} \boldsymbol{s}[k-1] \end{aligned} \tag{4.134}$$

Additionally, we recognize that the precomputed IVP solution from (4.132) resembles that of the precomputed filter coefficients $\boldsymbol{B}_f$ and $\boldsymbol{B}_b$ from (4.63) and (4.64) respectively. This is no coincidence as the digital estimation filter from Section 4.3.2 results in yet another IVP. This means that we can use similar IVP solvers when computing the DE's filter coefficients. However, we once more remind ourselves that the simulation, i.e. the interaction between the AS and DC, is a separate problem from that of the DE.

4.8.2 Adding Noise Sources

When simulating analog circuits it is often relevant to additionally include the effects of noise processes such as thermal noise. In general, adding random processes into our ODEs transforms them into stochastic differential equations (SDEs). These are typically much more demanding to simulate and evaluate. In this thesis we restrict ourselves to noise sources in the form of additive stochastic Gaussian processes. This greatly simplifies the general SDEs simulation as we can use the previously proposed IVP solvers with and additional step as described below.

Using the precomputed control contributions from Section 4.8.1, the system of SDEs follows from ODEs in (4.133) as

$$\mathrm{d}\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{x}(t)\,\mathrm{d}t + \boldsymbol{B}\boldsymbol{u}(t)\,\mathrm{d}t + \boldsymbol{\Psi}\,\mathrm{d}\boldsymbol{W}(t) \tag{4.135}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{N\times V}$ is referred to as the noise steering matrix,

$$\mathrm{d}\boldsymbol{W}(t) \triangleq \frac{\mathrm{d}\boldsymbol{W}(t)}{\mathrm{d}t} \cdot \mathrm{d}t, \tag{4.136}$$

and $\boldsymbol{W}(t)$ is a vectorized standard Brownian motion, i.e. $\boldsymbol{W}(0) = \boldsymbol{0}_V$ almost surely, $\mathrm{E}[\boldsymbol{W}(t)] = \boldsymbol{0}_V$, and $\mathrm{E}\big[\boldsymbol{W}(t)\boldsymbol{W}(t)^{\mathsf{T}}\big] = t \cdot \boldsymbol{I}_V$. Note that in the expressions above the dimensioning is such that we consider V independent noise sources.

Due to the additive nature of the Brownian motion, the solution to the SDE from (4.135) at times $t \in \{\dots, (k-1)T, kT, (k+1)T, \dots\}$ reduces to a normal random vector. Therefore, the corresponding multivariate Gaussian density, at time t, is parameterized by a mean vector $\boldsymbol{m}(t) \in \mathbb{R}^N$ and a covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{x}}(t) \in \mathbb{R}^{N\times N}$. The mean vector $\boldsymbol{m}(t)$ is determined, as previously covered, by the deterministic solution as in (4.130). Furthermore, the covariance matrix, evaluated at some time t, can be estimated by solving the IVP

$$\dot{\boldsymbol{\Sigma}}_{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{x}}(t) + \boldsymbol{\Sigma}_{\boldsymbol{x}}(t)\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}} \tag{4.137}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}}(0) = \boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}}. \tag{4.138}$$

In principle we could solve a vectorized version of (4.137) by the methods previously discussed. However, there is a closed form solution

$$\boldsymbol{\Sigma}_{\boldsymbol{x}}(t) \triangleq \int_0^t \exp\left(\boldsymbol{A}\tau\right) \boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}} \exp\left(\boldsymbol{A}^{\mathsf{T}}\tau\right) \mathrm{d}\tau. \tag{4.139}$$

Note that this general solution implies that

$$\dot{\boldsymbol{\Sigma}}_{\boldsymbol{x}}(t) = \exp\left(\boldsymbol{A}t\right) \boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}} \exp\left(\boldsymbol{A}^{\mathsf{T}}t\right) \tag{4.140}$$

which does not obviously agree with the condition (4.137). However, this

can be confirmed by the following manipulations

$$\begin{aligned}
\dot{\boldsymbol{\Sigma}}_{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{x}}(t) + \boldsymbol{\Sigma}_{\boldsymbol{x}}(t)\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}} && (4.141)\\
&= \int_0^t \boldsymbol{A}\exp\left(\boldsymbol{A}\tau\right)\boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}}\exp\left(\boldsymbol{A}^{\mathsf{T}}\tau\right)\mathrm{d}\tau \\
&\quad + \int_0^t \exp\left(\boldsymbol{A}\tau\right)\boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}}\exp\left(\boldsymbol{A}^{\mathsf{T}}\tau\right)\boldsymbol{A}^{\mathsf{T}}\mathrm{d}\tau + \boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}} && (4.142)\\
&= \left[\exp\left(\boldsymbol{A}\tau\right)\boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}}\exp\left(\boldsymbol{A}^{\mathsf{T}}\tau\right)\right]_{\tau=0}^{\tau=t} + \boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}} && (4.143)\\
&= \exp\left(\boldsymbol{A}t\right)\boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}}\exp\left(\boldsymbol{A}^{\mathsf{T}}t\right) - \boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}} + \boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}} && (4.144)\\
&= \exp\left(\boldsymbol{A}t\right)\boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{T}}\exp\left(\boldsymbol{A}^{\mathsf{T}}t\right) && (4.145)
\end{aligned}$$

where (4.142) follows from plugging in the solution (4.139) and (4.144) follows from the fact that $\exp\left(\boldsymbol{0}_{N\times N}\right) = \boldsymbol{I}_N$.

In summary, solving the system of SDEs divides into a sequence of steps where the solution at each time $t \in \{..., kT, (k+1)T, ...\}$, corresponding to a control update, is computed by

1. Determining the control signal $\boldsymbol{s}(t)$ for the times $t \in [(k-1)T, kT)$ by evaluating a quantized version of the $\tilde{\boldsymbol{\Gamma}}\boldsymbol{x}((k-1)T)$. This corresponds to the control update done by the DC.
2. Compute the mean vector $\boldsymbol{m}(kT)$ by solving the IVP as in (4.134) given the previous solution $\boldsymbol{x}((k-1)T)$, and evaluating the control contribution $\boldsymbol{s}(t)$ and input signal $\boldsymbol{u}(t)$ for $t \in [(k-1)T, kT)$.
3. Solve the SDE at time $t = kT$ by sampling the state vector $\boldsymbol{x}(kT)$ from the multivariate Gaussian density $\mathcal{N}\left(\boldsymbol{m}(kT), \boldsymbol{\Sigma}_{\boldsymbol{x}}(T)\right)$.

Chapter 5

Chain-of-Integrators Analog-to-Digital Converter

The chain-of-integrators ADC, first introduced in [19, 38] and extended in [20], is in several ways the textbook example of a control-bounded ADC. Partly because of the straightforward analog structure that in turn demonstrates most of the key concepts behind the control-bounded ADCs, and partly because it resembles and performs similarly to a MASH $\Delta\Sigma$ converter from Section 3.5.

The chain-of-integrators ADC demonstrate excellent nominal conversion performance. However, a naive implementation is error-prone due to its sensitivity to circuit imperfections and limit cycles. Additionally, the chain-of-integrators converter serves as a starting point for the converters presented in the following chapters.

5.1 General Structure

The chain-of-integrators AS and DC are shown in Figure 5.1. As suggested by the name, the analog part is a chain-of-integrators with an amplification factor β_ℓ for each ℓ-th node in the chain. Furthermore, the DC independently interacts with each node in the chain via a single bit

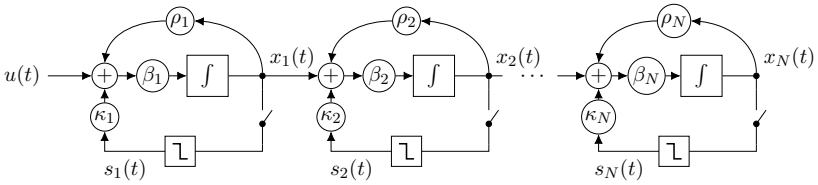


Figure 5.1: The chain-of-integrators ADC where each AS state is connected sequentially, thus forming a chain. Furthermore, the DC is local to each state. The figure only shows the AS and the DC as control-bounded ADCs has a general DE, outlined in Section 4.3.

quantizer, DAC and the weight κ_ℓ. We call such a DC local, and this will be the topic of Section 5.3.

The local DC together with the chain AS structure makes a very modular architecture where we can simply, at least in principle, add or remove nodes to achieve target performance.

5.2 Analog System

The simplistic structure is also revealed when writing out the corresponding state space representation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{\mathrm{CI}}\boldsymbol{x}(t) + \boldsymbol{B}_{\mathrm{CI}}\boldsymbol{u}(t) + \boldsymbol{\Gamma}_{\mathrm{CI}}\boldsymbol{s}(t) \tag{5.1}$$

where

$$\boldsymbol{A}_{\mathrm{CI}} = \begin{pmatrix} \rho_1 & & & \\ \beta_2 & \rho_2 & & \\ & \ddots & \ddots & \\ & & \beta_N & \rho_N \end{pmatrix} \in \mathbb{R}^{N\times N} \tag{5.2}$$

$$\boldsymbol{B}_{\mathrm{CI}} = \begin{pmatrix} \beta_1 & 0 & \cdots & 0 \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}^{N\times 1} \tag{5.3}$$

$$\boldsymbol{\Gamma}_{\mathrm{CI}} = \begin{pmatrix} \beta_1\kappa_1 & & & \\ & \ddots & & \\ & & & \beta_N\kappa_N \end{pmatrix} \in \mathbb{R}^{N\times N}. \tag{5.4}$$

Additionally, the control observation matrix follows as $\tilde{\boldsymbol{\Gamma}}_{\mathrm{CI}}^{\mathsf{T}} = \boldsymbol{I}_N$.

The chain-of-integrators ADC has a scalar input. However, the signal observation

$$\boldsymbol{y}(t) = \boldsymbol{C}_{\mathrm{CI}}^{\mathsf{T}} \boldsymbol{x}(t) \tag{5.5}$$

is not necessarily scalar. This leaves us with two reconstruction modes. Firstly, we could choose a scalar output signal via the signal observation matrix

$$\boldsymbol{C}_{\mathrm{CI_s}}^{\mathsf{T}} = \begin{pmatrix} 0 & \dots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{1 \times N}. \tag{5.6}$$

This choice is equivalent to selecting the last state $x_N(t)$ as the signal observation and is motivated by the chain-like structure were the largest amplification is sustained at the end of the chain. Secondly, choosing all of the states as observations by the signal observation matrix

$$\boldsymbol{C}_{\mathrm{CI_m}}^{\mathsf{T}} = \boldsymbol{I}_N \tag{5.7}$$

generally gives better performance at the expense of more involved analysis. Notice that the DE computational complexity is unchanged for both of these reconstruction modes. Note that the observation matrix $\boldsymbol{C}_{\mathrm{CI}}$ is a purely conceptual quantity that does not effect the implementation of the AS or DC. However, the difference between (5.6) and (5.7) can be described as that the DE considers all or only the last of the AS's state bounded.

Transfer Function Analysis

To analyse the transfer function of the chain-of-integrators converter we require the ATF matrix $\boldsymbol{G}(\omega)$, see (4.7), of the system. As we only consider scalar inputs, the elements of the ATF matrix (now being a column vector) are computed as

$$G_k(\omega) = \prod_{\ell=1}^{k} \frac{\beta_\ell}{i\omega - \rho_\ell}. \tag{5.8}$$

The two reconstruction modes presented above results in different AS amplifications. Specifically,

$$\|\boldsymbol{G}_{\mathrm{CI}_S}(\omega)\|_2^2 = |G_N(\omega)|^2 \tag{5.9}$$

$$= \left(\frac{\beta^2}{\omega^2 + \rho^2} \right)^N \tag{5.10}$$

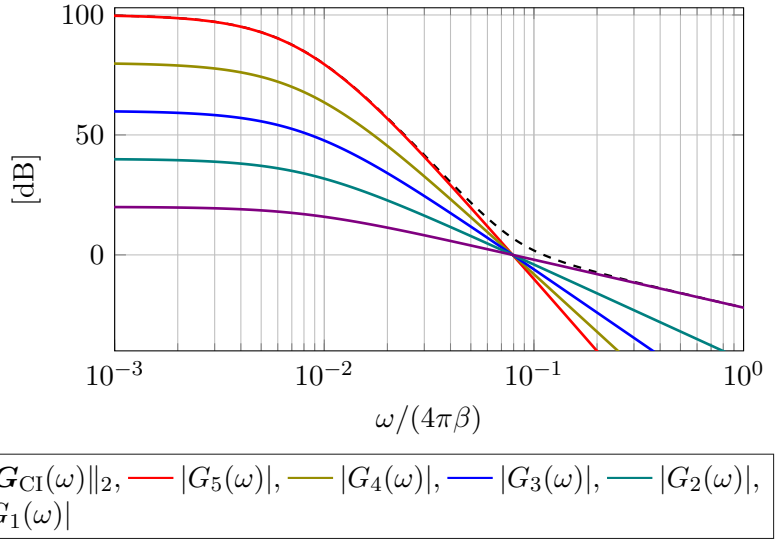


Figure 5.2: The amplitude response for the AS of chain-of-integrators converter where the ATF matrix $\boldsymbol{G}(\omega) = \big(G_1(\omega), \ldots, G_5(\omega)\big)^{\mathsf{T}}$, is parameterized as $\beta_1 = \cdots = \beta_5 = \beta$ and $\rho_1, = \cdots = \rho_5 = -\beta/10$.

for the single-output case, i.e., $\boldsymbol{C}_{\mathrm{CI_s}}$, and

$$\|\boldsymbol{G}_{\mathrm{CI}}(\omega)\|_2^2 = \frac{1 - \left(\frac{\omega^2+\rho^2}{\beta^2}\right)^N}{\left(\frac{\omega^2+\rho^2}{\beta^2}\right)^N \left(1 - \frac{\omega^2+\rho^2}{\beta^2}\right)} \tag{5.11}$$

for the multi-output case, i.e., $\boldsymbol{C}_{\mathrm{CI_m}}$, where we have fixed the parametrization as $\beta_1 = \cdots = \beta_N = \beta$ and $\rho_1 = \cdots = \rho_N = \rho$ to make the expressions more tractable. The two transfer function norms are visualized in Figure 5.2 for $\rho = -\beta/10$. From the figure, the incremental performance gain is visible as the chain goes from a single node chain $G_1(\omega)$, to a five node chain $G_5(\omega)$. Additionally, for low frequencies, we see the amplification-increase flatten out as a result of the local negative feedback ρ. Notice that only where the amplification difference is small, i.e., in the proximity of the unit-gain frequency of the AS

$$f_{0\mathrm{dB}} = \frac{\sqrt{\beta^2 - \rho^2}}{2\pi}, \tag{5.12}$$

do we recognize a substantial difference between the single-output AS and the multi-output AS approaches.

In summary, the multi-output AS reconstruction always outperforms the single-output AS. However, for analysis purposes, the single-output AS reconstruction is often more tractable. Also, as seen from Figure 5.2, the single-output AS reconstruction is a good proxy for a large portion of the frequency spectrum.

5.3 Local Digital Control

As is evident from Figure 5.1, for the chain-of-integrators ADC, each AS state is controlled via a dedicated local control. Furthermore, each local DC is of very low complexity as it interacts by choosing between two control contributions $s_\ell(t) \in \{+d_\ell(t - kT), -d_\ell(t - kT)\}$ for any $t \in [kT, (k + 1)T)$ given a binary control observation $\tilde{s}_\ell(kT) \in \{+1, -1\}$, see Section 4.2.1. Clearly, the local DCs low complexity is attractive from an implementation point of view. Additionally, it provides us with a recursive way of ensuring an effective control.

5.3.1 Effective Control

The local DC recursively ensures a bounded output given a bounded input for each node in the chain. To see this, we first only consider the first node of the chain. Furthermore, we denote the corresponding system impulse response by $g_1(t)$, its scalar state $x_1(t)$, and we assume the DAC waveform to be square as in (4.11). For an input signal, upper and lower bounded by $\pm b_u = \pm b_{\boldsymbol{x}}$, the growth term can then be written as

$$G_1(t) = \max_{u \in \mathcal{U}} |(g_1 * u)(t)| \tag{5.13}$$

$$= \mathrm{step}(t) \cdot b_u \tag{5.14}$$

where

$$\mathrm{step}(t) \triangleq \begin{cases} \beta_1 t & \text{if } \rho_1 = 0, \\ \frac{\beta_1}{\rho_1} \left(e^{\frac{t}{\rho_1}} - 1 \right) & \text{otherwise.} \end{cases} \tag{5.15}$$

Note that $G_\ell(t)$ represents the growth term and not the transfer function element $G_\ell(\omega)$.

Similarly, the remainder term can be written as

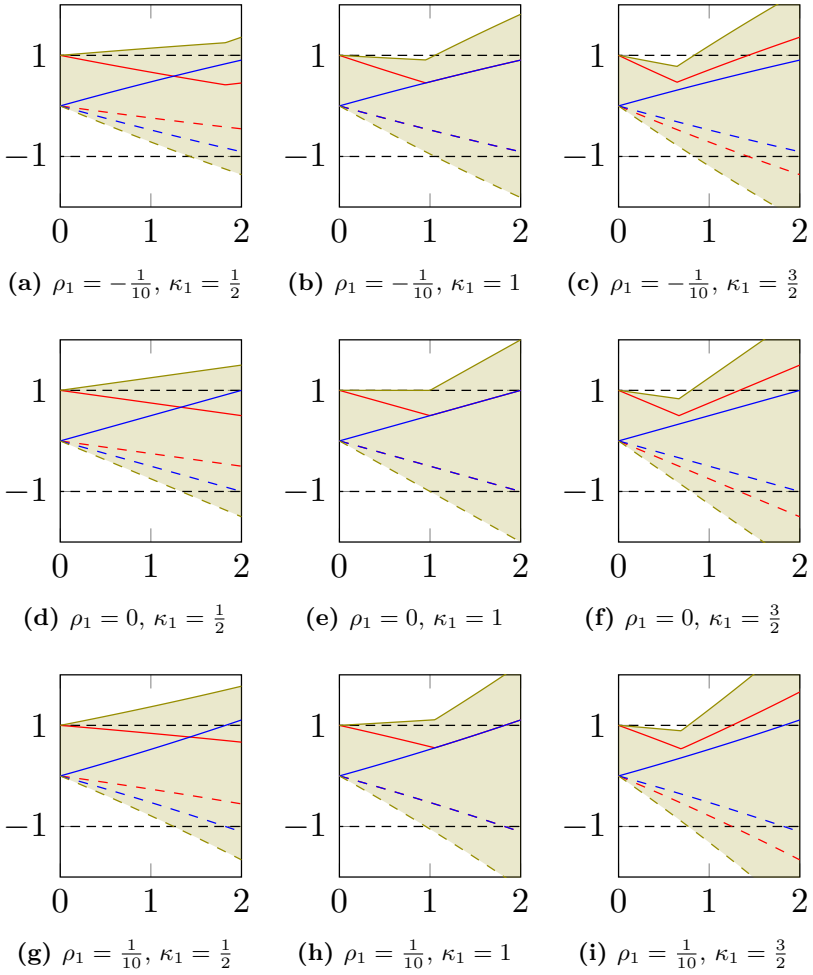
$$R_1(t) = \max_{x_1(0)\in[-b_{\boldsymbol{x}},b_{\boldsymbol{x}}]} |g_1(t)\cdot x_1(0) + (g_1 * s_1)(t)| \tag{5.16}$$

$$= \max_{x_1(0)\in[-b_{\boldsymbol{x}},b_{\boldsymbol{x}}]} \left\{\text{step}(t)\kappa_1,\ \ e^{\frac{t}{\rho_1}}\cdot x_1(0) - \text{step}(t)\kappa_1\right\}. \tag{5.17}$$

From (5.17) we recognize two extreme cases. The first one is when the state $x_1(0) \approx 0$ and the control superimpose with the growth term. The other extreme case occurs when the state is initially at its very maximum value $x_1(0) = \pm b_{\boldsymbol{x}}$, and therefore the control must reduce the magnitude of the state at a rate greater than the equivalent growth rate. It remains to determine, β_1, T, κ_1 such that a bounded AS state can be maintained for a worst-case input signal and initial AS state. To better illustrate the conditions presented next, Figure 5.3 depicts the growth term, remainder term, and the maximum state value as a function of time t. Specifically, all possible state trajectories are indicated by the shaded area contained within $x_1^{\max}(t)$ and $x_1^{\min}(t)$, the maximal and minimum state value $x_1(t)$ respectively at time t. Additionally, we assume, without loss of generality, the initial state to be positive, i.e. $x_1(0) \in [0, b_{\boldsymbol{x}}]$. Furthermore, for illustrative purposes both the "positive", $G_1(t), R_1(t)$ and "negative", $G_1^{(-)}(t), R_1^{(-)}(t)$ are included. The negative remainder term corresponds to the case where control contribution and signal superimpose and the initial state is zero.

From Figure 5.3 we recognize several intuitive results. Firstly, as previously mentioned, if κ_1 is chosen so small that the growth rate exceeds the decay of the remainder rate, the DC cannot maintain a bounded state. This is exemplified in Figure 5.3a, Figure 5.3d, and Figure 5.3g where the AS state bound is immediately exceeded for an initial state $x(0) = b_{\boldsymbol{x}}$. In contrast, for a control gain κ_1 exceeding the growth rate, overpowering the growth rate is no problem. However, the time until the superposition of input and control exceed the bound is reduced, and thereby this choice requires a shorter control period T to maintain a bounded state. This is exemplified in Figure 5.3c, Figure 5.3f, and Figure 5.3i. Finally, we notice that the effect of having a stable system (negative feedback $\rho_1 < 0$), reduces the growth term and thereby relaxes the necessary control, Figure 5.3b, as compared to the marginally stable case Figure 5.3e. Similarly, an unstable system (positive feedback $\rho_1 > 0$), has the opposite effect as can be seen from Figure 5.3h.

The previous discussion determined when a given set of parameters β_1,



(a) $\rho_1 = -\frac{1}{10}$, $\kappa_1 = \frac{1}{2}$ **(b)** $\rho_1 = -\frac{1}{10}$, $\kappa_1 = 1$ **(c)** $\rho_1 = -\frac{1}{10}$, $\kappa_1 = \frac{3}{2}$

(d) $\rho_1 = 0$, $\kappa_1 = \frac{1}{2}$ **(e)** $\rho_1 = 0$, $\kappa_1 = 1$ **(f)** $\rho_1 = 0$, $\kappa_1 = \frac{3}{2}$

(g) $\rho_1 = \frac{1}{10}$, $\kappa_1 = \frac{1}{2}$ **(h)** $\rho_1 = \frac{1}{10}$, $\kappa_1 = 1$ **(i)** $\rho_1 = \frac{1}{10}$, $\kappa_1 = \frac{3}{2}$

$G_1(t)$, $G_1^{(-)}(t)$, $R_1(t)$, $R_1^{(-)}(t)$, $x_1^{\max}(t)$, $x_1^{\min}(t)$

Figure 5.3: The state vector trajectories for permissible input and initial state configurations. Note that for all these figures we plot $x(t)/b_{\boldsymbol{x}}$ on the y-axis against time on the x-axis.

κ_1, and ρ_1 result in a bounded output, and if so, which control period T that would be necessary. A bounded output means that the input of the subsequent node will have a bounded input. Subsequently, β_ℓ, κ_ℓ, and ρ_ℓ can recursively be set and thereby ensure all states to be bounded. This approach results in an upper bound to the global growth and remainder term from (4.21) since we have locally upper-bounded each node separately.

Conditions for Effective Control

For the chain-of-integrators AS the conditions for an effective control can be summarized as

$$G_\ell(t) \leq R_\ell(t) \tag{5.18}$$

$$G_\ell(T) + R_\ell(T) \leq b_{\boldsymbol{x}} \tag{5.19}$$

for any $\ell \in [1, \ldots, N]$ and $t \in (0, T]$.

Specifically, for pure integrators, i.e., $\rho_\ell = 0$, the conditions result in

$$|\kappa_\ell| \geq b_{\boldsymbol{x}} \tag{5.20}$$

$$T\beta_\ell \left(|\kappa_\ell| + b_{\boldsymbol{x}}\right) \leq b_{\boldsymbol{x}}. \tag{5.21}$$

Operating at the border of stability is achieved when having equality in both of these two expressions. Equivalently, this is the same as having the largest permissible β_ℓ and T pair. In this case, their relation can be summarized as

$$\beta_\ell = \frac{1}{2T}, \;\; \kappa_\ell \; = \; b_{\boldsymbol{x}}. \tag{5.22}$$

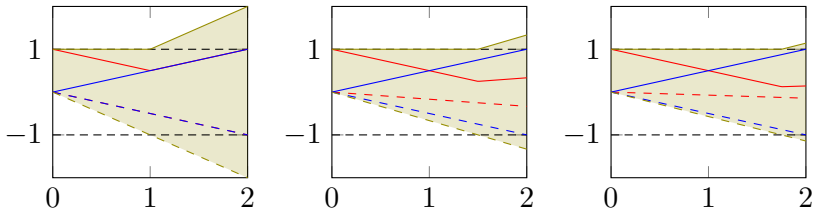
In a hardware implementation, it might not always be preferred to operate at the very border of stability as in (5.22). Instead, the T or β_ℓ might be slightly scaled-down such that we have a margin to the stability bound. To describe this behavior, we define the stability margin as

$$\epsilon \triangleq \max\{\epsilon_1, \ldots, \epsilon_N\} \tag{5.23}$$

where the local stability margin ϵ_ℓ is defined as

$$\epsilon_\ell \triangleq \frac{1}{\beta_\ell T} \geq 2. \tag{5.24}$$

The equality in the previous expressions corresponds to operating at the border of stability or equivalently, having no stability margin.



(a) $Q = 1$ **(b)** $Q = 2$ **(c)** $Q = 3$

$G_1(t)$, $G_1^{(-)}(t)$, $R_1(t)$, $R_1^{(-)}(t)$, $x_1^{\max}(t)$, $x_1^{\min}(t)$

Figure 5.4: Higher-order quantizers example where the control period can be extended as the number of bits Q used in the quantizer increases. Furthermore, $\rho = 0$ and $\kappa = 1$ and the axes of the figures are as in Figure 5.3.

In the case of positive or negative feedback, $\rho_\ell \neq 0$, (5.21) changes as

$$\frac{\beta_\ell}{\rho_\ell}\left(e^{\frac{T}{\rho_\ell}} - 1\right)(|\kappa_\ell| + b_{\boldsymbol{x}}) \leq b_{\boldsymbol{x}} \tag{5.25}$$

cf. (5.15). Subsequently, the related conditions can be adapted accordingly.

Higher-Order Quantizers

Increasing the number of bits in the quantizer, i.e., improving the quality of the control observation, enables us to decrease the stability margin ϵ_ℓ further. Specifically, the remainder term from (5.17) decreases for a Q bit quantizer as

$$R_\ell(t) = \min_{q \in \mathcal{Q}} \max_{x_1(0) \in [-b_{\boldsymbol{x}}, b_{\boldsymbol{x}}]} \left\{ 2^{(1-Q)}\text{step}(t)\kappa_1,\ \ e^{\frac{t}{\rho_1}} \cdot x_1(0) - \frac{q}{2^{Q-1}}\text{step}(t)\kappa_1 \right\} \tag{5.26}$$

where $\mathcal{Q} = \{1, \ldots, 2^{Q-1}\}$ are the positive levels of the corresponding DAC and $\rho = 0$. The new remainder term, combined with the growth term, is shown in Figure 5.4. From this figure, it is clear that, for the same growth rate, the control period can be substantially extended as we increase the number of bits Q in the quantizer and DAC.

Furthermore, for a higher-order quantizer, the conditions from (5.21) can be adapted as

$$T\beta_\ell \left(2^{(1-Q)}|\kappa_\ell| + b_{\boldsymbol{x}}\right) \leq b_{\boldsymbol{x}}. \tag{5.27}$$

In other words, the remainder term is kept smaller at the end of the control period T, which means that we can increase the relative amplification without violating the bound. This can be visualized by the minimum stability margin as

$$\epsilon_{\min} = 1 + 2^{(1-Q)} \tag{5.28}$$

and where, for $Q \to \infty$, $\epsilon_{\min} \to 1$.

5.3.2 Switched Capacitor Control

Switched capacitor circuits are popular for discrete-time $\Delta\Sigma$ modulators. One of their attractions is that the precision at which the ratio of two capacitors can be realized in CMOS technology can be advantageous [22].

The switched capacitor is a discrete-time concept. However, that does not mean that it cannot be used by the DC in the control-bounded ADC. For example, using the circuit from Figure 5.5 as the DAC of

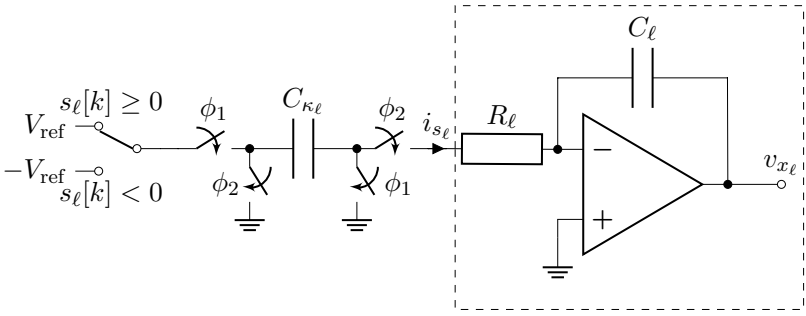


Figure 5.5: One-bit switched capacitor DAC where ϕ_1 and ϕ_2 are two clock phases that makeup one switch capacitor clock period. The dashed box symbolizes an integrator implemented using an operational amplifier and is thus part of the AS of the ADC.

the local DC is an alternative to the default square DAC waveform.

Specifically, the switched capacitor is operated with some time period $T_{\text{SC}} = T_{\phi_1} + T_{\phi_2} = T/\Phi$ where Φ is a positive integer. Furthermore, the capacitor C_{κ_ℓ} gets charged with a positive or negative voltage V_{ref}, depending on $s_\ell[k]$, during the first phase ϕ_1 and then discharged through R_ℓ onto C_ℓ during the second phase ϕ_2. In a switched capacitor circuit, the resistor R_ℓ would be as small as possible such that the charge is moved from C_{κ_ℓ} almost instantaneously. Ultimately, it is the opamp's speed that sets the fundamental limit on how quickly charge can be moved. Nevertheless, the R_ℓ provides a parameter to gradually control of the equivalent DAC's output shape. This offers additional flexibility as decreasing the switched-capacitor time period, or equivalently increasing Φ, a larger control feedback signal can be sustained. The previous statement assumes that ϕ_1, together with the switch resistance, to be such that the capacitor gets sufficiently charged during ϕ_1. In the other extreme case, for $\phi_1 \to 0$, $C_{\kappa_\ell} \to \infty$, and $\Phi = 1$ we approach the default square DAC waveform.

The proposed switched capacitor control requires adaptations of the local control conditions, from (5.18) and (5.19), as (5.15) will now change due to the new DAC waveform. Specifically,

$$d_{\text{SC}}(t) = -\sum_{\xi=0}^{T/T_{\text{sc}}-1} e^{-\frac{t-\xi T_{\text{sc}}-T_{\phi_1}}{R_\ell C_{\kappa_\ell}}} \cdot d_{\text{sq}}(t - \xi T_{\text{sc}} - T_{\phi_1}) \tag{5.29}$$

$$\kappa_\ell = V_{\text{ref}} \tag{5.30}$$

$$\beta_\ell = \frac{1}{R_\ell C_\ell} \tag{5.31}$$

where

$$d_{\text{sq}}(t) = \begin{cases} 1 & \text{if } t \in [0, T_{\phi_2}) \\ 0 & \text{otherwise} \end{cases} \tag{5.32}$$

is a unit step lasting the length of the second phase ϕ_2. Note that the digital estimator changes as (4.63) and (4.64) depends on the DAC waveform through $\boldsymbol{D}(t)$.

Interestingly, these changes only apply to the offline computations in the DE. Therefore, both the DC and the DE's operational complexity remains unchanged when using a switched capacitor DC.

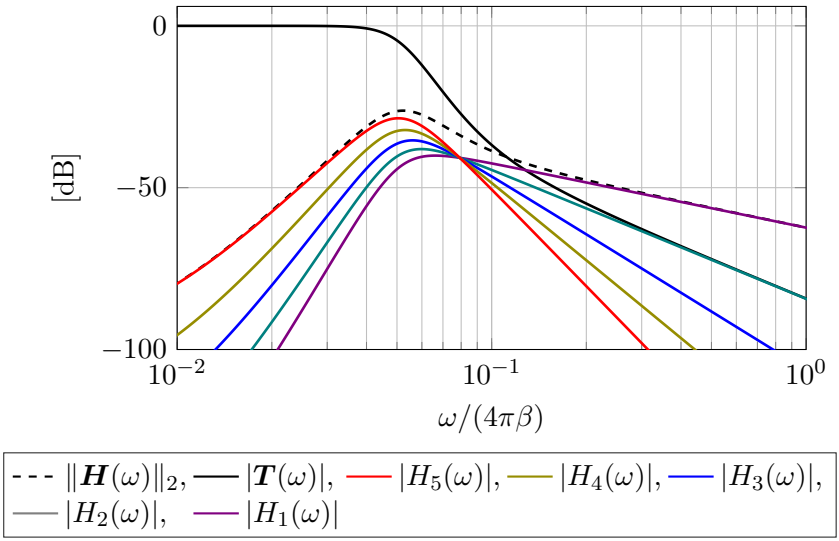


Figure 5.6: STF and NTF of a fifth order, $N = 5$, chain-of-integrators ADC

5.4 Digital Estimator

We remind ourselves that the DE is determined by the choice of AS and DC as was outlined in Section 4.3. In particular, the NTF and STF of the chain-of-integrators ADC follows from the expressions in (4.47) and (4.48) where we have plugged in the norm of the ATF matrix from (5.11). Figure 5.6 shows the STF and NTF as a function of frequency for a system parameterized as $\beta_1 = \cdots = \beta_N = \beta$ and $\rho_1 = \cdots = \rho = 0$. From the figure, we see that for this configuration, the DE filter mostly relies on the last output via the transfer function $H_5(\omega)$. Interestingly, as the frequency increase, the other states will contribute more and more to the estimate. At the unit-gain frequency of the filter, (5.12), all five signal observations contribute equally in magnitude to the estimate.

5.4.1 White Noise Analysis

The expected performance of a chain-of-integrators ADC can be approximated in a similar way as for $\Delta\Sigma$ modulators [8]. Specifically, we follow the steps in Section 4.4, and assume the signal observation $\boldsymbol{y}(t)$, (4.24),

to be bandlimited and have a white spectrum within the frequency band of interest as in (4.91). For the following analysis we assume a pure integrator chain $\rho_1 = \cdots = \rho_N = 0$ and each node in the chain to be parameterized equally as $\beta_1 = \cdots = \beta_N = \beta$ and $\kappa_1 = \cdots = \kappa_N = \kappa$. Expanding the approximated squared conversion error from (4.94) with the specific AS norm we can write

$$\boldsymbol{P}_{\boldsymbol{y}|\mathcal{B}} \approx \frac{\sigma^2_{\boldsymbol{y}|\mathcal{B}}}{2\pi} \int_{\mathcal{B}} \frac{\omega^{2N}}{\beta^{2N}} \, \mathrm{d}\omega \tag{5.33}$$

$$= \frac{\sigma^2_{\boldsymbol{y}|\mathcal{B}}}{2\pi} \cdot \frac{2}{2N+1} \cdot \beta^{-2N} \omega_{\mathcal{B}}^{2N+1} \tag{5.34}$$

where $\omega_{\mathcal{B}} = 2\pi f_{\mathcal{B}}$ corresponds to the frequency band of interest expressed in radians per second. We recognize the OSR for the control-bounded ADC as

$$\mathrm{OSR} = \frac{1}{2T f_{\mathcal{B}}}. \tag{5.35}$$

Notice that strictly speaking, the term OSR might appear misleading as we have made a point of not considering the control signals $s[k]$ as samples. Regardless, we use this terminology as it is well established in the $\Delta\Sigma$ community and have the same functional meaning for control-bounded ADC.

Using the OSR definition together with the stability margin from (5.23) we can rewrite (5.34) as

$$\boldsymbol{P}_{\boldsymbol{y}|\mathcal{B}} \approx \frac{\sigma^2_{\boldsymbol{y}|\mathcal{B}}}{T} \cdot \frac{1}{2N+1} \cdot (\epsilon\pi)^{2N} \cdot (\mathrm{OSR})^{-2N-1}. \tag{5.36}$$

Furthermore, by approximating

$$\sigma^2_{\boldsymbol{y}|\mathcal{B}} \approx \alpha T \frac{(2b_{\boldsymbol{x}})^2}{12} \tag{5.37}$$

we can write the SNR, for a full-scale sinusoidal input signal with amplitude A as

$$\mathrm{SNR} \approx \alpha^{-1} \cdot \frac{3A^2}{2b_{\boldsymbol{x}}^2} \cdot (2N+1) \cdot (\epsilon\pi)^{-2N} \cdot (\mathrm{OSR})^{2N+1}. \tag{5.38}$$

The SNR approximation in (5.38) is almost identical to the one of a N-th order $\Delta\Sigma$ as shown in Section 3.4.4. The two expressions only differ by the stability margin $\epsilon \geq 2$.

Furthermore, notice the importance and possible reward of keeping a tightly controlled bound, as shown in the approximation in (5.37). Additionally, the T factor in (5.37) accounts for the fact that the AS state vector is dominated by the control contributions $\boldsymbol{\Gamma}\boldsymbol{s}(t)$. Furthermore, since $\mathrm{E}\left[s_\ell(t)^2\right] = 1$, for any ℓ, the PSD of $\boldsymbol{s}(t)$ must scale with T. The same applies to the PSD of the signal observation $\boldsymbol{y}(t)$. Finally, the scale factor $\alpha > 0$ accounts for the discrepancy between the assumed uniform probability density on $\boldsymbol{y}(t)$ and the actual one. α can be determined through simulation.

5.4.2 Closing the Gap to Delta-Sigma Modulation

As previously mentioned, there is a ϵ^{-2N} discrepancy between the SNR approximation for control-bounded ADC and the one for conventional $\Delta\Sigma$ modulators. This does not mean that one is inferior to the other but can instead be remedied in two different ways.

Firstly, increasing the number of bits used in the quantizer has already been established to decrease ϵ towards one, i.e., leveling the two SNR expressions. Furthermore, as the remainder term shrinks so does the bound $b_{\boldsymbol{x}}$ in (5.37). From simulations, we have established roughly a 6 dB improvement per increased bit, which is the expected outcome in an equivalent $\Delta\Sigma$ modulator.

Secondly, the local DC used for the chain-of-integrators ADC is derived in a very restrictive way, i.e., we have bounded the output for the worst of all possible adversarial input signals imaginable. Thus it might be possible to venture beyond these bounds and thereby giving up the stability guarantee. This approach would require extensive simulations to ensure stability for any given set of input signals. However, this is the standard approach for most high order $\Delta\Sigma$ modulators as they operate without any stability guarantees.

5.4.3 Single vs. Multi-Output Analog System

We previously established that there exists two different modes for the DE of the chain-of-integrators ADC. Namely, considering all the state of the AS to be outputs of the system, using (5.7) or alternatively to only consider the last state $x_N(t)$ to be the output, using (5.6). A comparison of the STF and NTF for each mode is given in Figure 5.7. From the figure, we see that the multi-output AS approach outperforms the single-output

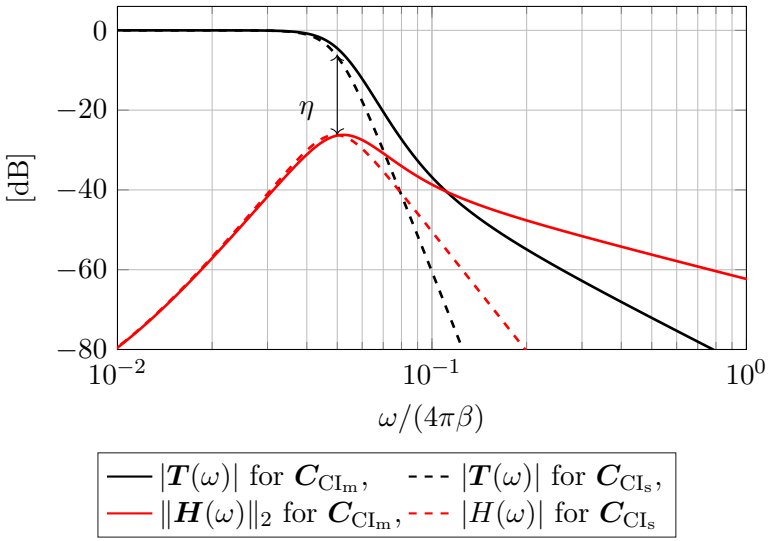


Figure 5.7: Comparison of STF and NTF for single output ($\boldsymbol{C}_{\mathrm{CI_s}}$), and multiple output ($\boldsymbol{C}_{\mathrm{CI_m}}$), reconstruction.

AS one. The SNR stretches over a broader frequency band of interest, and the NTF is lower for the same bandwidth. For a large OSR the difference diminishes.

5.4.4 Spline Basis Signal Processing

An intriguing observation is that for the chain-of-integrators AS and DC, the resulting control contribution $\boldsymbol{q}(t)$ seen at the signal observation, see (4.23), can be written as a weighted sum of B-splines. To see this, we first recognize that the square DAC waveform is a zero-order B-spline [30]. Furthermore, integrating a square DAC waveform for a control period T results in another B-spline with an order that is increased by one. Therefore, the contribution, originating from the ℓ-th local control in the chain to the final output, is a $(N-\ell)$-th order B-spline weighted by $s_\ell[k]$. The same applies to the other outputs of the AS. In other words, the proposed local DC uses B-spline waveforms to control the states of the AS.

The significance of this insight becomes clear when we consider some post-

processing filtering step to the samples of the estimate $\hat{u}(t)$. Specifically, as the DE is a linear filter, we can reverse the order of the DE and any post-filtering operation. However, if we now describe this post-processing filter by a B-spline basis function and coefficients as proposed in [30], the continuous-time filtering operation amounts to discrete-time filtering of the control signal $\boldsymbol{s}[k]$ and the post-filter spline coefficients. Reversing the order of the digital-estimation filter and any post-filter can be computationally beneficial; this is mainly because $\boldsymbol{s}[k]$, as opposed to $\boldsymbol{u}(kT)$, is, for the examples considered in this thesis, a vector containing binary elements. Therefore, the discrete-time filtering, between the control signal and B-spline filter coefficients, results in much simpler arithmetic operations. This could potentially make the post-filtering step very computationally attractive in a hardware implementation.

Note that the B-spline filtering proposed above also requires adaptation to the underlying spline basis, since convolving two B-splines involves computing new coefficients and also increases the B-spline basis order. For a given post-processing B-spline order this can be incorporating into $\boldsymbol{D}(t)$ and thereby the offline computations in Equations (4.63) and (4.64).

5.4.5 Computational Complexity

The computational complexity of the DE was covered in Section 4.3.4 (offline version) and Section 4.3.5 (online version). As the chain-of-integrators ADC only considerers scalar input signals ($L = 1$) and has as many independent DC control paths as AS states (N=M), we can summarize the DE computational complexity as

- $\mathcal{O}(N)$ real-valued scalar multiplications,
- $\mathcal{O}\left(N^2\right)$ real-valued scalar additions,
- and requires N bits and $4N + 2$ real-valued scalar values to be kept in memory

per estimated sample when implemented using the offline batch estimator from Algorithm 3 in Appendix E.

5.5 Simulations

To verify the functionality of the chain-of-integrators ADC we now proceed by conducting a series of simulations. This is done by simulating

the system of ODEs (4.4) for a given input signal $u(t)$, as described in Section 4.8. Subsequently, we reconstruct uniform samples as described in Section 4.3.2 and compute the corresponding PSD of the estimate as covered in Section 3.4.2.

For all simulations in this chapter we will assume a default parameterization $\beta_1 = \cdots = \beta_N = \beta = 10$, $\kappa_1 = \cdots = \kappa_N = \kappa = 1.05$, $\rho_1 = \cdots = \rho_N = \rho = 0$, $b_{\boldsymbol{x}} = 1$, $T = 1/21.5$, and therefore $\epsilon = 2.15$.

The bandwidth of the ADC is determined as in (4.49). For the single output system, $\boldsymbol{C}_{\mathrm{CI_s}}$, and given the equally parameterized nodes, this can be written as

$$\omega_{\mathrm{crit}} = \frac{|\beta|}{\eta^{\frac{1}{N}}} \tag{5.39}$$

using the OSR notation the bandwidth parameter could also directly be written as

$$\eta = \left(\frac{\mathrm{OSR}}{\epsilon\pi}\right)^N. \tag{5.40}$$

Note that these relations do not strictly hold for the multi-output AS case as it will have a slightly larger ATF matrix norm. However, they can still be used as a reasonably good approximation even in the multi-output AS case.

The resulting PSD is given in Figure 5.8. In this simulation, we excite the system with a full-scale sinusoidal input signal. For $N = 1$, the spectrum is heavily influenced by harmonics from the input signal and differs substantially from the white noise assumption, shaped by $\boldsymbol{H}(\omega)$, cf. Section 5.4.1. However, as the number of nodes increases, the spectrum flattens, making the mentioned white noise analysis more applicable. This is also confirmed by the corresponding SNR plot shown in Figure 5.9. From the figure, it seems that $\alpha = 1$ is too large as the simulations outperform the expected performance for $N > 1$. This indicates that the signal observation vector $\boldsymbol{y}(t)$ cannot have a flat frequency response in the frequency band of interest as assumed in the white noise analysis. Regardless the correct α can be determined from the simulations.

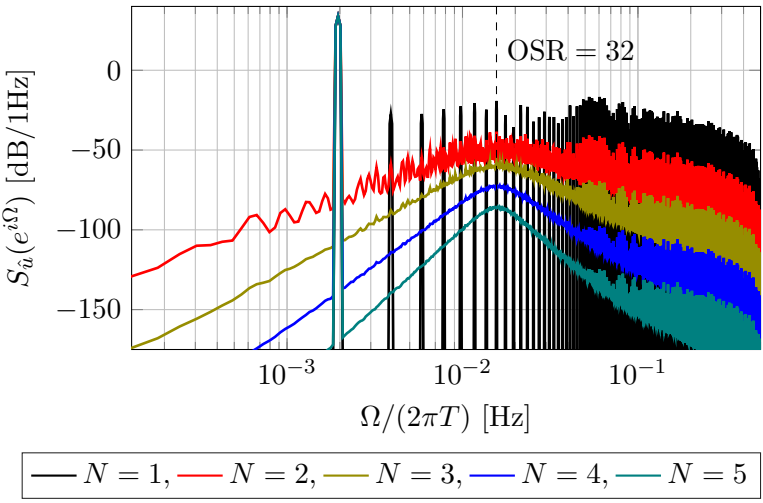


Figure 5.8: PSD of the estimate $\hat{u}(kT)$, see (4.55), for a chain-of-integrators ADC as the number of nodes is increased from one to five.

Note that the scaling in the x-axis in Figure 5.9 refers to decibel full scale (dBFS) with respect to the input signal, meaning 0 dB corresponds to a full scale input sinusodial signal.

Additionally, the zero input case, i.e. if the system is without input signal, is shown in Figure 5.10. From the figure, we recognize that, for $N > 1$, the PSD is almost identical except for the input signal peak seen in Figure 5.8. This confirms that the conversion error, which is a linear mapping of the signal observation $\boldsymbol{y}(t)$, see (4.43), is largely independent of the input signal $u(t)$. This can also be seen in the time domain as shown in Figure 5.11. The point of this figure is that it is hard, if not impossible, to distinguish which of the two-state trajectories corresponds to a sinusoidal input signal or a zero input signal.

5.5.1 Fundamental Resource Scaling

As is made clear from Figure 5.8 and Figure 5.9 the nominal conversion performance of the chain-of-integrators ADC is closely related to N, the number of integrators in the chain. Next we summarize what increasing

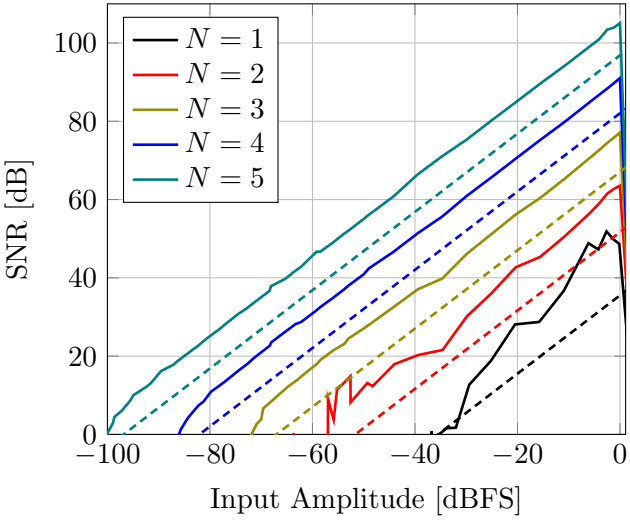


Figure 5.9: SNR for a chain-of-integrators ADC as the number of nodes is increased from one to five, and the input amplitude increases from zero to the full-scale amplitude. The dashed lines correspond to the approximation in (5.38) for the same number of nodes and an $\alpha = 1$.

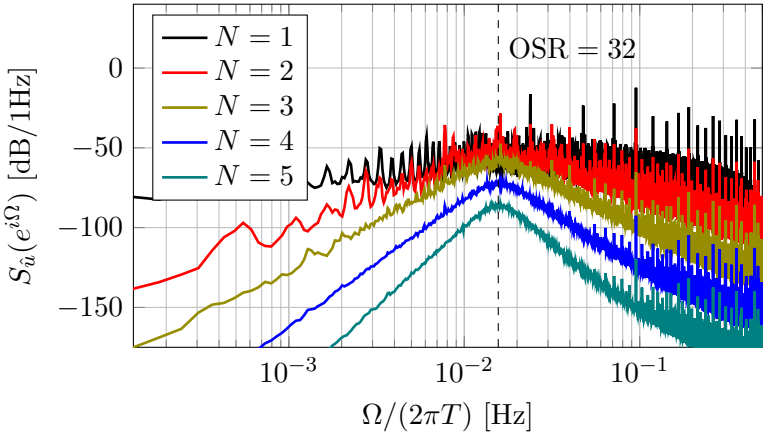


Figure 5.10: Same simulation setup as in Figure 5.8 except $u(t) = 0$.

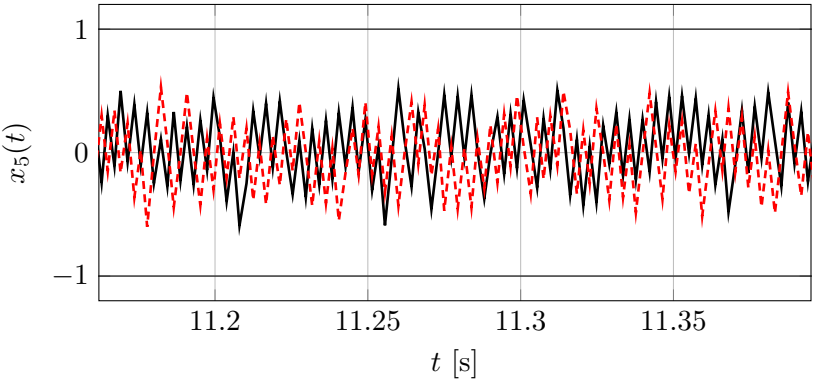


Figure 5.11: A snapshot of the time evolution of the $x_5(t)$ for a control-bounded ADC excited with two different input signals $u(t)$, one of them being $u(t) = 0$ and the other a sinusoidal input signal with significant amplification.

N entails for the AS, DC, and DE. Starting with the AS, as each integrator represents a single analog state, increasing N also increases the number of analog states. Subsequently, also the dimensions of $\boldsymbol{A}_{\mathrm{CI}}$, $\boldsymbol{B}_{\mathrm{CI}}$, and $\boldsymbol{C}_{\mathrm{CI}}$ increase correspondingly as given in (5.2), (5.3), and (5.5). Furthermore, as the chain-of-integrators DC has one local DC path per integrator the number of DC paths also scales with N and thereby also the corresponding dimensions of $\boldsymbol{\Gamma}_{\mathrm{CI}}$ and $\tilde{\boldsymbol{\Gamma}}_{\mathrm{CI}}$, see (5.4). As for the computational complexity of the DE this is outlined in Section 5.4.5 but essentially boils down to a computational complexity where the number of multiplications grows linearly and the number of additions quadratically with N.

5.5.2 Limit Cycles

The chain-of-integrators ADC does however come with some caveats. Arguably, the biggest one is its sensitivity to limit cycles. As an example, Figure 5.12 shows the PSD of a simulation as in Figure 5.10 where the input signal is a constant signal with a small input amplitude as $u(t) = 0.003$. A standard strategy against limit cycles is to use some sort of dithering [12, 23, 24, 28]. This is certainly applicable to the general control-bounded ADC as well.

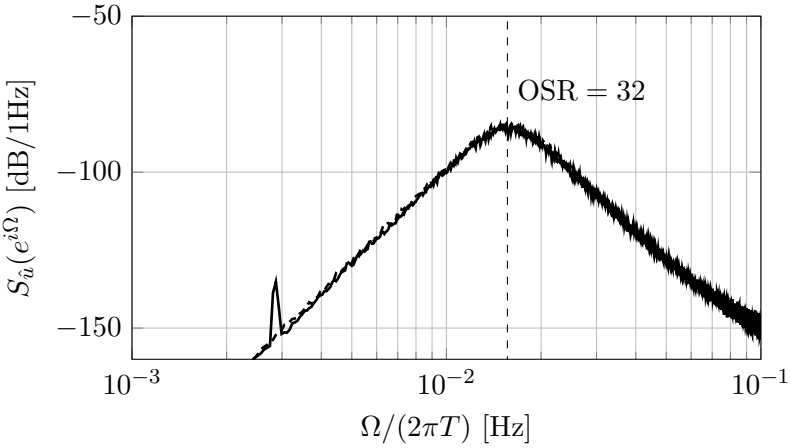


Figure 5.12: PSD of $\hat{u}(kT)$ where the input signal is a constant signal with an offset $u(t) = 0.003$. This signal choice exposes a limit cycle visible at $\Omega/(2\pi T) = 0.003T = 0.0645$. Except for the input signal the simulation parameters are as in Figure 5.8.

However, the general control-bounded ADC offers an additional implicit way of adding a dithering effect without actually adding noise to the estimate $\hat{u}(t)$. The dithering effect is achieved by the DC structure shown in Figure 5.13. Specifically, by feeding small contributions of all the controls back to the first stage, we effectively randomize the control signals. This method relies on the effective randomness of the control signals for large N and obviates the need for an additional source of randomness. Extensive simulations (as exemplified in Figure 5.12) have shown this method to be highly effective. Note that the augmented system as in Figure 5.13 still fits into the general scheme of Figure 4.1. In particular, the new feedback signals are known to the digital estimation filter, which can remove their effect on the analog signals. Note that, this cancellation is implicit since the control input matrix $\mathbf{\Gamma}$ changes for the DE computations.

When implementing this method, it should be noted that the additional feedback increases the required stability margin of the first stage. However, this increase is minor as the additional feedback can be quite small and yet enforce a dithering effect.

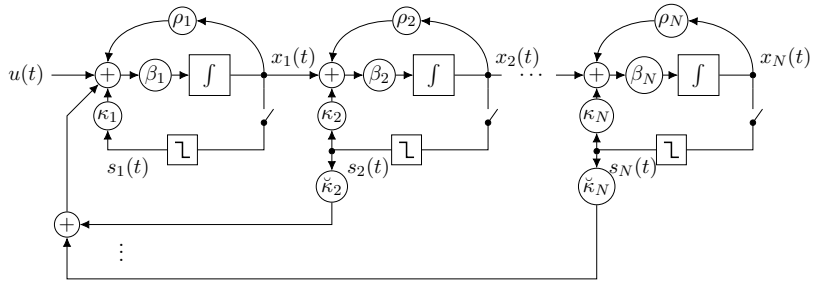


Figure 5.13: The chain-of-integrators where each analog state is connected to the next in a chain and the DC is local to each state.

In principle, a single $\breve{\kappa}_n \neq 0$ would suffice for creating a dithering effect. However, simulations indicate that a better effective randomness is achieved for many such feedback paths. Furthermore, this concept can be further expanded by introducing additional feedforward and feedback paths among the control signals as

$$\boldsymbol{\Gamma} = \begin{pmatrix} \kappa_1 & & \\ & \ddots & \\ & & \kappa_N \end{pmatrix} + \begin{pmatrix} \breve{\kappa}_{1,1} & \cdots & \breve{\kappa}_{1,N} \\ \vdots & \ddots & \vdots \\ \breve{\kappa}_{N,1} & \cdots & \breve{\kappa}_{N,N} \end{pmatrix} \tag{5.41}$$

where $\breve{\kappa}_{k,\ell}$ represents the dithering control paths.

5.5.3 Mismatch

The proposed dithering technique also turns out to mitigate the effects of mismatch. Figure 5.14 shows a mismatch simulation where the chain-of-integrators ADC as in Figure 5.1 is shown with solid lines and the dithered control version, Figure 5.13 in dashed lines. Furthermore, the black lines correspond to a 2% variation from the nominal values in the elements of the control matrix $\boldsymbol{\Gamma}$. The red lines correspond to a 2% variation in the elements of the system matrix $\boldsymbol{A}$. Both these simulations are for a zero input test signal $u(t) = 0$. From Figure 5.14, we see that the dithering mechanism spreads the errors over the spectrum avoiding disturbing peaks and ensuring a better overall SNR.

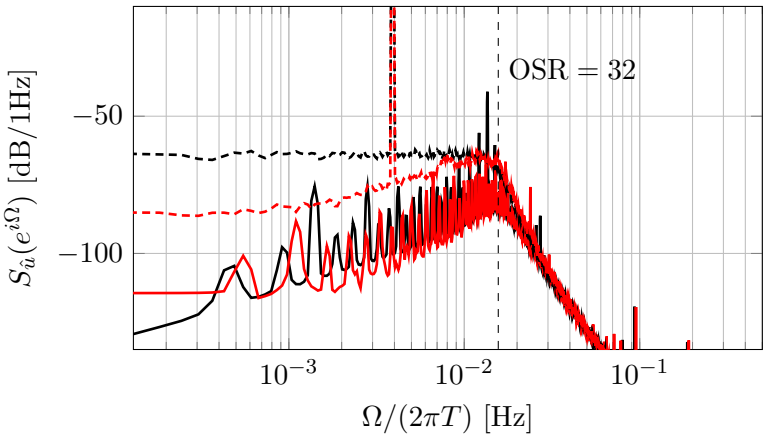


Figure 5.14: PSD of $\hat{u}(kT)$ for a mismatch simulations where the test input signal is $u(t) = 0$. Where the red lines correspond to a 2% mismatch in the elements of $\boldsymbol{\Gamma}$ and the black lines 2% mismatch in the elements of $\boldsymbol{A}$. The dashed lines correspond to the chain-of-integrators ADC using the dither feedback from Figure 5.13, and the solid lines correspond to the regular version as in Figure 5.1.

5.5.4 Comparison to MASH Converters

The chain-of-integrators ADC is not a MASH $\Delta\Sigma$ converter, as in Section 3.5. This can be seen as $s_1(t)$ and $u(t)$ enters the system at the same sum node and therefore have the same transfer function to each signal observation. In other words, the general cancellation condition from (4.128) will be overdetermined for any $N > 1$; for the chain-of-integrator ADC, perceiving $s_1[k], \ldots, s_N[k]$ as sampled and quantized versions of the input signal, will be limiting. This means that from the conventional viewpoint, using a generalized digital-cancellation logic, the chain-of-integrators ADC would, never outperform a first-order $\Delta\Sigma$ modulator.

Regardless, the structural similarities of the MASH $\Delta\Sigma$ and chain-of-integrators ADC makes their comparison interesting. In the following simulation we compare the fifth order, $N = 5$, chain-of-integrators ADC as in Figure 5.1 with a 1-1-1-1-1 discrete-time MASH $\Delta\Sigma$ modulator. The MASH $\Delta\Sigma$ is simulated using the python library [33], which in turn derives from the Schreier's MATLAB toolbox. Comparing to a

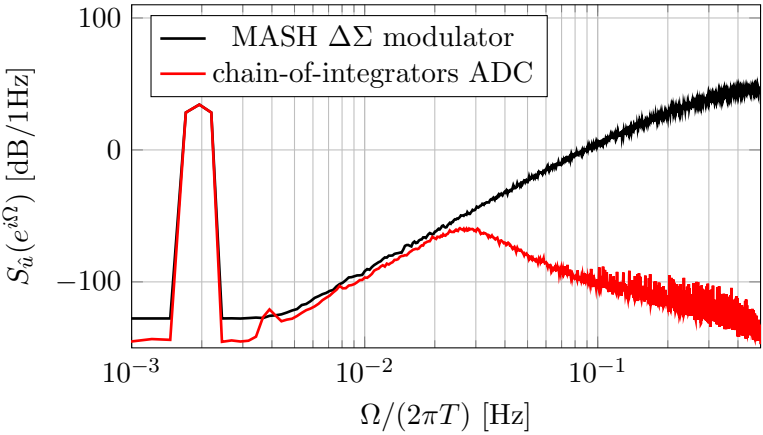


Figure 5.15: PSD of $\hat{u}(kT)$ comparison between a MASH $\Delta\Sigma$ modulator and the chain-of-integrators ADC.

discrete-time MASH $\Delta\Sigma$, and not a continuous-time, is motivated by the fact that the continuous-time versions are typically designed to approximate discrete-time versions. Therefore, nominal performance from a discrete-time $\Delta\Sigma$ modulator should function as a "best case" scenario of a continuous-time version. To make the comparison fair, the discrete-time system parameters are derived from the AS of the chain-of-oscillators nodes using the concepts from [26]. In this simulation we normalize the PSD such that equivalently $T = 1$. Note that the stability margin $\epsilon = 2$ is unchanged compared to previous simulations. The resulting comparison is given in Figure 5.15. From the figure we see that the 1-1-1-1-1 MASH $\Delta\Sigma$ almost performs equally to that of the fifth order chain-of-integrators ADC. Therefore, we can conclude that the chain-of-integrators ADC nominally performs similarly to its MASH $\Delta\Sigma$ modulator counterpart.

5.6 Hardware Implementation

To further prove the basic functionality of the chain-of-integrators ADC a hardware prototype was built[1]. The implementation follows the proposed architecture of Figure 5.13 where each node of the chain-of-integrators

[1]The hardware prototype was constructed by Jonas Biveroni and Patrik Strebel.

was realized using an inverting amplifier op-amp design as in Figure 5.16.

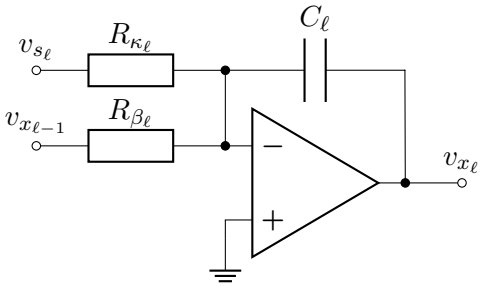


Figure 5.16: A single node of the chain-of-integrators AS.

Note that the hardware prototype was not designed to excel in terms of speed or accuracy, nor power consumption. Instead, it was meant as a proof of concept for the control-bounded conversion principle. Figure 5.17 shows a photo of the described prototype.

5.6.1 Results

Some measurements results of the hardware prototype are shown in Figure 5.18 and Figure 5.19. These measurements were conducted using a sinusoidal input signal with a frequency of 72.4 Hz. Figure 5.19 shows the PSD of the estimated input signal that generated the largest SNR measurement from Figure 5.18. The results show a spurious-free dynamic range (SFDR) of approximately 83 dB as well as a maximal SNDR of 74.5 dB.

5.6.2 Parametrization

The hardware prototype was made with $N = 5$ identically parameterized nodes. The integrators were realized with an operational amplifier (AD8615). The control period was $T = 54\mu$s. The nominal value of the capacitor C was 10 nF and the nominal value of both resistors (R_β and $R_{\kappa\beta}$) were 16 kΩ, resulting in $\beta = 1/(R_\beta C) = 6250$/sec and $\kappa = 1.25$. For the first stage, the feedback contributions are $R_{\kappa_{1,2}\beta} = \ldots = R_{\kappa_{1,5}\beta} = 64\,\text{k}\Omega$. The operating voltage was 5V, but all signals were confined to the range $0 \ldots 2.5$ V; "zero" in Figure 5.13 translates to $V_0 = 1.25$ V. The resistors

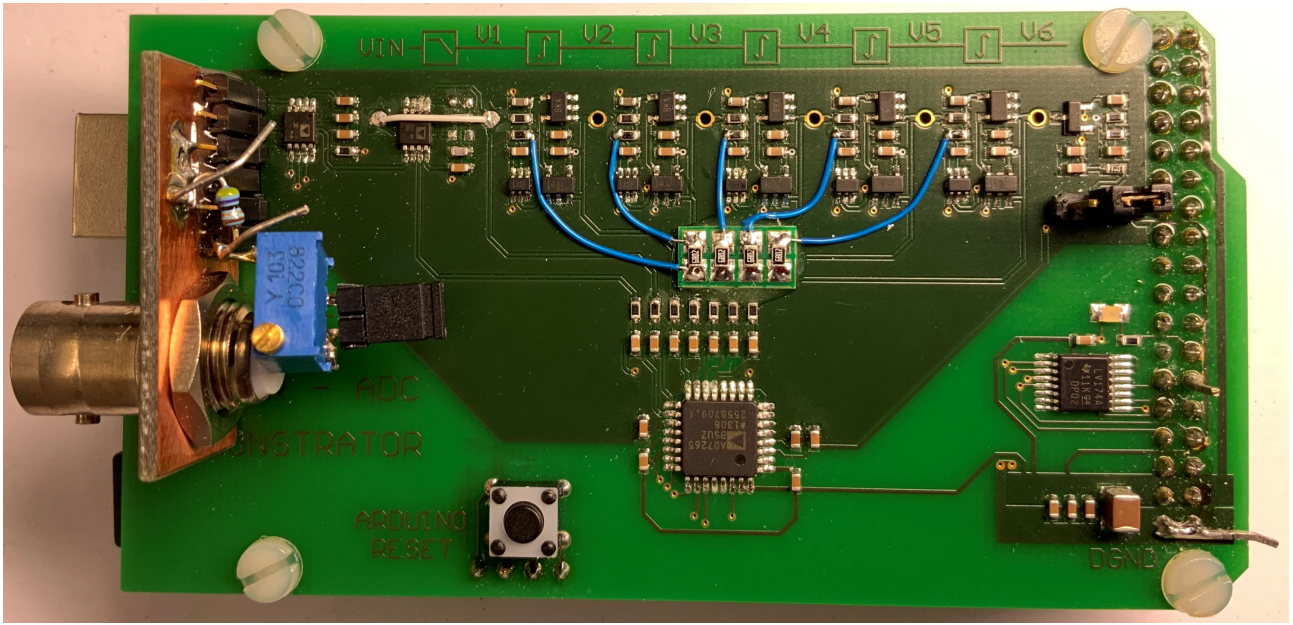


Figure 5.17: The hardware prototype showing the printed circuit board with discrete components piggybacked on an Arduino board.

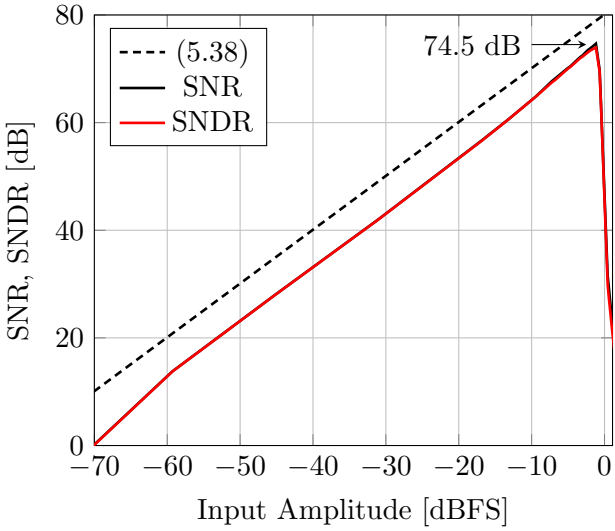


Figure 5.18: SNR for different input amplitudes of the hardware prototype. The solid black line corresponds to the SNR and the red one to the SNDR respectively (lines virtually coincides). Furthermore, the dashed black line is the analytical expression from (5.38) for $\alpha = 1$.

and capacitors are standard surface-mount devices with 1% tolerance; they were not preselected, and their actual values were not measured.

The control contribution $s_\ell(t)$ (i.e., the voltage v_{s_ℓ} in Figure 5.16) is generated from v_{x_ℓ} using a separate threshold circuit (TLV3201) and a separate analog switch (TS5A9411). The whole circuitry is realized on a printed circuit board, which is piggybacked on an Arduino board.

For the empirical results shown in Figure 5.18 and Figure 5.19, the DE works with nominal values of β and κ; neither the hardware prototype nor the digital filter uses any calibration or adjustment for actual (rather than nominal) values.

The parameter η of the digital filter is set according to (5.40) with $\mathrm{OSR} = 32$ and $\epsilon = 2$.

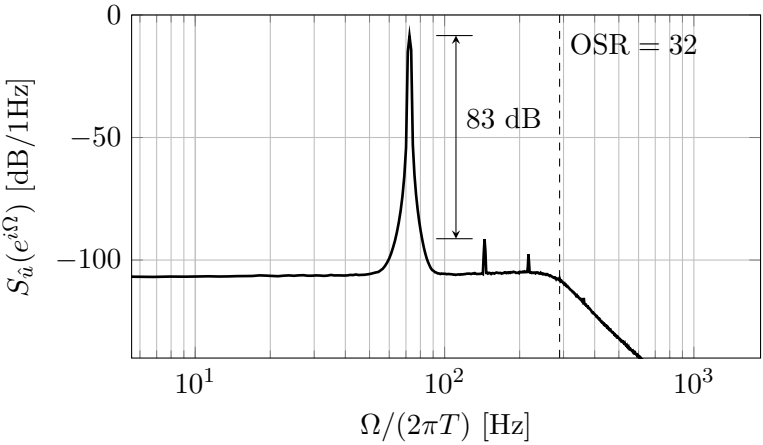


Figure 5.19: PSD of the estimate $\hat{u}(kT)$ for the hardware prototype. The input signal corresponds to the one that had the largest measured SNR in Figure 5.18.

5.6.3 Influence of Mismatch & Thermal Noise

Next, we derive the specific mismatch and thermal noise transfer functions for the given hardware prototype. In the following analysis, we refer to the single output reconstruction mode. Mismatch or thermal noise originating from the ℓ-th node in the chain to the signal observation of the chain-of-integrators ADC, has the transfer function

$$G_\ell(\omega) = \psi \cdot \left(\frac{\beta}{i\omega}\right)^{n-\ell-1} \tag{5.42}$$

to the signal observation of the chain-of-integrators ADC. In the given expression $\psi = 1 - \frac{\bar{\beta}}{\beta}$ or $\psi = 1 - \frac{\bar{\kappa}}{\kappa}$ respectively, depending if the mismatch is in β or κ. Furthermore, $\bar{\beta}$ and $\bar{\kappa}$ denote the nominal values while β and κ are the actual values of the circuit prototype. Additionally, in the thermal noise case $\psi = 1$. Furthermore, the product of (5.42), and the NTF of the digital estimator, constitute the conversion error contribution from a specific mismatch or thermal noise source.

As previously stated, the nodes of the hardware implementation are all dimensioned equally. Therefore, the prototype will be primarily sensitive to errors introduced in the first nodes of the chain. Using the analysis

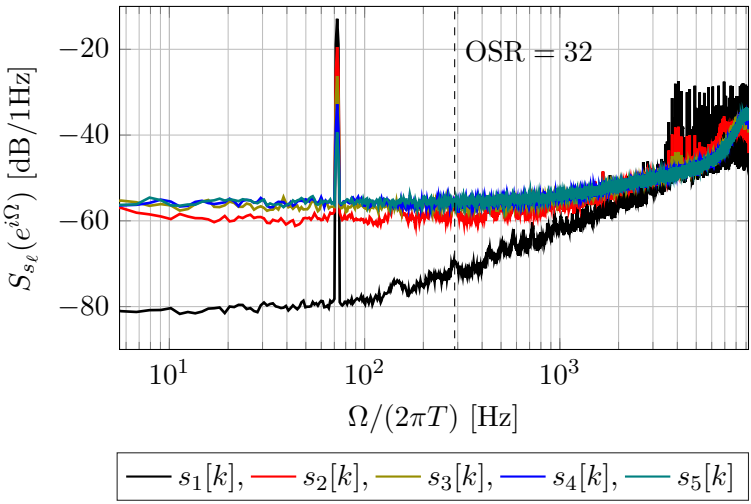


Figure 5.20: PSD of the control signals for hardware full-scale input test signal.

from Section 4.6, at room temperature, the thermal noise caused by the resistors, see Figure 5.16, should cause a noise floor in $\hat{u}(t)$ at roughly -157 dB/1Hz. From Figure 5.19, we can determine that thermal noise is not the dominating error source of the hardware prototype as the noise floor is significantly larger than -157 dB/1Hz.

Instead, the prototype is most likely limited by component mismatch. In principle the analysis of Section 4.6 applies, but the PSD of the control signals, shown in Figure 5.20, contradicts the white noise assumption. These control signals are the result of the full-scale input test signal as in Figure 5.19.

Interestingly, instead of having a flat spectrum, the control signals, and in particular $s_1[k]$, are very favorably shaped as its distribution tends to be more concentrated at higher frequencies outside the band of interest. This observation has been observed, empirically, to generalize for other types of input signals as well.

Chapter 6

Leapfrog Analog-to-Digital Converter

The leapfrog ADC is a control-bounded ADC, much like the chain-of-integrators from Chapter 5, with a particular analog feedback structure. The feedback enables complex conjugate pole pairs in the AS transfer function and can therefore provide both more amplification as well as a sharper transition between passband and stopband compared with the chain-of-integrators AS.

6.1 General Structure

The general structure of the AS and DC is given in Figure 6.1. From the figure, we notice a local DC identical to that from Section 5.3. We also notice a chain-of-integrators structure where each node is connected by the scalar amplification factors $\beta_1, \ldots, \beta_N$. The feedback structure, defined by $\rho_1, \ldots, \rho_N$, connects the last analog state $x_N(t)$, and any intermediate state, with the first $x_1(t)$ via a series of feedback loops. The given structure is also known as a leapfrog filter, which has given this control-bounded ADC its name. There are two features associated with the proposed AS. Firstly, the feedback structure enables complex pole

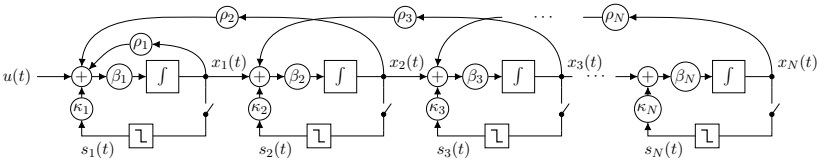


Figure 6.1: The AS and DC of a leapfrog ADC. Notably, the leapfrog ADC has the same DC as the chain-of-integrators ADC in Figure 5.1. The AS has a leapfrog type feedback structure which also warrants its name.

pairs, which can enhance the amplification in the signal band of interest. Secondly, the leapfrog filter's transfer function are known to have low sensitivity to component mismatch. Note that the mentioned mismatch insensitivity applies to the overall AS and not necessarily the component mismatch between the DE and AS as was analyzed in Section 4.6.2.

6.2 Analog System

The dynamical system of a leapfrog ADC, as in Figure 6.1, can be described by the system of ODEs

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{\mathrm{LF}}\boldsymbol{x}(t) + \boldsymbol{B}_{\mathrm{LF}}\boldsymbol{u}(t) + \boldsymbol{\Gamma}_{\mathrm{LF}}\boldsymbol{s}(t) \tag{6.1}$$

where the state transition matrix is

$$\boldsymbol{A}_{\mathrm{LF}} = \begin{pmatrix} \rho_1 & \rho_2 & & & \\ \beta_2 & 0 & \rho_3 & & \\ & \beta_3 & 0 & \ddots & \\ & & \ddots & \ddots & \rho_N \\ & & & \beta_N & 0 \end{pmatrix}. \tag{6.2}$$

Furthermore, the input vector $\boldsymbol{B}_{\mathrm{LF}}$, the control input matrix $\boldsymbol{\Gamma}_{\mathrm{LF}}$, the control observation matrix $\tilde{\boldsymbol{\Gamma}}_{\mathrm{LF}}^{\mathsf{T}}$, and the signal observation matrix $\boldsymbol{C}_{\mathrm{LF}}^{\mathsf{T}}$ are all of the same form and parametrization as the chain-of-integrators ADC, cf. (5.3)-(5.7). For the remainder of this chapter we will only consider the single-output reconstruction, i.e. $\boldsymbol{C}_{\mathrm{LF}}^{\mathsf{T}}$ as in (5.6). However, the material presented below also apply to the multi-output AS mode. The reason for using the single-output AS mode is that the resulting analytical expressions, more clearly, highlight the fundamental functionality. A consequence is that all ATF matrices in this chapter will be scalars.

6.2.1 Transfer Function Analysis

The given state space representation results in an ATF matrix which can be written as

$$G_{\mathrm{LF}}(\omega) = \boldsymbol{C}_{\mathrm{LF}}^{\mathsf{T}} \left(i\omega \boldsymbol{I}_N - \boldsymbol{A}_{\mathrm{LF}}\right)^{-1} \boldsymbol{B}_{\mathrm{LF}} \tag{6.3}$$

$$= \boldsymbol{C}_{\mathrm{LF}}^{\mathsf{T}} \frac{\operatorname{adj}\left(i\omega \boldsymbol{I}_N - \boldsymbol{A}_{\mathrm{LF}}\right)}{\det\left(i\omega \boldsymbol{I}_N - \boldsymbol{A}_{\mathrm{LF}}\right)} \boldsymbol{B}_{\mathrm{LF}} \tag{6.4}$$

where $\operatorname{adj}\left(i\omega \boldsymbol{I}_N - \boldsymbol{A}_{\mathrm{LF}}\right) \in \mathbb{C}^{N\times N}$ computes the adjugate matrix of the inverse which is the same as the transpose of the cofactor matrix and $\det\left(i\omega \boldsymbol{I}_N - \boldsymbol{A}_{\mathrm{LF}}\right) \in \mathbb{C}$ is the determinant of the same inverse.

Interestingly, the structure of the $\boldsymbol{A}_{\mathrm{LF}}$ matrix is such that the determinant, or equivalently the poles of the system, can be written recursively as

$$\det\left(i\omega \boldsymbol{I}_N - \boldsymbol{A}_{\mathrm{LF}}\right) = p_N(i\omega) \tag{6.5}$$

$$= i\omega \cdot p_{(N-1)}(i\omega) - \beta_N \cdot \rho_N \cdot p_{(N-2)}(i\omega) \tag{6.6}$$

where

$$p_0(i\omega) = 1 \tag{6.7}$$

$$p_1(i\omega) = i\omega - \rho_1. \tag{6.8}$$

For the single-output AS estimator mode, only the $(N,1)$-th element of the adjugate matrix need to be computed. Subsequently, the ATF matrix expression from (6.4) can be written as

$$G_{\mathrm{LF}}(\omega) = \frac{\prod_{\ell=1}^{N} \beta_\ell}{p_N(i\omega)}. \tag{6.9}$$

The transfer function matrix from (6.9) has the same nominator as the corresponding chain-of-integrators (5.10), the denominator on the other hand, for parameter choices $\beta_\ell \rho_\ell < 0$, can have complex pole pairs.

Introducing Zeros in the Nominator

For greater flexibility we might consider adding zeros to the ATF matrix in (6.9). One way to achieve this is by altering the input matrix as

$$\boldsymbol{B}_{\mathrm{LF}} = \begin{pmatrix} \beta_1, & \tilde{\beta}_2, & \ldots, & \tilde{\beta}_N \end{pmatrix}^{\mathsf{T}} \tag{6.10}$$

which results in the ATF matrix

$$G_{\mathrm{LF}}(\omega) = \frac{\prod_{j=1}^{N} \beta_j + \sum_{k=2}^{N} \tilde{\beta}_k p_{k-1}(i\omega) \prod_{\ell=k+1}^{N} \beta_\ell}{p_N(i\omega)}. \tag{6.11}$$

This gives the possibility to implement any general N-th order transfer function polynomial

$$G_{\mathrm{LF}}(\omega) = \frac{\sum_{\ell=0}^{N} b_\ell (i\omega)^{N-\ell}}{(i\omega)^N + \sum_{k=1}^{N} a_k s^{N-k}} \tag{6.12}$$

by the parameter choices $\beta_1, \ldots, \beta_N$, $\tilde{\beta}_2, \ldots, \tilde{\beta}_N$, and $\rho_1, \ldots, \rho_N$. Notice that, as discussed in Section 4.5, the AS performance is determined by $\|G_{\mathrm{LF}}(\omega)\|_2$ and it is therefore not only a unit gain filter but also a substantial gain filter, which is of interest when designing (6.12).

Loop Filter Topologies

For higher-order $\Delta\Sigma$ modulators, and a given target loop filter transfer function, there are multiple ways of realizing the actual filter topology. Examples are cascade-of-integrator with feedforward structure (CIFF), cascade-of-integrator with feedback structure (CIFB), cascade-of-resonators with feedforward structure (CRFF), and cascade-of-resonators with feedback structure (CRFB) [25]. All these variations demonstrate different ways of constructing excellent ASs and, when combined with a local DC, would most likely make interesting alternatives to the leapfrog AS proposed in this chapter. However, this falls outside the scope of this thesis.

6.2.2 A Special Case

A particular interesting case of the leapfrog ADC is when

$$\beta_1 = \sqrt{\beta} \cdot \omega_{p/2} \tag{6.13}$$

$$\rho_1 = 0 \tag{6.14}$$

$$\rho_2 = -\frac{1}{\sqrt{\beta}} \cdot \omega_{p/2} \tag{6.15}$$

$\beta_2 = \cdots = \beta_N = \beta_1$, $\rho_3 = \cdots = \rho_N = \rho_2$, and $\tilde{\beta}_2 = \ldots, \tilde{\beta}_N = 0$. Specifically, for this parametrization, the poles of the system are defined

by

$$p_N(i\omega) = \prod_{k=1}^{N} \left(i\omega - i2\omega_{p/2} \cos\left(\frac{k\pi}{N+1} \right) \right). \tag{6.16}$$

The particular pole pattern results in a ATF matrix as shown in Figure 6.2 where $N = 10$, $\beta = 1$, $\omega_{p/2} = \pi/8$. From the figure we see that the ATF

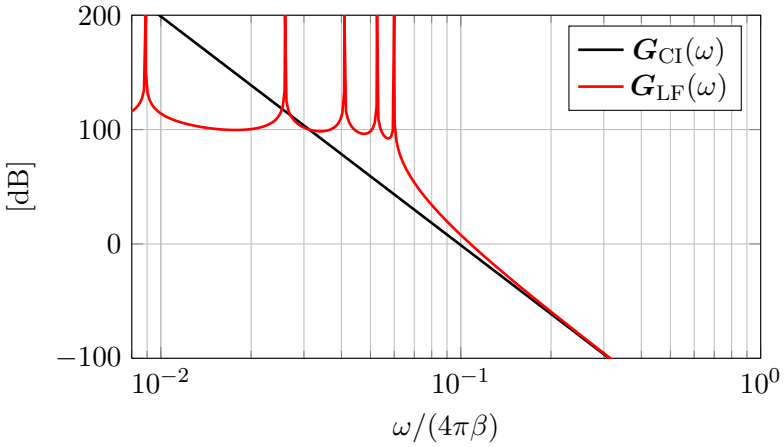


Figure 6.2: ATF matrix for a tenth order (N=10) chain-of-integrators ADC and leapfrog ADC respectively. As the poles are spread over the frequency band of interest for the leapfrog ADC, it has a larger bandwidth for the same amplification.

expression has a larger bandwidth at the expense of a more constant amplification in the frequency band of interest compared to the chain-of-integrators ADC. Additionally, the poles are visible from the transfer function as spikes in the passband. Note that since the poles only have an imaginary part, the AS amplification at the frequency corresponding to the poles will have an infinite amplification.

6.3 Digital Estimator

The greater bandwidth of the leapfrog ADC can have a dramatic effect on the STF and NTF of the reconstruction filter. To see this we use the AS and parameterization presented in Section 6.2.2. Furthermore, we

consider $\omega_p/(4\pi) = 1/16$ as the cutoff frequency of the leapfrog's ATF expression, see (6.16), and thereby determine

$$\eta^2_{\mathrm{LF}} = \|G_{\mathrm{LF}}(\omega_p)\|_2^2. \tag{6.17}$$

$$\eta^2_{\mathrm{CI}} = \|G_{\mathrm{CI}}(\omega_p)\|_2^2. \tag{6.18}$$

For comparison we use a chain-of-integrators ADC with the same N and β.

Using the given parameter settings the two STF and NTF are shown in Figure 6.3. The figure reveals a massive reduction of the leapfrog's

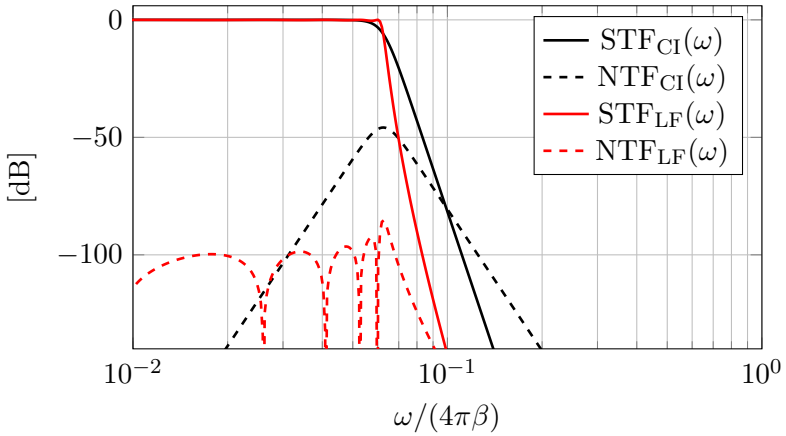


Figure 6.3: Comparison of the STF and NTF for a leapfrog vs a chain-of-integrators ADC given the same parametrization. The leapfrog ADC has a significantly lower NTF at the cutoff frequency with the expense of a flat overall NTF in the frequency band of interest and ripples in the STF.

NTF's magnitude close to the cutoff frequency compared to the chain-of-integrators. The leapfrog NTF also has a more constant magnitude at roughly -100 dB for the frequency band of interest.

Ripples in the STF Passband

Figure 6.3 also shows another difference between the two ASs; the leapfrog STF has ripples in the passband. Figure 6.4 shows a different scaling of the y-axis of the mentioned STFs where the ripples are clearly visible.

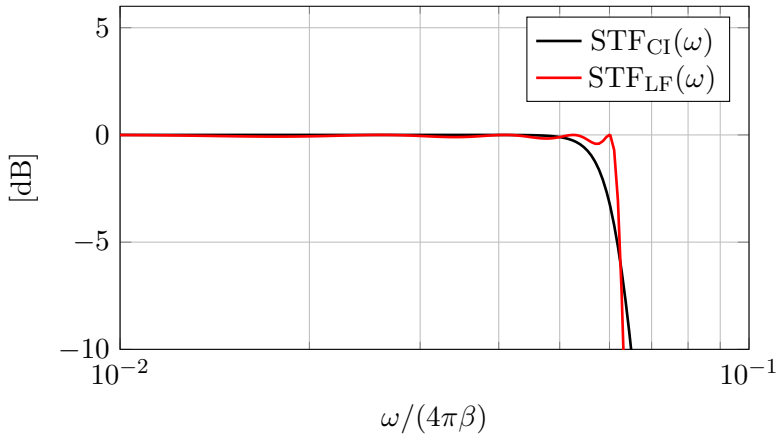


Figure 6.4: The leapfrog ADC can have ripples in the passband STF.

The reason for these ripples can be determined directly from (4.48) and when considering the $\|G_{\mathrm{LF}}(\omega)\|_2^2$ as shown in Figure 6.2. Specifically, as $\|G_{\mathrm{LF}}(\omega)\|_2^2$ is not monotonically increasing with ω, the influence of η^2 in the denominator of (4.48) becomes visible.

The ripples can be suppressed by choosing $\eta^2 \ll \|G_{\mathrm{LF}}(\omega_p)\|_2^2$. An example is given in Figure 6.5 and Figure 6.6, where the STF and NTF is shown for two identical leapfrog systems with an digital estimator parameterized with different η^2.

As is shown in Figure 6.5, suppressing the passband ripples in the STF by lowering the η^2 additionally extends the bandwidth of the estimate. This means that additional post-filtering might be necessary to suppress any unwanted out-of-band signal that now could appear (the green dashed line in Figure 6.6 in the estimate.

Computational Complexity

As the leapfrog ADC only differs from the chain-of-integrators ADC in terms of the AS parameters $\boldsymbol{A}_{\mathrm{LF}}$ and $\boldsymbol{C}_{\mathrm{LF}}^{\mathsf{T}}$ the computational complexity of these two DEs are identical, see Section 5.4.5.

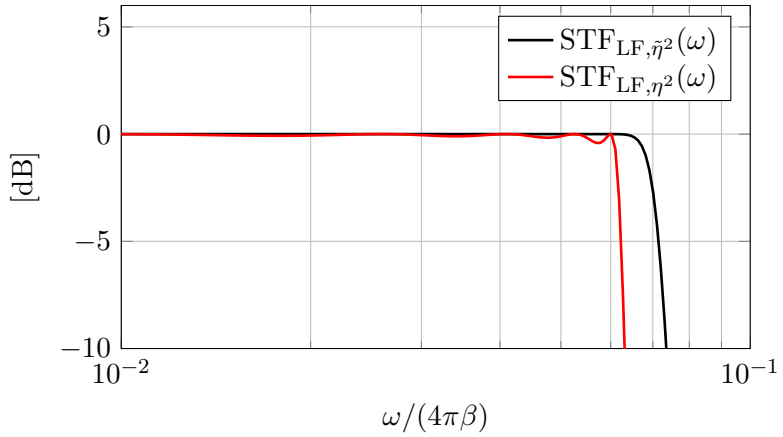


Figure 6.5: Leapfrog ADC STF for two different η^2 parameterizations where $\mathrm{STF}_{\mathrm{LF},\eta^2}$ has an η^2 as in (6.17) and $\mathrm{STF}_{\mathrm{LF},\tilde{\eta}^2}$ with $\tilde{\eta^2} = 10^{-3}\eta^2$.

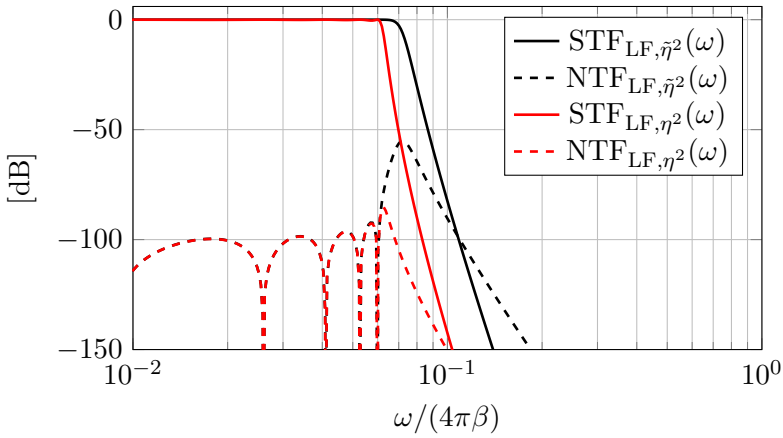


Figure 6.6: Same as in Figure 6.5 with a different y and x-axis scaling.

6.4 Proposed Hardware Implementation

The leapfrog AS using the parametrization from Section 6.2.2 clearly resembles the transmission line model with additional amplification. Next, we propose a hardware implementation of the leapfrog AS and DC inspired by such a lossless transmission line model. The amplification will be done using transconductance amplifiers. This will also allow us to circumvent the need for coils. A gm-C circuit implementation is shown in Figure 6.7 for $N = 5$. From the figure we also see the current going into the ℓ-th capacitor as i_{x_ℓ} and similarly v_{x_ℓ} as the voltage drop over the same capacitor. The latter will symbolize the states of the AS. Specifically, we see that

$$i_{x_\ell}(t) = \mathrm{gm}_\beta \cdot v_{x_{\ell-1}}(t) - \mathrm{gm}_\rho \cdot v_{x_{\ell+1}}(t) - \mathrm{gm}_\kappa \cdot s_\ell(t) \tag{6.19}$$

$$\dot{v}_{x_\ell}(t) = \frac{1}{C} i_{x_\ell} \tag{6.20}$$

where $v_{x_0}(t) = u(t)$ and the last output's current is given by

$$i_{x_N}(t) = \mathrm{gm}_\beta \cdot v_{x_{N-1}}(t) - \mathrm{gm}_\kappa \cdot s_N(t). \tag{6.21}$$

The corresponding state space parameters can then be written as

$$\boldsymbol{A}_{\mathrm{LF}} = \begin{pmatrix} 0 & -\frac{\mathrm{gm}_\rho}{C} & & \\ \frac{\mathrm{gm}_\beta}{C} & \ddots & \ddots & \\ & \ddots & & -\frac{\mathrm{gm}_\rho}{C} \\ & & \frac{\mathrm{gm}_\beta}{C} & 0 \end{pmatrix}, \tag{6.22}$$

$$\boldsymbol{B}_{\mathrm{LF}} = \begin{pmatrix} \frac{\mathrm{gm}_\beta}{C}, & 0, & \ldots \end{pmatrix}^{\mathsf{T}}, \tag{6.23}$$

$$\boldsymbol{\Gamma}_{\mathrm{LF}} = -\frac{\mathrm{gm}_\kappa}{C} \boldsymbol{I}_N, \tag{6.24}$$

and

$$\tilde{\boldsymbol{\Gamma}}_{\mathrm{LF}}^{\mathsf{T}} = \boldsymbol{I}_N. \tag{6.25}$$

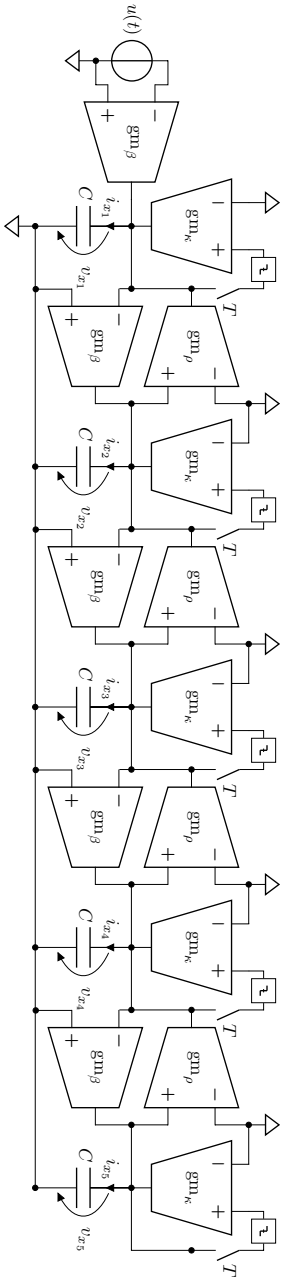


Figure 6.7: A Leapfrog AS implemented using gm-C filters for $N = 5$.

Chapter 7

Chain-of-Oscillators Analog-to-Digital Converter

The chain-of-integrators ADC from Chapter 5 described an A/D conversion principle which was designed to convert baseband analog signals, cf. Figure 5.2. In other words, the frequency band of interest was centered around $f = 0$ Hz. However, there are applications were the signal band of interest is centered around a carrier frequency $f_c \neq 0$. Clearly, for such an application, we could extend the gain β of the AS to also include this off-centered frequency band of interest. Such an increase in amplification requires a reduction in the control period T in order to maintain an effective DC. Specifically, the $\beta \cdot T$ product cannot be scaled unconditionally, as was covered in Section 5.3.1.

An alternative approach is to down-convert the signal, using a modulator, such that the signal is re-centered around $f = 0$ prior to the ADC. There also exist hybrid solutions where the signal is modulated to some intermediate frequency that is more suited for A/D conversion.

The chain-of-oscillators ADC presented next, takes another route; here, the AS amplifies around the f_c, and up- and down-conversion only applies to the DC. There are two main advantages to this approach. Firstly, the DC can be operated with a control period T determined by the

frequency band of interest (as in previous chapters), regardless of where the frequency band is centered. Secondly, as the modulation is not part of the signal path of the converter, it can be implemented with less precision. Note that using modulation as a pre-processing step to the ADC requires the modulator to have superior precision compared to the ADC since any imperfection is directly destructive for the conversion process.

7.1 General Structure

The AS and DC structure of a chain-of-oscillators ADC is given in Figure 7.1. The figure shows a series of analog oscillators consisting of two integrators connected by a specific feedback pattern with the multiplication ω_ℓ. The feedback is such that each oscillator has a resonance frequency of $\omega_\ell/(2\pi)$. Additionally, each node has two scalar inputs that enter the AS state vector via four multiplicative weights $\beta_{\ell,1,1}$, $\beta_{\ell,1,2}$, $\beta_{\ell,2,1}$, and $\beta_{\ell,2,2}$. Each node has two local DCs, with the corresponding control signals $s_{\ell,1}[k]$ and $s_{\ell,2}[k]$, which interact with the integrators through a local modulator. Note that since the state vector and control contribution are grouped in pairs, we introduce the index n as the number of oscillator nodes in the chain-of-oscillators ADC. The total number of AS states can therefore be determined as $N = 2n$.

At this point, it might not be clear that the chain-of-oscillators ADC have similar performance properties as the chain-of-integrators ADC from Chapter 5. To see this, we will need to cover several fundamental properties of oscillators. The general description of the AS and DC will therefore continue in Section 7.3.

7.2 Oscillator Node

We recognize that a single chain-of-oscillators node resembles the concept of a frequency translating $\Delta\Sigma$ modulator, cf. [7,29]. This means that from an implementation standpoint, best practices of the frequency translating $\Delta\Sigma$ modulator can be used. However, since the control-bounded A/D conversion framework focuses on the continuous-time nature of this structure, we must diverge from the traditional explanation model [25].

For a single chain-of-oscillators node ℓ, the corresponding state space

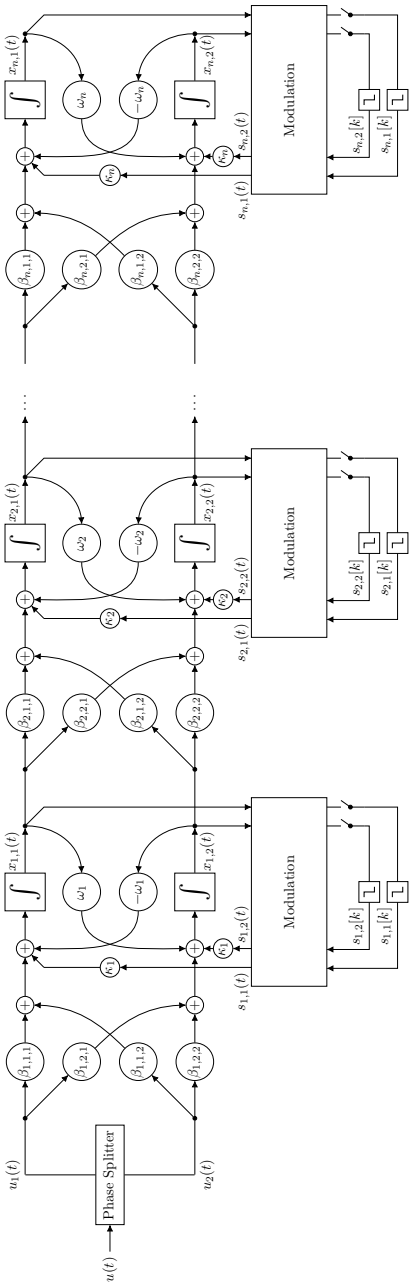


Figure 7.1: The chain-of-oscillators AS and DC where a series of oscillator nodes are connected in a chain, and the DC is local to each oscillator node. The modulation block receding each DC is given in Figure 7.4 and further explained in Section 7.4.

representations follows as

$$\frac{d}{\mathrm{d}t}\begin{pmatrix} x_{\ell,1}(t) \\ x_{\ell,2}(t) \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & -\omega_\ell \\ \omega_\ell & 0 \end{pmatrix}}_{\boldsymbol{A}_\ell} \begin{pmatrix} x_{\ell,1}(t) \\ x_{\ell,2}(t) \end{pmatrix} + \underbrace{\begin{pmatrix} \beta_{\ell,1,1} & \beta_{\ell,1,2} \\ \beta_{\ell,2,1} & \beta_{\ell,2,2} \end{pmatrix}}_{\boldsymbol{B}_\ell} \begin{pmatrix} x_{\ell-1,1}(t) \\ x_{\ell-1,2}(t) \end{pmatrix} + \underbrace{\kappa_\ell \boldsymbol{I}_2}_{\boldsymbol{\Gamma}_\ell} \begin{pmatrix} s_{\ell,1}(t) \\ s_{\ell,2}(t) \end{pmatrix} \tag{7.1}$$

where

$$\begin{pmatrix} x_{0,1}(t) \\ x_{0,2}(t) \end{pmatrix} = \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}. \tag{7.2}$$

Additionally, the parameters of the input matrix are constrained to the form

$$\boldsymbol{B}_\ell \triangleq \beta_\ell \boldsymbol{\Theta}(\phi_{\beta_\ell}) \tag{7.3}$$

where $\phi_{\beta_\ell} \in (0, 2\pi]$, $\beta_\ell \in \mathbb{R}$ is the per node amplification, and $\boldsymbol{\Theta}(\phi)$ is a rotation matrix, see Appendix C.

The control observations matrix is defined as

$$\begin{pmatrix} \tilde{s}_{\ell,1}(t) \\ \tilde{s}_{\ell,2}(t) \end{pmatrix} = \underbrace{\begin{pmatrix} \cos(\omega_\ell t) & \sin(\omega_\ell t) \\ -\sin(\omega_\ell t) & \cos(\omega_\ell t) \end{pmatrix}}_{\tilde{\boldsymbol{\Gamma}}(t)=\boldsymbol{\Theta}(-\omega_\ell t)} \begin{pmatrix} x_{\ell,1}(t) \\ x_{\ell,2}(t) \end{pmatrix}. \tag{7.4}$$

The time-variate structure of the control observations matrix amounts to a demodulation stage and will be further described in Section 7.4

7.2.1 Two-Dimensional Input Signal

A fundamental difference between the chain-of-integrators node and the chain-of-oscillators node is that the latter has a two-dimensional input vector, i.e.,

$$\boldsymbol{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}. \tag{7.5}$$

The two-dimensional nature of the input signal originates from the fact that a general scalar sinusoidal signal can be decomposed as

$$u(t) = a_1(t)\cos(\omega_u t + \phi_u) - a_2(t)\sin(\omega_u t + \phi_u), \tag{7.6}$$

where $a_1, a_2 : \mathbb{R} \to \mathbb{R}$. Furthermore, the largest AS gain is achieved for a two-dimensional input signal where the two terms from (7.6) are separated as

$$\boldsymbol{u}(t) = \begin{pmatrix} u(t) \\ u\left(t - \pi/2\right) \end{pmatrix} \tag{7.7}$$

$$= \begin{pmatrix} a_1(t)\cos(\omega_u t + \phi_u) - a_2(t)\sin(\omega_u t + \phi_u) \\ a_1(t)\cos(\omega_u t + \phi_u - \frac{\pi}{2}) - a_2(t)\sin(\omega_u t + \phi_u - \frac{\pi}{2}) \end{pmatrix}, \tag{7.8}$$

i.e., the signal is paired together with a $\frac{\pi}{2}$ phase-shifted version of itself. Another way of describing this division is by decomposing the signal into an in-phase and quadrature component [11], which are common concepts for frequency modulation in communication system applications.

We refer to this process as phase splitting. The decomposition in (7.8) is ideal and might be difficult to realize in a physical circuit. As an alternative, neglecting the phase splitting by assigning

$$\boldsymbol{u}(t) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t), \tag{7.9}$$

results in an AS amplification reduction. The use of phase splitting will be discussed further in Section 7.2.3.

7.2.2 Amplification Behavior

To see the largest possible amplification of an chain-of-oscillators node, we consider an input signal as in (7.8) where the signal's frequency and the resonance frequency of the node are the same, i.e., $\omega_u = \omega_1$ and $a_1(t) = 1$ and $a_2(t) = 0$.

Feeding the input signal into the oscillator node, and assuming initial states as $x_{1,1}(t_0) = x_{1,2}(t_0) = 0$, results in the states at time $t_1 > t_0$ as

$$\boldsymbol{x}_1(t_1) = \begin{pmatrix} x_{1,1}(t_1) \\ x_{1,2}(t_1) \end{pmatrix} \tag{7.10}$$

$$= \int_{t_0}^{t_1} \exp\left(\boldsymbol{A}_1(t_1 - \tau)\right) \boldsymbol{B}_1 \boldsymbol{u}(\tau)\, \mathrm{d}\tau \tag{7.11}$$

$$= \int_{t_0}^{t_1} \boldsymbol{\Theta}(\omega_1(t_1 - \tau)) \boldsymbol{B}_1 \boldsymbol{u}(\tau)\, \mathrm{d}\tau \tag{7.12}$$

$$= \beta_1 \int_{t_0}^{t_1} \boldsymbol{\Theta}\left(\omega_1(t_1 - \tau) + \phi_{\beta_1}\right) \boldsymbol{u}(\tau)\, \mathrm{d}\tau \tag{7.13}$$

$$= \beta_1 \int_{t_0}^{t_1} \begin{pmatrix} \cos\left(\omega_1(t_1-\tau) + \phi_{\beta_1} + \omega_u\tau + \phi_u\right) \\ \sin\left(\omega_1(t_1-\tau) + \phi_{\beta_1} + \omega_u\tau + \phi_u\right) \end{pmatrix} \mathrm{d}\tau \quad (7.14)$$

$$= \beta_1 \begin{pmatrix} \cos\left(\omega_1 t_1 + \phi_{\beta_1} + \phi_u\right) \\ \sin\left(\omega_1 t_1 + \phi_{\beta_1} + \phi_u\right) \end{pmatrix} \int_{t_0}^{t_1} \mathrm{d}\tau \quad (7.15)$$

$$= (t_1 - t_0) \cdot \boldsymbol{B}_1 \boldsymbol{u}(t_1). \quad (7.16)$$

The derivations above used several properties of rotation matrices. These are covered in Appendix C. Specifically, (7.12) follows from the property (C.1), and (7.13) and (7.14) uses the property from (C.8).

From the resulting expression in (7.16), we see that the oscillator integrates the input signal onto the oscillating states of the oscillator nodes. Similar to the integrator case the $\|\boldsymbol{x}(t)\|_2$ term grows linearly with time for a sinusoidal input signal of the resonance frequency.

Note that for a $a_2(t) \neq 0$ the steps above need to be repeated twice but results in the same expression (7.16).

Sinusodial Input Signal not at a Resonance Frequency

For an input signal as in (7.8), where the signal's frequency and the resonance frequency of the node are not the same, i.e., $\omega_u \neq \omega_1$, $a_1(t) = 1$, and $a_2(t) = 0$, the states at time t_1 can be computed in a similar way. Specifically, the computational steps from (7.11)-(7.13) are identical. Furthermore, we recognize that the order at which rotation matrices are multiplied does not matter as (7.13) becomes

$$\boldsymbol{x}_1(t_1) = \boldsymbol{B}_1 \boldsymbol{\Theta}\left(\omega_1 t_1\right) \int_{t_0}^{t_1} \boldsymbol{\Theta}\left(-\omega_1 \tau\right) \boldsymbol{u}(\tau)\, \mathrm{d}\tau \quad (7.17)$$

$$= \boldsymbol{B}_1 \boldsymbol{\Theta}\left(\omega_1 t_1\right) \int_{t_0}^{t_1} \begin{pmatrix} \cos((\omega_u - \omega_1)\tau + \phi_u) \\ \sin((\omega_u - \omega_1)\tau + \phi_u) \end{pmatrix} \mathrm{d}\tau \quad (7.18)$$

$$= \boldsymbol{B}_1 \boldsymbol{\Theta}(\omega_1 t_1 + \phi_u) \frac{1}{\Delta_\omega} \begin{pmatrix} \sin\left(\Delta_\omega t_1\right) - \sin\left(\Delta_\omega t_0\right) \\ \cos\left(\Delta_\omega t_0\right) - \cos\left(\Delta_\omega t_1\right) \end{pmatrix} \quad (7.19)$$

where $\Delta_\omega \triangleq \omega_u - \omega_1$. Interestingly, as $\Delta_\omega \to 0$, we recover the previous solution (7.16) as

$$\lim_{\Delta_\omega \to 0} \begin{pmatrix} \frac{t_1 \sin(\Delta_\omega t_1)}{t_1 \Delta_\omega} - \frac{t_0 \sin(\Delta_\omega t_0)}{t_0 \Delta_\omega} \\ \frac{\cos(\Delta_\omega t_0) - \cos(\Delta_\omega t_1)}{\Delta_\omega} \end{pmatrix} = (t_1 - t_0) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (7.20)$$

where we have used $\lim_{x\to 0} \frac{\sin(x)}{x} = 1$ and L'Hôpital's rule, such that

$$\lim_{\Delta_\omega \to 0} \frac{\cos(\Delta_\omega t_0) - \cos(\Delta_\omega t_1)}{\Delta_\omega} = \lim_{\Delta_\omega \to 0} t_1 \sin(\Delta_\omega t_1) - t_0 \sin(\Delta_\omega t_0) = 0 \tag{7.21}$$

for a $t_1, t_0 \in \mathbb{R}$. Additionally, as (7.19) is monotonically increasing for both $\Delta_\omega \to 0^+$ and $\Delta_\omega \to 0^-$, we conclude that the largest amplification is given for an input signal of a frequency corresponding to the resonance frequency of the chain-of-oscillators node.

7.2.3 Phase Splitting

As previously mentioned, the scalar input signal $u(t)$ needs to be converted into a vector-valued version $\boldsymbol{u}(t)$ by the process of phase splitting. The name is suggestive since the ideal result would be a decomposition as in (7.8). In the latter, we essentially achieve a $\pi/2$ phase delay between the two elements of the vector for every frequency in the frequency band of interest without altering each element's amplitude.

Before digging deeper into this operation, let us motivate it by considering an input vector with no phase delayed element as in (7.9) and with $a_1(t) = 1$, $a_2(t) = 0$. For this input, (7.17) becomes

$$\boldsymbol{x}_1(t_1) = \boldsymbol{B}_1 \boldsymbol{\Theta}\left(\omega_1 t_1\right) \int_{t_0}^{t_1} \begin{pmatrix} \cos(-\omega_1 \tau) \\ \sin(-\omega_1 \tau) \end{pmatrix} \cos(\omega_u \tau + \phi_u)\, \mathrm{d}\tau \tag{7.22}$$

$$= \boldsymbol{B}_1 \boldsymbol{\Theta}\left(\omega_1 t_1\right) \frac{1}{2} \int_{t_0}^{t_1} \begin{pmatrix} \cos(\Delta_\omega \tau + \phi_u) + \cos(\Delta_{\tilde{\omega}} \tau + \phi_u) \\ \sin(\Delta_\omega \tau + \phi_u) - \sin(\Delta_{\tilde{\omega}} \tau + \phi_u) \end{pmatrix} \mathrm{d}\tau \tag{7.23}$$

where $\Delta_{\tilde{\omega}} = (\omega_1 + \omega_u)$. Evaluating the integral from (7.23) results in

$$\boldsymbol{x}_1(t_1) = \boldsymbol{B}_1 \boldsymbol{\Theta}\left(\omega_1 t_1\right) \left(\frac{1}{2\Delta_\omega} \begin{pmatrix} \sin(\Delta_\omega t_1 + \phi_u) - \sin(\Delta_\omega t_0 + \phi_u) \\ \cos(\Delta_\omega t_0 + \phi_u) - \cos(\Delta_\omega t_1 + \phi_u) \end{pmatrix} + \frac{1}{2\Delta_{\tilde{\omega}}} \begin{pmatrix} \sin(\Delta_{\tilde{\omega}} t_1 + \phi_u) - \sin(\Delta_{\tilde{\omega}} t_0 + \phi_u) \\ \cos(\Delta_{\tilde{\omega}} t_1 + \phi_u) - \cos(\Delta_{\tilde{\omega}} t_0 + \phi_u) \end{pmatrix} \right). \tag{7.24}$$

The first term of (7.24) closely resemble that of (7.19) with the exception of a $\frac{1}{2}$ scaling. Additionally, as $\omega_u \to \omega_1$, we have the same behavior as in (7.19). The second term in (7.24) resides at a much higher frequency $\omega_1 + \omega_u$ and since, for a typical application, $\beta \ll \omega_1$ this term vanishes.

Therefore, to avoid the complex expressions, additional signal terms, and loss of amplification, we advocate not assigning $\boldsymbol{u}(t)$ as in (7.9).

One way of realizing the phase splitting is by filtering the scalar input signal with a first-order high-pass and low-pass filter, with transfer functions as

$$H_{\mathrm{HP}}(\omega) = \frac{i\omega}{\omega_1 + i\omega} \tag{7.25}$$

$$H_{\mathrm{LP}}(\omega) = \frac{\omega_1}{\omega_1 + i\omega} \tag{7.26}$$

respectively. The proposed phase splitter would then result in a input vector

$$\boldsymbol{u}(t) = \begin{pmatrix} \rho_1(\omega) \exp\left(i\left(\frac{\pi}{2} - \arctan\left(\frac{\omega_u}{\omega_1}\right)\right)\right) \\ \rho_2(\omega) \exp\left(-i \arctan\left(\frac{\omega_u}{\omega_1}\right)\right) \end{pmatrix} \tag{7.27}$$

where $\rho_1(\omega), \rho_2(\omega) : \mathbb{R} \to \mathbb{R}$. The expression shows the desired $\frac{\pi}{2}$ phase shift behavior between the two elements of $\boldsymbol{u}(t)$. However, the amplitude response of $\rho_1(\omega)$ and $\rho_2(\omega)$ has a frequency dependent effect on the corresponding amplitudes of each input vector element. Therefore, this essentially only achieves the desired effect of (7.8) in a narrow frequency band centered around ω_1. Better phase splitting can perhaps be obtained by using all-pass filters instead of the high and low-pass filters presented above. In this thesis, this topic will not be pursued further.

7.2.4 Transfer Function Analysis

We will next analyse the ATF matrix of the chain-of-oscillators node. Since $\boldsymbol{A}_1$ is a two-by-two matrix, the matrix inverse in (4.7) has a closed form solution as

$$\boldsymbol{G}_{\mathrm{O}}(\omega) = \boldsymbol{C}_1^{\mathsf{T}} \left(i\omega \boldsymbol{I}_2 - \boldsymbol{A}_1\right)^{-1} \boldsymbol{B}_{\mathbf{1}} \begin{pmatrix} 1 \\ -i \end{pmatrix} \tag{7.28}$$

$$= \boldsymbol{C}_1^{\mathsf{T}} \begin{pmatrix} i\omega & \omega_1 \\ -\omega_1 & i\omega \end{pmatrix}^{-1} \boldsymbol{B}_1 \begin{pmatrix} 1 \\ -i \end{pmatrix} \tag{7.29}$$

$$= \frac{1}{\omega_1^2 - \omega^2} \boldsymbol{C}_1^{\mathsf{T}} \begin{pmatrix} i\omega & -\omega_1 \\ \omega_1 & i\omega \end{pmatrix} \boldsymbol{B}_1 \begin{pmatrix} 1 \\ -i \end{pmatrix} \tag{7.30}$$

$$= \frac{1}{\omega_1^2 - \omega^2} \boldsymbol{C}_1^{\mathsf{T}} \boldsymbol{B}_1 \begin{pmatrix} i\omega & -\omega_1 \\ \omega_1 & i\omega \end{pmatrix} \begin{pmatrix} 1 \\ -i \end{pmatrix} \tag{7.31}$$

where (7.31) follows from (C.18). The $\boldsymbol{C}_1^{\mathsf{T}} = \boldsymbol{I}_2$ has no effect on these expression and will therefore, from here on, be neglected. Furthermore, the right most vector $\begin{pmatrix}1 & -i\end{pmatrix}^{\mathsf{T}}$ corresponds to the ideal phase splitting as in (7.8).

The system amplification can then be written as

$$\begin{aligned}
\|\boldsymbol{G}_{\mathrm{O}}(\omega)\|_2^2 &= \boldsymbol{G}_O(\omega)^{\mathsf{H}}\boldsymbol{G}_O(\omega) && (7.32)\\
&= \frac{\beta_1^2}{(\omega_1^2-\omega^2)^2}\begin{pmatrix}1 & i\end{pmatrix}\begin{pmatrix}-i\omega & \omega_1\\ -\omega_1 & -i\omega\end{pmatrix}\begin{pmatrix}i\omega & -\omega_1\\ \omega_1 & i\omega\end{pmatrix}\begin{pmatrix}1\\ -i\end{pmatrix} && (7.33)\\
&= \frac{\beta_1^2}{(\omega_1^2-\omega^2)^2}\begin{pmatrix}1 & i\end{pmatrix}\begin{pmatrix}\omega_1^2+\omega^2 & i2\omega\omega_1\\ -i2\omega\omega_1 & \omega_1^2+\omega^2\end{pmatrix}\begin{pmatrix}1\\ -i\end{pmatrix} && (7.34)\\
&= 2\beta_1^2\cdot\frac{(\omega_1+\omega)^2}{(\omega_1^2-\omega^2)^2} && (7.35)\\
&= 2\beta_1^2\cdot\frac{(\omega_1+\omega)^2}{((\omega_1-\omega)(\omega_1+\omega))^2} && (7.36)\\
&= \frac{2\beta_1^2}{(\omega_1-\omega)^2} && (7.37)
\end{aligned}$$

where (7.33) follows from $\boldsymbol{B}_1^{\mathsf{T}}\boldsymbol{B}_1 = \beta_1^2\boldsymbol{I}_2$.

Additionally, for the non-phase split input vector as in (7.9) the expression can be modified as

$$\begin{aligned}
\|\boldsymbol{G}_{\mathrm{O_{nps}}}(\omega)\|_2^2 &= \frac{\beta_1^2}{(\omega_1^2-\omega^2)^2}\begin{pmatrix}1 & 0\end{pmatrix}\begin{pmatrix}\omega_1^2+\omega^2 & i2\omega\omega_1\\ -i2\omega\omega_1 & \omega_1^2+\omega^2\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} && (7.38)\\
&= \beta_1^2\cdot\frac{\omega_1^2+\omega^2}{(\omega_1^2-\omega^2)^2}. && (7.39)
\end{aligned}$$

Once more, it can be seen that, in the frequency band of interest, this amounts to a reduction in amplification as

$$\begin{aligned}
\frac{\|\boldsymbol{G}_{\mathrm{O}}(\omega)\|_2^2}{\|\boldsymbol{G}_{\mathrm{O_{nps}}}(\omega)\|_2^2} &= 2\frac{(\omega_1+\omega)^2}{\omega_1^2+\omega^2} && (7.40)\\
&= 2+4\frac{\omega\omega_1}{\omega_1^2+\omega^2}. && (7.41)
\end{aligned}$$

7.3 Analog System

We now consider the actual chain-of-oscillators ADC by considering $n > 1$. The AS is specified in terms of its corresponding state space model

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{\mathrm{CO}}\boldsymbol{x}(t) + \boldsymbol{B}_{\mathrm{CO}}\boldsymbol{u}(t) + \boldsymbol{\Gamma}_{\mathrm{CO}}(t)\boldsymbol{s}(t) \tag{7.42}$$

$$\tilde{\boldsymbol{s}}(t) = \tilde{\boldsymbol{\Gamma}}_{\mathrm{CO}}^{\mathsf{T}}(t)\boldsymbol{x}(t) \tag{7.43}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}_{\mathrm{CO}}^{\mathsf{T}}\boldsymbol{x}(t) \tag{7.44}$$

where the state vector, the control contribution vector, the control observation vector, the signal observation on vector, and the input vector are organized in consecutive pairs as

$$\boldsymbol{x}(t) = \begin{pmatrix} x_{1,1}(t), & x_{1,2}(t), & x_{2,1}(t), & x_{2,2}(t), & \dots \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}^{2n} \tag{7.45}$$

$$\boldsymbol{s}(t) = \begin{pmatrix} s_{1,1}(t), & s_{1,2}(t), & s_{2,1}(t), & s_{2,2}(t), & \dots \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}^{2n} \tag{7.46}$$

$$\tilde{\boldsymbol{s}}(t) = \begin{pmatrix} \tilde{s}_{1,1}(t), & \tilde{s}_{1,2}(t), & \tilde{s}_{2,1}(t), & \tilde{s}_{2,2}(t), & \dots \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}^{2n} \tag{7.47}$$

$$\boldsymbol{y}(t) = \begin{pmatrix} y_{1,1}(t), & y_{1,2}(t), & y_{2,1}(t), & y_{2,2}(t), & \dots \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}^{2n} \tag{7.48}$$

$$\boldsymbol{u}(t) = \begin{pmatrix} u_1(t), & u_2(t) \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}^2 \tag{7.49}$$

Furthermore, the chain structure, as in the chain-of-integrators AS (5.2), is recognized when considering

$$\boldsymbol{A}_{\mathrm{CO}} = \begin{pmatrix} \boldsymbol{A}_1 & & & \\ \boldsymbol{B}_2 & \boldsymbol{A}_2 & & \\ & \ddots & \ddots & \\ & & \boldsymbol{B}_n & \boldsymbol{A}_n \end{pmatrix} \in \mathbb{R}^{2n \times 2n} \tag{7.50}$$

where $\boldsymbol{A}_\ell$ and $\boldsymbol{B}_\ell$ are the local oscillator state specified as in (7.1). Also, the input matrix, the signal observation matrix, the control input matrix, and the control observation matrix are specified as

$$\boldsymbol{B}_{\mathrm{CO}} = \begin{pmatrix} \boldsymbol{B}_1, & \boldsymbol{0}_{2\times 2}, & \dots, & \boldsymbol{0}_{2\times 2} \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}^{2n \times 2} \tag{7.51}$$

$$\boldsymbol{C}_{\mathrm{CO}}^{\mathsf{T}} = \boldsymbol{I}_{2n} \tag{7.52}$$

$$\boldsymbol{\Gamma}_{\mathrm{CO}} = \begin{pmatrix} \kappa_1 \boldsymbol{I}_2 & & \\ & \ddots & \\ & & \kappa_n \boldsymbol{I}_2 \end{pmatrix} \in \mathbb{R}^{2n \times 2n} \tag{7.53}$$

$$\tilde{\boldsymbol{\Gamma}}_{\mathrm{CO}}^{\mathsf{T}}(t) = \begin{pmatrix} \boldsymbol{\Theta}(-\omega_1 t) & & \\ & \ddots & \\ & & \boldsymbol{\Theta}(-\omega_n t) \end{pmatrix} \in \mathbb{R}^{2n \times 2n}. \tag{7.54}$$

The time-dependent control observation matrix $\tilde{\boldsymbol{\Gamma}}_{\mathrm{CO}}(t)$ will be further explained in Section 7.4.1.

Transfer Function Analysis

Due to the chain structure, the transfer function of a chain-of-oscillators follows from multiplying the individual oscillator node transfer functions from Section 7.2.4. Specifically, the transfer function from the input to the ℓ-th node in the chain follows as

$$\boldsymbol{G}_{\mathrm{CO}_\ell}(\omega) = \left(\prod_{k=1}^{\ell} \frac{1}{\omega_k^2 - \omega^2} \boldsymbol{B}_k \begin{pmatrix} i\omega & -\omega_k \\ \omega_k & i\omega \end{pmatrix} \right) \begin{pmatrix} 1 \\ -i \end{pmatrix} \tag{7.55}$$

assuming $\beta_1 = \cdots = \beta_n = \beta$, $\omega_1 = \cdots = \omega_n = \omega_p$, and $\phi_{\beta_1} = \cdots = \phi_{\beta_n} = \phi_\beta$ the expression can be written as

$$\boldsymbol{G}_{\mathrm{CO}_\ell}(\omega) = \left(\frac{\beta}{\omega_p^2 - \omega^2} \right)^{\ell} \boldsymbol{\Theta}\left(\ell\phi_\beta\right) \boldsymbol{Q}\boldsymbol{\Lambda}^{\ell}\boldsymbol{Q}^{-1} \begin{pmatrix} 1 \\ -i \end{pmatrix} \tag{7.56}$$

where we have used the eigendecomposition from (C.17) as

$$\begin{pmatrix} i\omega & -\omega_k \\ \omega_k & i\omega \end{pmatrix} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1} \tag{7.57}$$

and subsequently resulting in

$$\|\boldsymbol{G}_{\mathrm{CO}_\ell}(\omega)\|_2^2 = 2 \left(\frac{\beta}{\omega_p - \omega} \right)^{2\ell}. \tag{7.58}$$

The resulting expression for

$$\|\boldsymbol{G}_{\mathrm{CO}}(\omega)\|_2 = \| \left(\boldsymbol{G}_{\mathrm{CO}_1}(\omega), \quad \ldots, \quad \boldsymbol{G}_{\mathrm{CO}_n}(\omega) \right) \|_2 \tag{7.59}$$

i.e. for $\boldsymbol{C}_{\mathrm{CO}}^{\mathsf{T}} = \boldsymbol{I}_{2n}$, is shown in Figure 7.2 and Figure 7.3. Both figures show the ATF matrix norm for a fifth-order, $n = 5$, chain-of-oscillators ADC. Additionally, the figures confirm a similar frequency behavior as for the chain-of-integrators, cf. Figure 5.2. The main difference being that the ATF norm is centered around $f_c \neq 0$.

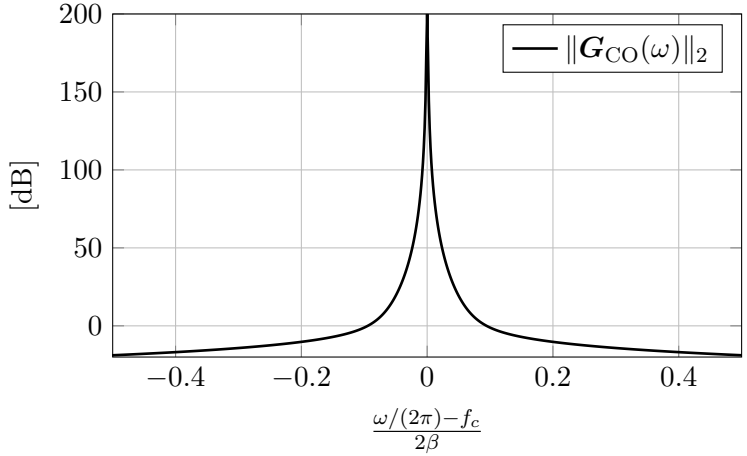


Figure 7.2: The AS amplification as in (7.59) for the chain-of-oscillators ADC. Note that the x-axis is centered around the carrier frequency f_c.

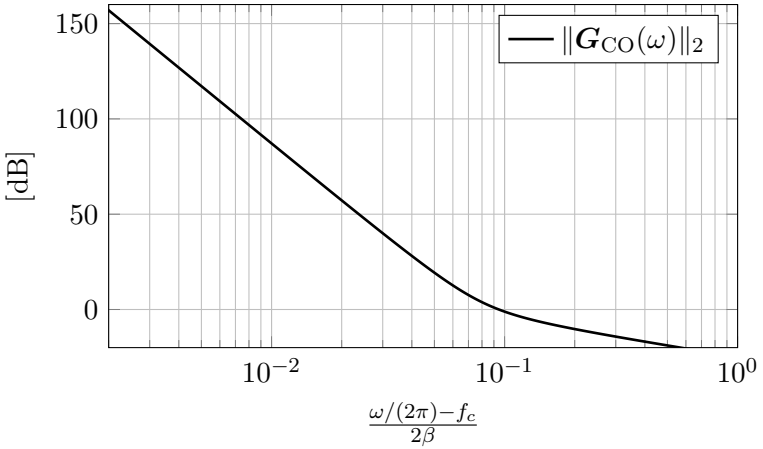


Figure 7.3: Same as in Figure 7.2 but only showing half of the spectrum with an logarithmic x-axis as was done for the chain-of-integrators ADC Figure 5.2.

7.4 Digital Control

The chain-of-oscillators from Figure 7.1 includes both a demodulation step between the AS states $\boldsymbol{x}(t)$ and the control-observations $\tilde{\boldsymbol{s}}(t)$, and a modulation step between the control signal $\boldsymbol{s}[k]$ and the control contribution $\boldsymbol{s}(t)$. Both these operations are done independently for each chain-of-oscillators node. The purpose of the modulation is removing and adding the frequency offset f_c respectively, such that the control task can be done without concern for the actual resonance frequency of the chain-of-oscillators AS node. The described modulator is shown in Figure 7.4.

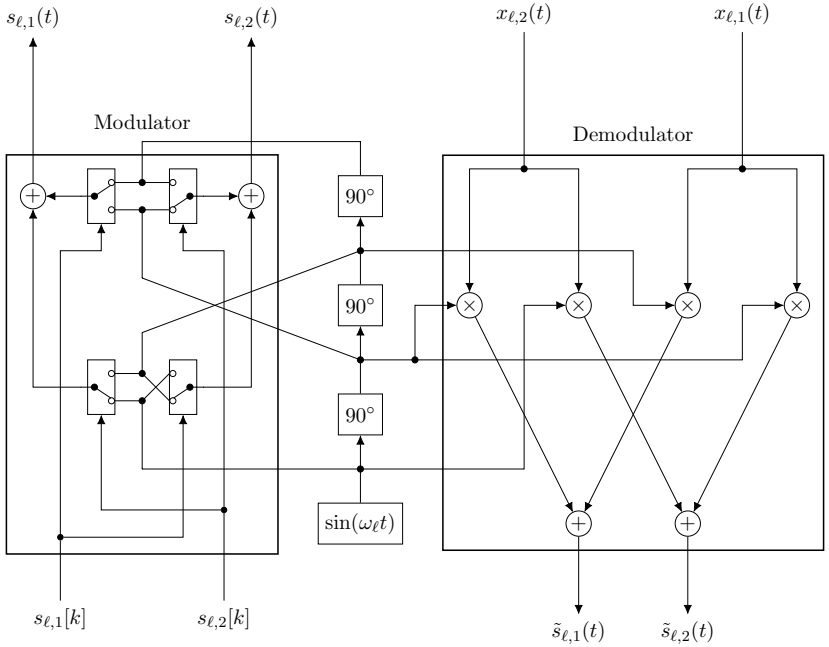


Figure 7.4: The figure shows a modulator block as those given in Figure 7.1. The modulator converts signals to and from a given frequency $\omega_\ell/(2\pi)$. Note that due to the binary nature of $s_{\ell,1}[k], s_{\ell,2}[k]$ the multiplication in the modulation step can be simplified using only switches.

7.4.1 Control Contribution

For the chain-of-oscillator DC we think of the modulation task, i.e., the left side of Figure 7.4, as part of the DAC waveform $\boldsymbol{d}_\ell(t)$, as was covered in Section 4.2.1. Specifically, the relation between the control signal and control contribution can be written as

$$\begin{pmatrix} s_{\ell,1}(t) \\ s_{\ell,2}(t) \end{pmatrix} = \underbrace{\begin{pmatrix} \cos(\omega_\ell t) & -\sin(\omega_\ell t) \\ \sin(\omega_\ell t) & \cos(\omega_\ell t) \end{pmatrix}}_{\boldsymbol{d}_\ell(t)} \begin{pmatrix} s_{\ell,1}[k] \\ s_{\ell,2}[k] \end{pmatrix} \tag{7.60}$$

for $t \in (kT, (k+1)T]$. From the expression (7.60) we once more recognize the rotation matrix that appeared frequently in Section 7.2. Furthermore, when comparing (7.60) to (7.8) we recognize that $s_{\ell,1}(t)$ is of the same form as the input vector from (7.8) for $a_1(t) = s_{\ell,1}[k], a_2(t) = 0$, and likewise $s_{\ell,2}(t)$ is of the same form as when $a_1(t) = 0, a_2(t) = s_{\ell,2}[k]$. This is not a coincidence, in fact, $s_{\ell,1}(t)$ can be thought of as the control contribution intended for the quadrature input component and similarly, $s_{\ell,2}(t)$ the control contribution corresponding to the in-phase component.

Since the rotation matrix is rotating with the same frequency as the resonance frequency of the oscillator node, we can use the result from (7.16) to determine how the AS state evolves for different controls. Specifically, the four possible control combinations

$$\begin{pmatrix} s_{\ell,1}[k] \\ s_{\ell,2}[k] \end{pmatrix} \in \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix} \right\} \tag{7.61}$$

$$= \left\{ \boldsymbol{s}^{(1,1)}, \boldsymbol{s}^{(1,-1)}, \boldsymbol{s}^{(-1,1)}, \boldsymbol{s}^{(-1,-1)} \right\} \tag{7.62}$$

result in the control contributions

$$\begin{pmatrix} x_{\ell,1}(t) \\ x_{\ell,2}(t) \end{pmatrix} = \kappa_\ell \boldsymbol{\Theta}(\omega_\ell t_1) \int_{kT}^{t} \boldsymbol{\Theta}(-\omega_\ell \tau) \boldsymbol{d}_\ell(\tau) \begin{pmatrix} s_{\ell,1}[k] \\ s_{\ell,2}[k] \end{pmatrix} \mathrm{d}\tau \tag{7.63}$$

$$= \kappa_\ell (t - kT) \left(\boldsymbol{\Theta}(\omega_\ell t) - \boldsymbol{\Theta}(\omega_\ell kT) \right) \begin{pmatrix} s_{\ell,1}[k] \\ s_{\ell,2}[k] \end{pmatrix} \tag{7.64}$$

where we have assumed $t \in (kT, (k+1)T]$. The expression from (7.64) is visualized in Figure 7.5. The figure is normalized for unit growth during one time period of the oscillator resonance frequency. Furthermore, the figure shows the state evolution for three such time periods. As is evident from the figure. The proposed control contribution results in a $\|\boldsymbol{x}_\ell(t)\|_2$ term, which grows linearly over time.

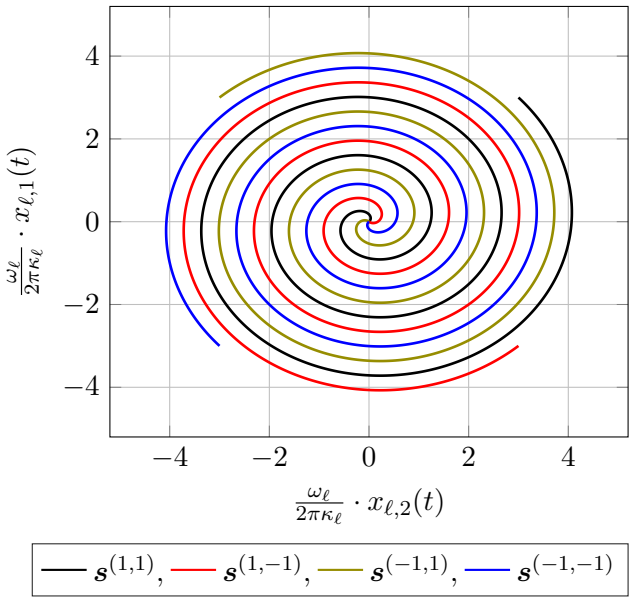


Figure 7.5: AS state evolution (7.64) visualized for the four different control signal configurations as in (7.62). The x and y-axis are normalized for unit growth with respect to the time period at oscillation frequency ω_ℓ. The figure shows the growth during three such time periods

Demodulation

The demodulation from Figure 7.4 can be written in a similar way as in (7.60) namely,

$$\begin{pmatrix} \tilde{s}_{\ell,1}(t) \\ \tilde{s}_{\ell,2}(t) \end{pmatrix} = \underbrace{\begin{pmatrix} \cos(\omega_\ell t) & \sin(\omega_\ell t) \\ -\sin(\omega_\ell t) & \cos(\omega_\ell t) \end{pmatrix}}_{\boldsymbol{\Theta}(-\omega_\ell t)} \begin{pmatrix} x_{\ell,1}(t) \\ x_{\ell,2}(t) \end{pmatrix}. \tag{7.65}$$

Interestingly, for an arbitrary AS state vector as

$$\begin{pmatrix} x_{\ell,1}(t) \\ x_{\ell,2}(t) \end{pmatrix} = a_{\ell,1}(t) \begin{pmatrix} \cos(\omega_\ell t + \phi_\ell) \\ \sin(\omega_\ell t + \phi_\ell) \end{pmatrix} + a_{\ell,2}(t) \begin{pmatrix} -\sin(\omega_\ell t + \phi_\ell) \\ \cos(\omega_\ell t + \phi_\ell) \end{pmatrix} \tag{7.66}$$

the demodulation operation results in

$$\begin{pmatrix} \tilde{s}_{\ell,1}(t) \\ \tilde{s}_{\ell,2}(t) \end{pmatrix} = \begin{pmatrix} a_{\ell,1}(t) \\ a_{\ell,2}(t) \end{pmatrix} \cos(\phi_\ell). \tag{7.67}$$

In other words, the demodulation process separates the two control dimensions defined by the modulated DC as in (7.64). One way to visualize this is by considering the AS state trajectory of an oscillator node without input

$$\begin{aligned} \begin{pmatrix} x_{\ell,1}(t) \\ x_{\ell,2}(t) \end{pmatrix} = \kappa_\ell t \left(s_{\ell,1}[0] \begin{pmatrix} \cos(\omega_\ell t) \\ \sin(\omega_\ell t) \end{pmatrix} + s_{\ell,2}[0] \begin{pmatrix} -\sin(\omega_\ell t) \\ \cos(\omega_\ell t) \end{pmatrix} \right) \\ + \boldsymbol{\Theta}(\omega_\ell t) \begin{pmatrix} x_{\ell,1}(0) \\ x_{\ell,2}(0) \end{pmatrix} \end{aligned} \tag{7.68}$$

for $t \in (0, T]$. From the expression we recognize the general state form from (7.66) with

$$a_{\ell,1}(t) = \kappa_\ell t s_{\ell,1}[0] + x_{\ell,1}(0) \tag{7.69}$$

$$a_{\ell,2}(t) = \kappa_\ell t s_{\ell,2}[0] + x_{\ell,2}(0) \tag{7.70}$$

As an alternative, the demodulation could also be implemented as

$$\begin{pmatrix} \tilde{s}_{\ell,1}(t) \\ \tilde{s}_{\ell,2}(t) \end{pmatrix} = \begin{pmatrix} \cos(\omega_\ell t) & 0 \\ 0 & \cos(\omega_\ell t) \end{pmatrix} \begin{pmatrix} x_{\ell,1}(t) \\ x_{\ell,2}(t) \end{pmatrix} \tag{7.71}$$

as for an arbitrary oscillator node state in (7.66),

(7.71) can then be written as

$$\begin{aligned} \begin{pmatrix} \tilde{s}_{\ell,1}(t) \\ \tilde{s}_{\ell,2}(t) \end{pmatrix} = \frac{a_{\ell,1}(t)}{2} \begin{pmatrix} \cos(-\phi_\ell) + \cos(2\omega_\ell t + \phi_\ell) \\ \sin(2\omega_\ell t + \phi_\ell) - \sin(-\phi_\ell) \end{pmatrix} \\ + \frac{a_{\ell,2}(t)}{2} \begin{pmatrix} \sin(-\phi_\ell) - \sin(2\omega_\ell t + \phi_\ell) \\ \cos(2\omega_\ell t + \phi_\ell) + \cos(-\phi_\ell) \end{pmatrix}. \end{aligned} \tag{7.72}$$

Furthermore, by elementwise low-pass filtering the expressions above, the $2\omega_\ell$ terms are suppressed, and the demodulated signal is obtained. The difference of this latter approach (7.71) is that only two mixing operations are necessary for the demodulation at the expense of an additional low-pass filtering step.

7.4.2 General Remarks

Note that the demodulation task does not need to be implemented with the same precision as the modulation task. This is because the 1-bit quantizers only use the demodulated signal, whereas the modulated signal adds to the signal path. Thankfully, due to the binary nature of the control signals $s_{\ell,1}(t)$, $s_{\ell,2}(t)$, the modulation can be implemented as switching between a different phase-delayed version of a global free-running oscillator.

Furthermore, as the same oscillator is used for both modulation and demodulation, see Figure 7.4, the absolute phase of the oscillator has no influence for the modulation and demodulation of the controls and states respectively.

For an n-th order chain-of-oscillators the control input and observation matrix follows as shown in (7.53) and (7.54). The expression in (7.54) reveals that each oscillator node is demodulated independently as in (7.65). For the control contribution, the modulation is described as part of the DAC waveform. Therefore,

$$\boldsymbol{D}(t) = \begin{pmatrix} d_1(t) & & \\ & \ddots & \\ & & d_n(t) \end{pmatrix} \in \mathbb{R}^{2n \times 2n} \tag{7.73}$$

where each DAC waveform $d_\ell(t)$ is as in (7.60).

A key argument for using the DC as described above is the fact that the modulation and demodulation reduce the control problem to the same as in the chain-of-integrators. This means that β_ℓ, κ_ℓ and T can be dimensioned exactly as outlined in Section 5.3.1.

Finally, note that during nominal conditions the oscillator nodes can be completely stabilized by the DC. This means that the involved oscillators could ideally be operated without an oscillation amplitude limiter.

7.4.3 Non-Oscillating Digital Control

The oscillator nodes, as in Section 7.2, can also be controlled via a DC without any oscillating DAC waveforms. An extreme example would be to use the previously used square DAC waveform as was done for both the chain-of-integrators ADC as well as the leapfrog ADC. To be precise, what is proposed is to remove the modulator block from Figure 7.1 such

that the DC directly control the AS states using $2n$ independent local DCs. For a single oscillator node, the proposed DC would result in the state vector

$$\begin{pmatrix} x_{\ell,1}(t_1) \\ x_{\ell,2}(t_1) \end{pmatrix} = \kappa_\ell \boldsymbol{\Theta}\left(\omega_\ell t_1\right) \int_{t_0}^{t_1} \boldsymbol{\Theta}\left(-\omega_\ell \tau\right) \mathrm{d}\tau \begin{pmatrix} s_{\ell,1}[k] \\ s_{\ell,2}[k] \end{pmatrix} \tag{7.74}$$

$$= \kappa_\ell \boldsymbol{\Theta}(\omega_\ell t_1) \left(\boldsymbol{\psi}_\ell(-\omega_\ell t_0) - \boldsymbol{\psi}_\ell(-\omega_\ell t_1)\right) \begin{pmatrix} s_{\ell,1}[k] \\ s_{\ell,2}[k] \end{pmatrix} \tag{7.75}$$

$$= \kappa_\ell \left(\boldsymbol{\psi}_\ell(\omega_\ell(t_1 - t_0)) - \boldsymbol{\psi}_\ell(0)\right) \begin{pmatrix} s_{\ell,1}[k] \\ s_{\ell,2}[k] \end{pmatrix} \tag{7.76}$$

where

$$\boldsymbol{\psi}_\ell(\phi) \triangleq \frac{1}{\omega_\ell} \begin{pmatrix} \sin(\phi) & \cos(\phi) \\ -\cos(\phi) & \sin(\phi) \end{pmatrix} \tag{7.77}$$

$$= \frac{1}{\omega_\ell} \boldsymbol{\Theta}\left(\phi + \frac{\pi}{2}\right). \tag{7.78}$$

The corresponding trajectories are given in Figure 7.6. From the figure we recognize that the resulting state trajectories are periodic with the time period $T_{\omega_\ell} = \frac{2\pi}{\omega_\ell}$ corresponding to the resonance frequency of the oscillator node. Specifically, each of the four control contributions returns to $\big(0, 0\big)$ at the beginning of time period T_{ω_ℓ}. To demonstrate this more clearly, Figure 7.7 shows the state trajectory resulting from the control contribution $s_{\ell,1}(t) = s_{\ell,2}(t) = 1$. From the figure we see that the oscillator node state vector norm $\| \big(x_{\ell,1}(t), \quad x_{\ell,2}(t)\big) \|_2$ reaches its largest and smallest value at the times $t = \xi \frac{T_{\omega_\ell}}{2}$ and $t = \xi T_{\omega_\ell}$ respectively, where ξ is a positive integer. A direct consequence is that for a control period $T = \xi T_{\omega_\ell}$ the control contribution has no effect at the end of a control period and cannot control the oscillator node states.

For a $T = \xi T_{\omega_\ell} + \frac{T_{\omega_\ell}}{2}$, the resulting state vector can be written as

$$\begin{pmatrix} x_{\ell,1}(T) \\ x_{\ell,2}(T) \end{pmatrix} = \frac{2\kappa_\ell}{\omega_\ell} \begin{pmatrix} -s_{\ell,2}[k] \\ s_{\ell,1}[k] \end{pmatrix} \tag{7.79}$$

which is also confirmed from Figure 7.7. These cases also correspond to the largest AS state trajectories from a square DAC control contribution. Furthermore, we notice that the expression from (7.76) does not grow linearly with t_1 like in the case for the oscillating DC, cf. (7.64). This means that for a carrier frequency much greater than the frequency band of interest, an excessively large κ_ℓ might be necessary to control the oscillator node's state vector.

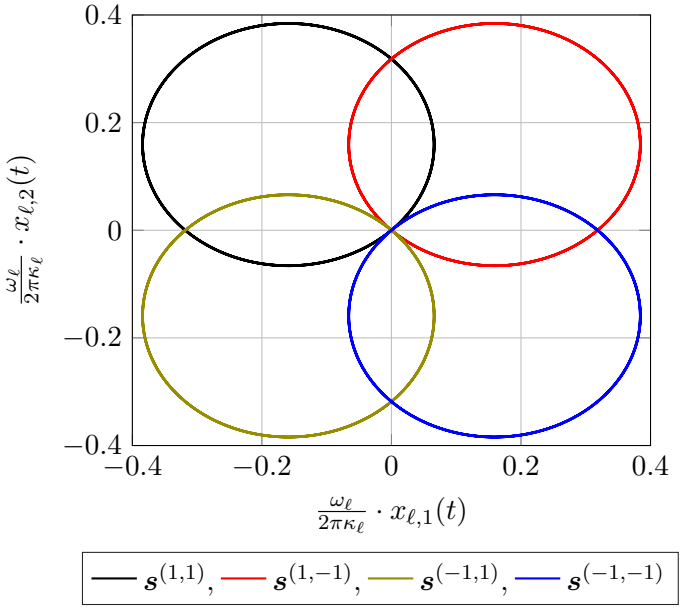


Figure 7.6: AS state trajectories for a single oscillator node from a control contribution given by a square DAC waveform. Note that the trajectories are all periodic with the time period $T_{\omega_\ell} = \frac{2\pi}{\omega_\ell}$.

Furthermore, the control observation needs to be aligned such that the control ends up opposing the state at the end of each control period. This can be done by a fixed rotation, i.e., independent of time

$$\begin{pmatrix} \tilde{s}_{\ell,1}(t) \\ \tilde{s}_{\ell,2}(t) \end{pmatrix} = \boldsymbol{\Theta}\left(\phi_\ell\right) \begin{pmatrix} x_{\ell,1}(t) \\ x_{\ell,2}(t) \end{pmatrix} \tag{7.80}$$

where

$$\phi_\ell = -\left(\omega_\ell T + \frac{\pi}{2}\right). \tag{7.81}$$

The proposed rotation can be incorporated into the control observation matrix as

$$\tilde{\boldsymbol{\Gamma}}_{\mathrm{CO}} = \begin{pmatrix} \boldsymbol{\Theta}(\phi_1) & & \\ & \ddots & \\ & & \boldsymbol{\Theta}(\phi_n) \end{pmatrix}. \tag{7.82}$$

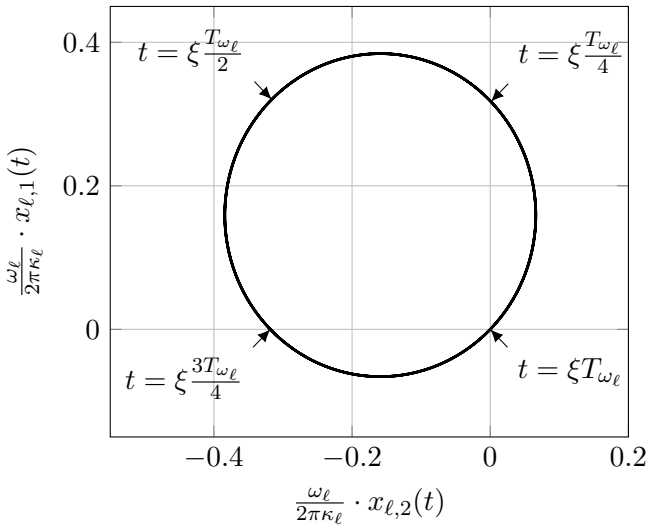


Figure 7.7: Resulting state trajectory for $s_{\ell,1}(t) = s_{\ell,2}(t) = 1$. The evolution follows the drawn line in counterclockwise direction and the specific time of $\xi T_{\omega_\ell}, \xi\frac{T_{\omega_\ell}}{4}, \xi\frac{T_{\omega_\ell}}{2}, \xi\frac{3T_{\omega_\ell}}{4}$ are indicated for any positive integer ξ.

Note that the control observation matrix is not time dependent, which was the case for the oscillating DC, cf. (7.54).

The steps presented in this section can be repeated for non-oscillating DAC waveforms other than the squared one. In particular, the switched capacitor DC from Section 5.3.2 is particularly interesting since, if the discharge of the capacitor is much smaller than T_{ω_ℓ}, it behaves very similar to how it operates for a chain-of-integrators ADC.

7.5 Digital Estimator

The DE of the chain-of-oscillators ADC is similar to that of the chain-of-integrators DE presented in Section 5.4. In particular, for $n = 5$, $\omega_1 = \cdots = \omega_5 = 2\pi f_c$, and $\beta_1 = \cdots = \beta_5 = \beta$, the norm of the NTF and STF are evaluated in Figure 7.8 as a function of frequency. Note that in that figure, the x-axis is centered around f_c. Furthermore, the same figure with an logarithmic x-axis is shown in Figure 7.9. In this case, we

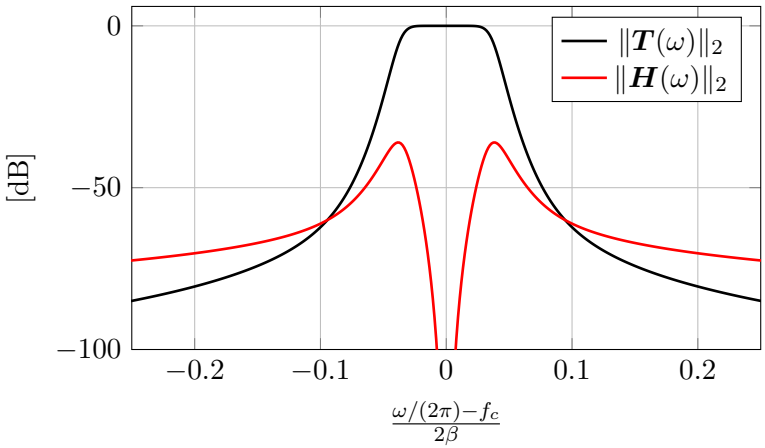


Figure 7.8: The resulting STF and NTF plotted for the chain-of-oscillators ADC. Note that we have centered the x-axis around the resonance frequency of the oscillator node.

only show the positive half of the frequency band of interest.

The Figure 7.9 closely resemble that of Figure 5.6 which once more confirm that the chain-of-oscillators ADC behaves like a chain-of-integrators ADC centered around some frequency f_c.

Computational Complexity

What is particularly noteworthy with the chain-of-oscillator DE is that the time-dependent DAC waveform does not affect the digital estimation filter's complexity. Instead, these are incorporated in the offline precomputations of (4.63) and (4.64). Also, note that the control observation matrix $\tilde{\boldsymbol{\Gamma}}_{\text{CO}}^{\text{T}}(t)$ takes no part in the DE. In other words, the digital estimation filter follows, as before, from Equations (4.53)-(4.55), and the resulting filter coefficients (matrices) are time-invariant.

Therefore, the chain-of-oscillators DE computational complexity scales similarly to the chain-of-integrators DE, with the number of oscillator nodes as $M = N = 2n$ and $L = 1$. Using the derivations of Section 4.3.4, we can summarize the DE's computational complexity as

- $\mathcal{O}(n)$ real-valued scalar multiplications,

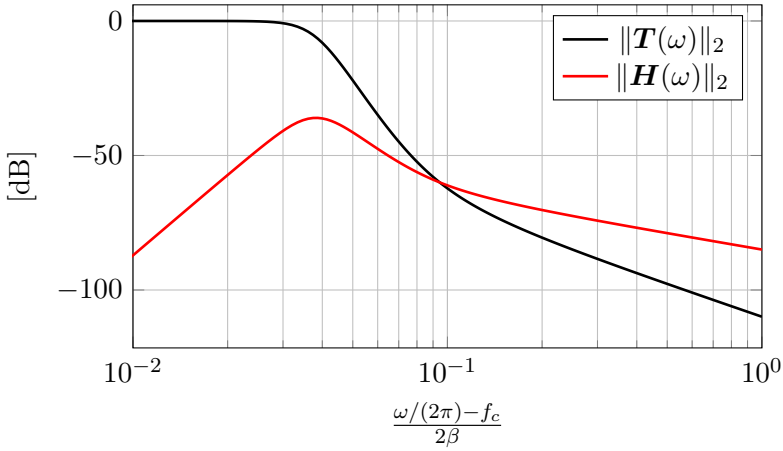


Figure 7.9: The positive half of Figure 7.8 plotted with a logarithmic x-axis.

- $\mathcal{O}\left(n^2\right)$ real-valued scalar additions,
- and requires $2n$ bits and $8n + 2$ real-valued scalar values to be kept in memory

per estimated sample when implemented using the offline batch estimator from Algorithm 3 from Appendix E.

Chapter 8

Hadamard Analog-to-Digital Converter

The goal with the Hadamard ADC is to distribute component mismatch sensitivity equally over the involved circuit components. The desired effect is achieved by separating the signal dimensions, or the logical signal paths, from the physical ones. The Hadamard ADC by itself will not render any nominal performance improvement. However, it is a significant building block together with the overcomplete control from Chapter 9 for the multi-input ADC presented in Chapter 10.

Furthermore, the Hadamard ADC examples presented here are all extensions of the chain-of-integrators ADC from Chapter 5. However, the general principle could just as well be applied to any control-bounded ADC.

8.1 Analog System

The mentioned separation of physical and logical states or signal dimensions is achieved by rotating the state space. The rotation is done with a scaled Hadamard matrix $\boldsymbol{H}_N$, which also warrants the name of this control-bounded ADC.

A Hadamard matrix can be defined recursively as

$$\boldsymbol{H}_N \triangleq \boldsymbol{H}_2 \otimes \boldsymbol{H}_{N/2} \tag{8.1}$$

where

$$\boldsymbol{H}_2 \triangleq \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{8.2}$$

and $\otimes$ is the Kronecker product. From (8.1) it is clear that the Hadamard matrix is only defined for N being powers of two.

Furthermore, the Hadamard matrix has two, especially nice, properties that we will highlight next. In particular, the Hadamard matrix is symmetric, i.e., $\boldsymbol{H}_N = \boldsymbol{H}_N^\mathsf{T}$ and its inverse is a scaled version of itself as

$$\boldsymbol{H}_N^\mathsf{T}\boldsymbol{H}_N = N \cdot \boldsymbol{I}_N \tag{8.3}$$

or equivalently, $\boldsymbol{H}_N^{-1} = \frac{1}{N}\boldsymbol{H}_N^\mathsf{T}$.

The mentioned rotation is implemented by applying the Hadamard transformation to the state vector of the AS. This means that the Hadamard ADC's AS is described by the equations

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_\mathrm{H}\boldsymbol{x}(t) + \boldsymbol{B}_\mathrm{H}\boldsymbol{u}(t) + \boldsymbol{\Gamma}_\mathrm{H}\boldsymbol{s}(t) \tag{8.4}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}_\mathrm{H}^\mathsf{T}\boldsymbol{x}(t) \tag{8.5}$$

$$\tilde{\boldsymbol{s}}(t) = \tilde{\boldsymbol{\Gamma}}_\mathrm{H}^\mathsf{T}\boldsymbol{x}(t) \tag{8.6}$$

where

$$\boldsymbol{A}_\mathrm{H} = \frac{1}{N}\boldsymbol{H}_N\boldsymbol{A}_\mathrm{CI}\boldsymbol{H}_N^\mathsf{T}, \tag{8.7}$$

$$\boldsymbol{B}_\mathrm{H} = \frac{1}{N}\boldsymbol{H}_N\boldsymbol{B}_\mathrm{CI}, \tag{8.8}$$

$$\boldsymbol{C}_\mathrm{H}^\mathsf{T} = \boldsymbol{C}_\mathrm{CI}^\mathsf{T}\boldsymbol{H}_N^\mathsf{T}, \tag{8.9}$$

$$\tilde{\boldsymbol{\Gamma}}_\mathrm{H}^\mathsf{T} = \tilde{\boldsymbol{\Gamma}}_\mathrm{CI}^\mathsf{T}\boldsymbol{H}_N^\mathsf{T}, \tag{8.10}$$

and

$$\boldsymbol{\Gamma}_\mathrm{H} = \frac{1}{N}\boldsymbol{H}_N\boldsymbol{\Gamma}_\mathrm{CI}, \tag{8.11}$$

where $\boldsymbol{H}_N$ is a Hadamard matrix and $\boldsymbol{A}_\mathrm{CI}$, $\boldsymbol{B}_\mathrm{CI}$, $\boldsymbol{C}_\mathrm{CI}$, $\boldsymbol{\Gamma}_\mathrm{CI}$, and $\tilde{\boldsymbol{\Gamma}}_\mathrm{CI}$ refers to the chain-of-integrators parameterization from Equations (5.2)-(5.7). Notice the scaling in (8.7), (8.8) and (8.11). These are necessary to maintain an effective control and will be motivated in Section 8.2.

Transfer Function Analysis

The transfer function analysis of the Hadamard ADC is identical to that of the chain-of-integrators ADC from Section 5.2. To see this consider the following manipulations

$$\begin{aligned}
\boldsymbol{G}_{\mathrm{H}}(\omega) &= \boldsymbol{C}_{\mathrm{H}}^{\mathsf{T}} \left(i\omega \boldsymbol{I}_N - \boldsymbol{A}_{\mathrm{H}}\right)^{-1} \boldsymbol{B}_{\mathrm{H}} && (8.12)\\
&= \boldsymbol{C}_{\mathrm{CI}}^{\mathsf{T}} \boldsymbol{H}_N^{\mathsf{T}} \left(i\omega \boldsymbol{I}_N - \frac{1}{N} \boldsymbol{H}_N \boldsymbol{A}_{\mathrm{CI}} \boldsymbol{H}_N^{\mathsf{T}}\right)^{-1} \frac{1}{N} \boldsymbol{H}_N \boldsymbol{B}_{\mathrm{CI}} && (8.13)\\
&= \boldsymbol{C}_{\mathrm{CI}}^{\mathsf{T}} \boldsymbol{H}_N^{\mathsf{T}} \left(\frac{1}{\sqrt{N}} \boldsymbol{H}_N \left(i\omega \boldsymbol{I}_N - \boldsymbol{A}_{\mathrm{CI}}\right) \frac{1}{\sqrt{N}} \boldsymbol{H}_N^{\mathsf{T}}\right)^{-1} \frac{1}{N} \boldsymbol{H}_N \boldsymbol{B}_{\mathrm{CI}} && (8.14)\\
&= \boldsymbol{C}_{\mathrm{CI}}^{\mathsf{T}} \boldsymbol{H}_N^{\mathsf{T}} \left(\frac{1}{\sqrt{N}} \boldsymbol{H}_N^{\mathsf{T}}\right)^{-1} \left(i\omega \boldsymbol{I}_N - \boldsymbol{A}_{\mathrm{CI}}\right)^{-1} \left(\frac{1}{\sqrt{N}} \boldsymbol{H}_N\right)^{-1} \frac{1}{N} \boldsymbol{H}_N \boldsymbol{B}_{\mathrm{CI}} && (8.15)\\
&= \boldsymbol{C}_{\mathrm{CI}}^{\mathsf{T}} \left(\frac{1}{N} \boldsymbol{H}_N^{\mathsf{T}} \boldsymbol{H}_N \left(i\omega \boldsymbol{I}_N - \boldsymbol{A}_{\mathrm{CI}}\right)^{-1} \frac{1}{N} \boldsymbol{H}_N^{\mathsf{T}} \boldsymbol{H}_N\right) \boldsymbol{B}_{\mathrm{CI}} && (8.16)\\
&= \boldsymbol{C}_{\mathrm{CI}}^{\mathsf{T}} \left(i\omega \boldsymbol{I}_N - \boldsymbol{A}_{\mathrm{CI}}\right)^{-1} \boldsymbol{B}_{\mathrm{CI}} && (8.17)\\
&= \boldsymbol{G}_{\mathrm{CI}}(\omega), && (8.18)
\end{aligned}$$

where (8.14) follows from (8.3).

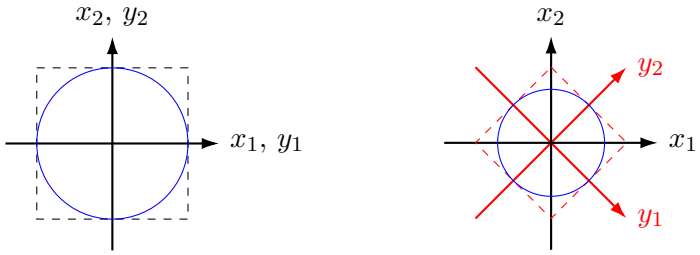
8.2 Digital Control

As the Hadamard A/D converter operates in a transformed signal space, additional care has to be taken with respect to the DC. The fundamental problem is that our nominal performance is determined by $\|\boldsymbol{G}_{\mathrm{H}}(\omega)\|_2^2$, whereas the effective control until now has been ensured locally via bounding the state vector as $\|\boldsymbol{x}(t)\|_\infty$. This is illustrated in Figure 8.1.

Assuming a second order system, $N = 2$, Figure 8.1a indicates the permissible state by the dashed box. Additionally, the blue circle represents the maximal AS gain ($\|\cdot\|_2$).

As seen in Figure 8.1b, rotating the state space, means shrinking the blue circle, i.e., the permissible AS amplification, for the same $\|\cdot\|_\infty$ bound of the state vector.

At first sight, the reduction in amplification appears to be a significant performance limitation essentially reducing the expected SNR by a factor of $(1/N)^N$ if applied to every node of the chain. However, Figure 8.1b



(a) Chain-of-integrators ADC **(b)** Hadamard ADC

Figure 8.1: States and control bounds for a chain-of-integrators and a Hadamard ADC.

also presents the solution; as the AS gain is reduced, so is the growth term, and subsequently, less control gain κ is necessary, which means that also the remainder term shrinks, cf. Section 4.2.2. This means that both amplification and the control bound $b_{\boldsymbol{y}}$ are scaled down equally, and therefore, the nominal SNR of the converter is preserved.

The mentioned scaling can be translated to the state space parameterization in Equations (8.7)-(8.11) in three steps. First, rotating the state space with $\frac{1}{\sqrt{N}} \cdot \boldsymbol{H}_N$ preserves the L_2 norm and is therefore realized as

$$\boldsymbol{A}_{\mathrm{H}} \leftarrow \frac{1}{\sqrt{N}} \cdot \boldsymbol{H}_N^{\mathsf{T}} \boldsymbol{A}_{\mathrm{CI}} \boldsymbol{H}_N \cdot \frac{1}{\sqrt{N}} \tag{8.19}$$

$$\boldsymbol{B}_{\mathrm{H}} \leftarrow \frac{1}{\sqrt{N}} \cdot \boldsymbol{H}_N \boldsymbol{B}_{\mathrm{CI}} \tag{8.20}$$

$$\boldsymbol{\Gamma}_{\mathrm{H}} \leftarrow \frac{1}{\sqrt{N}} \cdot \boldsymbol{H}_N \boldsymbol{\Gamma}_{\mathrm{CI}} \tag{8.21}$$

$$\boldsymbol{C}_{\mathrm{H}}^{\mathsf{T}} \leftarrow \frac{1}{\sqrt{N}} \cdot \boldsymbol{C}_{\mathrm{CI}}^{\mathsf{T}} \boldsymbol{H}_N^{\mathsf{T}} \tag{8.22}$$

$$\tilde{\boldsymbol{\Gamma}}_{\mathrm{H}}^{\mathsf{T}} \leftarrow \frac{1}{\sqrt{N}} \cdot \tilde{\boldsymbol{\Gamma}}_{\mathrm{CI}}^{\mathsf{T}} \boldsymbol{H}_N^{\mathsf{T}} \tag{8.23}$$

Secondly, we scale down both the input matrix $\boldsymbol{B}_{\mathrm{H}}$ and control input matrix $\boldsymbol{\Gamma}_{\mathrm{H}}$ by an additional $\frac{1}{\sqrt{N}}$ as previously discussed. Finally, scaling up the two signal observation matrices $\boldsymbol{C}_{\mathrm{H}}^{\mathsf{T}}$ and $\tilde{\boldsymbol{\Gamma}}_{\mathrm{H}}^{\mathsf{T}}$ by $\sqrt{N}$ does not violate any bounds. In particular, for the $\boldsymbol{C}_{\mathrm{H}}^{\mathsf{T}}$ matrix this is especially uneventfully, since this is a conceptual quantity only used in the offline computations of the DE. The scaling in $\boldsymbol{C}_{\mathrm{H}}^{\mathsf{T}}$ results in an increased

control bound $b_{\boldsymbol{y}}$ and amplification such that these are identical to the corresponding chain-of-integrators example.

Additionally, these rotations means that we can apply the same local DC strategy, as in Section 5.3, to ensure an effective control. The only difference being that the bounded states lie in a rotated state space compared to the physical one.

Misaligned Control and Signal Dimensions

Rotating the signal space does not necessarily imply that the control also must be rotated as in (8.10) and (8.11). In fact, maintaining the local control as in the chain-of-integrators ADC from Chapter 5 turns out to suppress high-frequency noise coming from the local DC, see Figure 8.2 and Figure 8.3.

To be precise, a Hadamard ADC using local control can be described by the state space description

$$\boldsymbol{A}_{\mathrm{H}} = \frac{1}{N}\boldsymbol{H}_N \boldsymbol{A}_{\mathrm{CI}} \boldsymbol{H}_N^{\mathsf{T}}, \tag{8.24}$$

$$\boldsymbol{B}_{\mathrm{H}} = \frac{1}{N}\boldsymbol{H}_N \boldsymbol{B}_{\mathrm{CI}}, \tag{8.25}$$

$$\boldsymbol{C}_{\mathrm{H_{LC}}}^{\mathsf{T}} = \boldsymbol{C}_{\mathrm{CI}}^{\mathsf{T}}, \tag{8.26}$$

$$\tilde{\boldsymbol{\Gamma}}_{\mathrm{H_{LC}}}^{\mathsf{T}} = \tilde{\boldsymbol{\Gamma}}_{\mathrm{CI}}^{\mathsf{T}}, \tag{8.27}$$

and

$$\boldsymbol{\Gamma}_{\mathrm{H_{LC}}} = \boldsymbol{\Gamma}_{\mathrm{CI}}. \tag{8.28}$$

An immediate effect of this parameter choice is that we cannot maintain the same bound on the signal observation $\boldsymbol{y}(t)$ as in the regular Hadamard ADC or an equivalent chain-of-integrators ADC. This can be seen from the geometry of Figure 8.1b where, as before, we have to reduce the gain but are unable to maintain the tighter bound due to the misalignment.

Figure 8.2 shows the PSD for a full-scale input at a frequency $\Omega/(2\pi T) \approx 0.062$, for $N = 5$. The figure shows both the regular Hadamard converter (HC) and the Hadamard converter with local control (HCL). From the figure, we notice a clear difference between the two approaches at high frequencies; the local control has much less control artifacts. This makes the misaligned local control especially interesting for higher-order systems operated with small OSR.

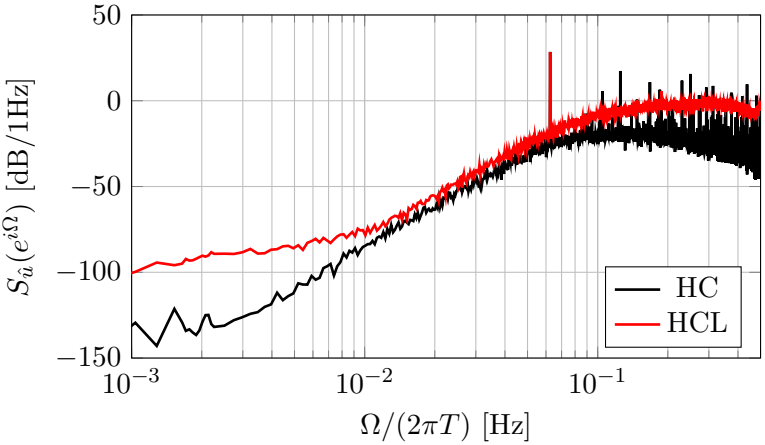


Figure 8.2: The PSD of $\hat{u}(kT)$ for a Hadamard converter with local control (HCL) compared to the standard version (HC). For the simulations a full-scale sinusoidal input signal at $\Omega/(2\pi T) \approx 0.062$ and OSR $= 4$ has been used.

The misaligned local DC can alternatively be viewed as a way of creating a dithering effect, much like in Figure 5.13 from Section 5.5.2. This view is confirmed by the results from the limit cycle simulation in Figure 8.3. The limit cycle at $\Omega/(2\pi) = 0.003$ essentially vanishes for the Hadamard ADC when using local control, as compared to the standard version.

8.3 Digital Estimator

Interestingly, the digital estimator filter coefficients (4.60)-(4.64) are identical to those of the chain-of-integrators DE for any orthonormal $\mathbf{\Gamma}$ matrix when the AS as in (8.7)-(8.11). In other words, the chain-of-Integrators DE can be used on the control signals $\boldsymbol{s}[k]$ to form the final estimate. This insight once more highlights that the Hadamard ADC is means of enhancing the robustness properties of AS. However, it does not fundamentally change the underlying structure in terms of nominal performance.

Clearly, the identical filter coefficients does not apply for the misaligned control case, (8.24)-(8.28), as this both changes the DE filter coefficients

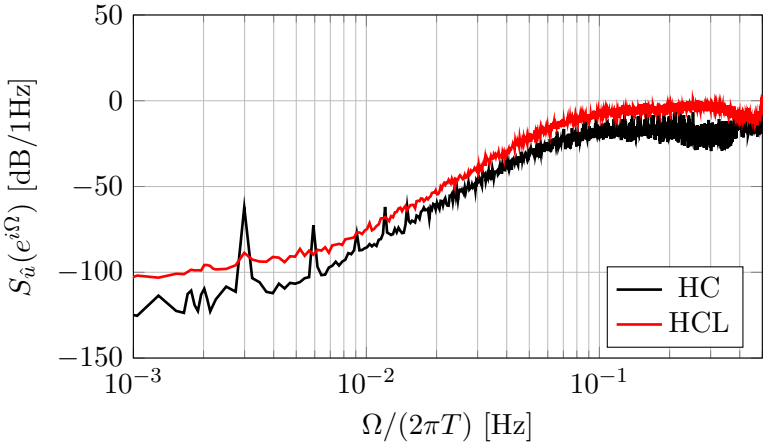


Figure 8.3: The PSD of $\hat{u}(kT)$ for a limit cycle simulation where $u(t) = 0.003$ as in Figure 5.12, comparing the Hadamard converter with local control (HCL) and the standard version (HC).

and futhermore the characteristics of the AS and DC interaction.

Finally, for any square matrix $\boldsymbol{H}_N$ the DE computational complexity is unchanged and identical to that of the chain-of-integrators DE from Section 5.4.5.

8.4 Proposed Hardware Implementation

As previously stated, the Hadamard ADC does not increase the nominal performance. Instead, it was motivated by its mismatch suppression capabilities and non-centralized design. This can be visualized in terms of a possible hardware implementation as in Figure 8.4. This architecture is essentially the Hadamard extension to the chain-of-integrators ADC from Section 5.6 for $N = 4$. Furthermore, the Hadamard matrix $\{1, -1\}$ multiplications can be implemented conveniently using differential amplifiers, since it amounts to crossing, alternatively not crossing, the wires of resistors. Using differential amplifiers also means that the signal is represented as the voltage potential between a positive and a negative wire as opposed to a wire and a signal ground. To illustrate this, we use the

notation

$$x_\ell^+(t) - x_\ell^-(t) = x_\ell(t) \tag{8.29}$$

and

$$\tilde{s}_\ell^+(t) - \tilde{s}_\ell^-(t) = \tilde{s}_\ell(t). \tag{8.30}$$

Specifically, the fully differential operational amplifiers with capacitive feedback make up the integrators from Figure 8.4 and the $\boldsymbol{H}_4(R)$ blocks ensure the rotated state space. Note that the capacitors shown in the figure are all dimensioned the same way, with a capacitance C. Furthermore, there are multiple ways to implement these Hadamard networks, either as a passive resistor network as in Figure 8.5 or by using additional voltage buffers, as shown in Figure 8.6. The latter is inspired by the fast Walsh-Hadamard transform, where the additions are broken down into $\log_2(N)$ steps.

To accommodate the specific scaling from (8.7), (8.8), (8.11), several adaptations are made. Specifically, the time constant RC is adapted such that

$$\frac{\sqrt{N}}{RC} = \beta. \tag{8.31}$$

Furthermore, (8.8) is ensured by restricting

$$b_{\boldsymbol{u}} = \frac{b_{\boldsymbol{x}}}{\sqrt{N}}. \tag{8.32}$$

Finally, the scaling in (8.11) is realized by increasing the resistance in the corresponding Hadamard resistor network by a factor $\sqrt{N} = 2$. Additionally, R_∞ symbolizes resistor values which are merely used for averaging and are not part of the gain constant. Therefore, R_∞ is preferably chosen relatively large to limit power consumption.

The Hadamard converter utilizes many more resistors (N^2) than the corresponding chain-of-integrators ADC (N). However, for a fixed C, and due to the increased R according to (8.31), each resistor consumes less power, since the power consumption is distributed over a larger number of resistors. This will be covered in detail in Section 8.4.3, showing that the total current into each capacitor, and therefore the total power consumed by all the resistors, remain constant between the Hadamard ADC and the chain-of-integrators ADC.

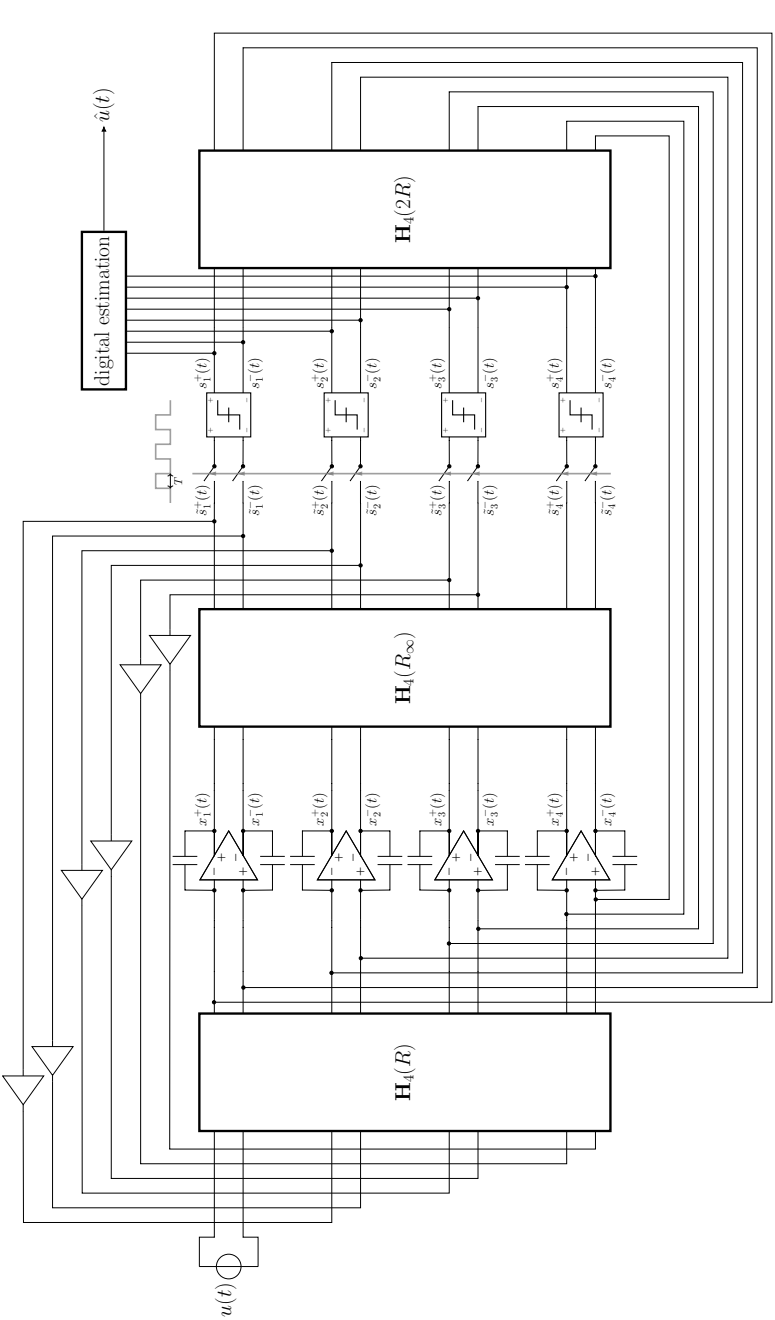


Figure 8.4: Circuit implementation of the control-bounded Hadamard converter for $N = 4$. Alternative implementation for the resistor networks $\boldsymbol{H}_4(R)$ are shown in Figure 8.5 and Figure 8.6. The capacitors in the figure are all of equal size and denoted C. Furthermore, feedback amplifiers represents voltage buffers.

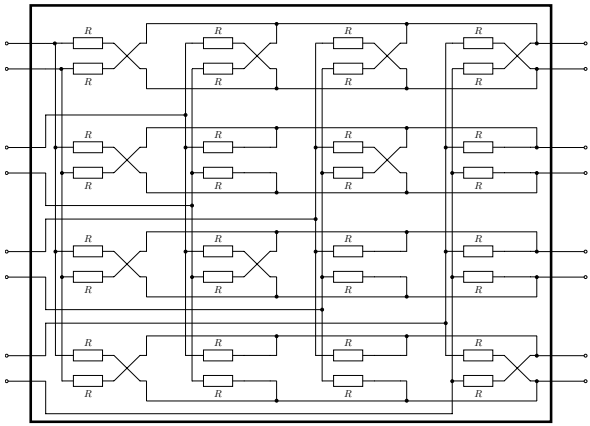


Figure 8.5: A $\boldsymbol{H}_4(R)$ Hadamard resistor network where the k-th differential output is connected to the ℓ-th differential input via the k-th row ℓ-th column resistor pair in the figure.

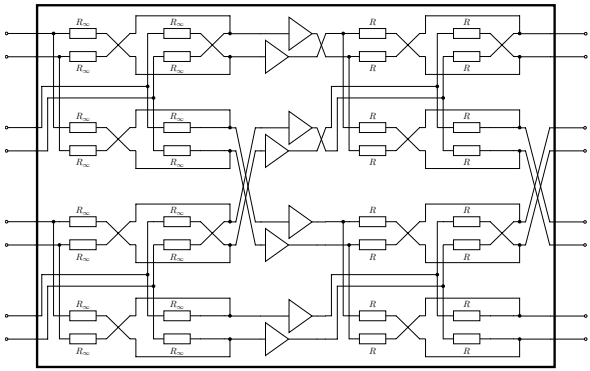


Figure 8.6: A $\boldsymbol{H}_4(R)$ network implemented in the style of a fast Walsh-Hadamard transform where the left hand side terminals are the inputs and the right hand side terminals the outputs of the network. Note that in comparison to Figure 8.5 this implementation requires eight additional voltage buffers.

Notably, both the input and analog feedback paths feed into all of the integrators. A direct consequence is a uniform sensitivity to the involved circuit components, since no signal observation dimension is defined by a single state, or equivalently, a single circuit component. Additionally, each path is connected via N resistors stacked in parallel. This is advantageous since small independent component imperfections are averaged out with respect to the AS's signal dimension. The effect of this averaging becomes dramatic when considering a component mismatch scenario as in Figure 8.7. Here the resistor components of the different architectures are altered during simulation by randomly selecting them from a uniform distribution with a support of 1% deviation from their respective nominal values. In contrast, the digital estimation is done with nominal values.

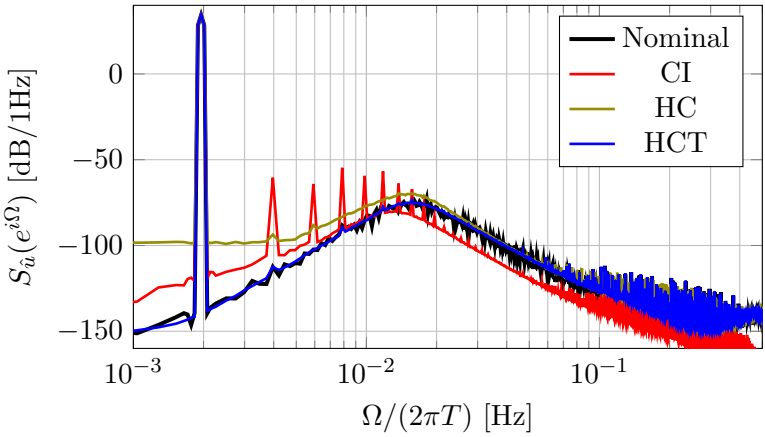


Figure 8.7: Averaged PSD of the estimate $\hat{u}(kT)$ for a mismatch simulation where the resistors of each architecture, are randomly selected with a deviation up to 1% from their nominal values. Furthermore, CI is the chain-of-integrators from Chapter 5, HC is the Hadamard ADC using the Hadamard resistor network from Figure 8.5, and HCT is the Hadamard ADC with the resistor network as in Figure 8.6.

Figure 8.7 shows the average PSD, of the estimate $\hat{u}(t)$, for each architecture as a result of more than 500 mismatch simulations. The nominal case and the mismatched chain-of-integrators converter architecture from

Chapter 5, are also included in the figure.

The simulation results show significant performance degradation for the chain-of-integrators ADC where harmonics, due to the introduced mismatch, are visible in PSD spectrum. The Hadamard converter using the resistor network (marked HC in Figure 8.7) as in Figure 8.5, averages these imperfections and thereby suppresses the harmonic distortion. Additionally, we see the noise floor rising compared to the nominal case. Furthermore, the Hadamard converter using the resistor network from Figure 8.6 (marked HCT in Figure 8.7) is essentially unaffected by the mismatch and almost maintain the noise floor of the nominal case. The robustness of this $\boldsymbol{H}_N$ implementation, as opposed to the one from Figure 8.5, comes from the fact that the circuit not only averages the involved components but furthermore averages products of averages. This will be further described below in Section 8.4.2

8.4.1 Misalignment due to Mismatch

The Hadamard converter architecture has an additional benefit. Namely, that any mismatch creates significant misalignment between the state vector and the DC, which for the digital estimation filter appears similar to that of dithering. This means that in any practical implementation, there is naturally occurring dithering that in turn suppresses limit cycles. The same does not apply for the chain-of-integrators circuit implementation from Chapter 5.

8.4.2 Fast Walsh-Hadamard Transform

In this section, we have showed two versions of the Hadamard resistor network. The latter from Figure 8.6, denoted HCT, uses the concept of a fast Walsh-Hadamard transform to break down the corresponding Hadamard matrix into $\log_2(N)$ steps. Traditionally, the Walsh-Hadamard transform is a divide-and-conquer algorithm, with much resemblance to the fast Fourier transform, which reduces the computational complexity when computing the Walsh spectrum. However, for our application, it is not motivated by the number of necessary additions, but instead from its robustness properties. For the case when $N = 4$, as in the hardware

prototype, the equivalent matrix product can be written out as

$$\boldsymbol{H}_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \tag{8.33}$$

$$= \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}. \tag{8.34}$$

Note that the rightmost matrix corresponds to the resistor network (R_∞) before the voltage buffers in Figure 8.6, and similarly, the left matrix corresponds to the resistor network after the same voltage buffer.

Furthermore, for a general N-th order Hadamard matrix the same principle can be applied recursively as

$$\boldsymbol{H}_N = \begin{pmatrix} \boldsymbol{I}_{N/2} & \boldsymbol{I}_{N/2} \\ \boldsymbol{I}_{N/2} & -\boldsymbol{I}_{N/2} \end{pmatrix} \begin{pmatrix} \boldsymbol{H}_{N/2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_{N/2} \end{pmatrix} \tag{8.35}$$

8.4.3 Power Consumption

It is hard to make sure that the different circuits in Figure 8.7, and more so the corresponding resistor networks, are scaled fairly. One way to argue for the presented comparison is by considering the power consumption. We will next compare the Hadamard ADC to the chain-of-integrators ADC by computing the power consumption for both cases. Specifically, we will expose each ADC to the largest permissible input $\|u(t)\|_\infty = b_u$ combined with the largest permissible initial state $\|\boldsymbol{x}(t)\|_\infty = b_{\boldsymbol{x}}$. Furthermore, we will neglect any power consumption associated with amplifiers.

For the chain-of-integrators, each node's power consumption, as in Figure 5.16, can be written as

$$P_{x_\ell} = \frac{v_{x_{\ell-1}}^2}{R_{\beta_\ell}} + \frac{v_{s_\ell}^2}{R_{\kappa_\ell}} \tag{8.36}$$

$$\leq 2\frac{v_{b_{\boldsymbol{x}}}^2}{R} \tag{8.37}$$

where $b_u = \kappa = b_{\boldsymbol{x}}$, $v_{b_{\boldsymbol{x}}} = \max_t(v_{s_l}(t), v_{x_{\ell-1}}(t))$, and $R = R_{\beta_\ell} = R_{\kappa_\ell}$. This means that the proposed parameter settings result in a stability

margin of $\epsilon = 2$. Subsequently, a N-th order chain-of-integrators ADC could consume up to NP_{x_ℓ} Watt.

Applying the same reasoning to the Hadamard circuit from Figure 8.4, we can write the power consumed with respect to a single maximum input signal and initial AS state as

$$P_{x_\ell} = \sum_{\ell=1}^{N} \frac{v_{\tilde{s}_\ell}^2}{R_\mathrm{H}} + \sum_{\ell=1}^{N} \frac{v_{s_\ell}^2}{\sqrt{N} R_\mathrm{H}}. \tag{8.38}$$

For this to be true we have assumed the R_∞ to consume no power. As already discussed, the Hadamard circuit's resistors are larger than their corresponding chain-of-integrators version. Specifically, from (8.31),

$$R_\mathrm{H} = \sqrt{N} R \tag{8.39}$$

for the same amplification factor β and capacitance C. Furthermore, the control observation $\tilde{\boldsymbol{s}}(t)$ is smaller compared to its equivalent chain-of-integrators version due to the scaling done in Section 8.2. The same applies to the voltage, which can be at most $v_{\tilde{s}_\ell} = \frac{1}{\sqrt{N}} v_{b_\infty}$, whereas $\max_t v_{s_\ell}(t) = v_{b_\infty}$ as before. All these things considered, (8.38) is upper bounded by

$$P_{x_\ell} \leq 2 \frac{v_{b_\infty}^2}{R}, \tag{8.40}$$

i.e., given the stated assumptions, the Hadamard converter consumes the same power as the chain-of-integrators circuit with respect to the analog signal paths, at most.

The computations above were an upper bound on the power consumption since we assumed a worst-case signal and initial state. To get a better understanding of the average consumed power, we next estimate the probability density function of the control observations $\tilde{\boldsymbol{s}}(t)$ when simulating a full-scale sinusoidal input signal. For the chain-of-integrators ADC, the control observation and the actual physical states is the same thing as $\tilde{\boldsymbol{\Gamma}}_\mathrm{H}^\mathsf{T} = \boldsymbol{I}_N$. The estimated probability densities are given in Figure 8.8. From the figure, we conclude the $\sqrt{N} = 2$ scaling difference between the different amplitudes and an average norm that is far from the assumed worst-case scenario, which would be $\|\tilde{s}(t)\|_2 = \sqrt{N}$ or $N^{\frac{1}{4}}$ for each case respectively. This means that both the average and maximum power consumption can be assumed substantially lower than the one given in (8.37) and (8.40).

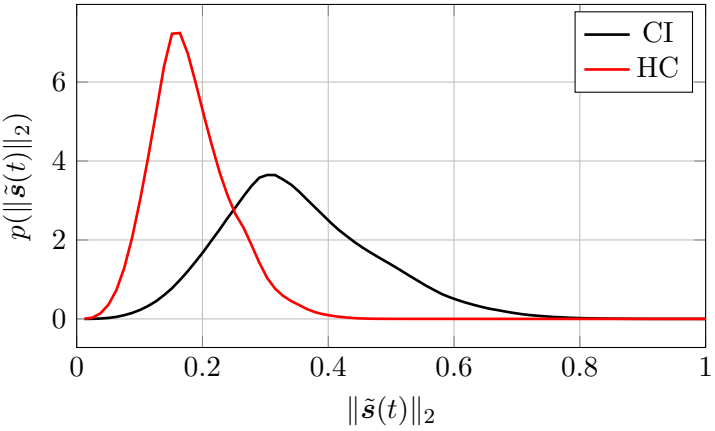


Figure 8.8: The estimated probability density function of the L_2 norm of the control observations $\tilde{\boldsymbol{s}}(t)$, as in (4.5), for the chain-of-integrators and Hadamard converter, respectively. Each ADC is excited with a full-scale sinusoidal input signal.

8.5 Thermal Noise Suppression

We already concluded that rotating the state space, using the Hadamard matrix, essentially distributes all signal paths over all the involved circuit components. More precisely, signals from each node of the equivalent logical chain can be found on all integrators of the AS.

This means that the power invested in the circuit to suppress thermal noise, is consumed uniformly by all integrators. This is certainly a big difference compared to the chain-of-integrators from Chapter 5, where essentially all of the mentioned power would have to be consumed in the first node of the chain to avoid this being the performance bottleneck from a thermal noise point of view.

For the Hadamard ADC this can be taken one step further by not allocating the state space equally among the signal dimensions of the AS. Practically, this means increasing amplification for some signal dimensions with the expense of decreasing the amplification for others (to maintain an effective control). The net effect of not having uniform amplification between nodes of the chain is nominal performance degradation since the AS norm depends on products of amplification. On the other hand,

increasing the amplification for the first signal dimension, i.e., where the input signal enters the circuit, can significantly increase the thermal noise insensitivity with respect to this signal dimension. It is a general fact that, for chain-of-integrator type structures, we are always more sensitive to disturbance at the first node. In other words, the possibility of allocating power consumption unequally over the different signal dimensions could increase overall conversion performance. Especially, for the case when the ADC is limited by thermal noise.

The results of a thermal noise simulation, for a $N = 4$ Hadamard ADC, is shown in Figure 8.9. Simulating thermal noise is outlined in Section 4.8.2.

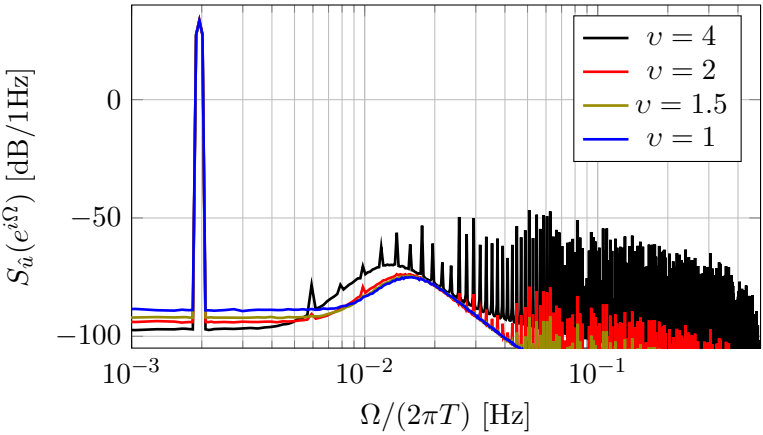


Figure 8.9: The results of a thermal noise simulation, where a $N = 4$ Hadamard ADC performance is limited by the simulated thermal noise. The four different simulations correspond to differently allocated signal dimensions as more space is given to the first signal dimension, and thus rendering better noise suppression capabilities.

For the given simulation the thermal noise has an relative average power of -87 dB/$b_{\boldsymbol{x}}^2$. This is confirmed by the PSD in Figure 8.9 as the blue line has a noise floor corresponding to -87 dB/1Hz. Additionally, this tells us that the simulated system's performance is limited by the added thermal noise source.

The four different simulations vary as

$$\beta_1 = \upsilon \cdot \tilde{\beta} \tag{8.41}$$

$$\beta_2 = \frac{1}{\upsilon} \cdot \tilde{\beta} \tag{8.42}$$

and $\tilde{\beta}$ is normalized such that $\| \left(\beta_1, \ldots, \beta_N\right) \|_1 = N\beta$. The proposed scaling ensure that the local DC remains effective, given that it was effective for $\upsilon = 1$. Note that for $\upsilon = 1$, we retain the previously used scaling for the Hadamard ADC.

From Figure 8.9 we confirm the previously stated tradeoff. Namely, as we give more room to the first dimension, the thermal noise floor decreases at the expense of the nominal performance. Additionally, we notice that the high-frequency DC noise intensifies as we increase the amplification of the first dimension. Finally, we notice that substantial noise suppression can be achieved for relatively small nominal performance loss.

Note that the v parameter does not change the general structure of neither the AS, the DC, or the DE therefore there are no complexity penalties associated with this variable.

In summary, being able to allocate the state space between different signal dimensions can be a powerful tool for tuning the power efficiency of the Hadamard ADC. However, a non-uniform allocation results in a nominal performance degradation as the overall AS amplification is reduced.

8.6 Generalized Transformation

This chapter covered how a chain-of-integrators ADC could be transformed into a Hadamard ADC using a Hadamard matrix $\boldsymbol{H}_N$. This concept more generally applies to any orthonormal matrix $\boldsymbol{H}_N$ and, equally important, these transformations could be applied to any control-bounded ADC's AS and DC.

In summary, transformations that rotate the logical signal dimensions relative to the physical states is a mean to enhance the AS physical properties without changing the nominal performance, i.e., the underlying transfer functions and the DC interaction with the system. We have demonstrated this using the Hadamard transformation. However, there might be more interesting transformations tailored to a given scenario. In

particular, in the case of prior knowledge on the underlying distribution of $\boldsymbol{u}(t)$, one could optimize $\boldsymbol{H}_N$ to further distribute the signal energy among the AS's physical states.

Chapter 9

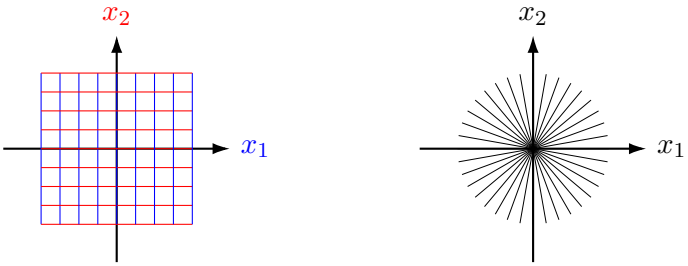
Overcomplete Digital Control

The overcomplete DC concept is inspired by several of the previously mentioned topics such as the higher-order quantizer concept in Section 5.3.1, the self dithering control shown in Figure 5.13, and the misaligned control from Section 8.2.

The the overcomplete DC's objective is to enable a scaling of the complexity of the DC in a distributed way and, by doing so, avoiding implementation bottlenecks. Increasing the complexity of a DC refers to increasing the resolution at which the DC can interact with the AS. This could mean both increasing the order of the quantizers used for observing the AS control observations as well as increasing the total number of possible control contributions. In this context, we do not consider shortening the control period T, i.e., to oversample, as a means of increasing the DC complexity.

The issue related to increasing the complexity of the DC by using higher-order quantizers is that the required component precision also increases. Therefore, this approach becomes impractical even for moderate complexity as the corresponding decision thresholds (in the quantizer) and digital representation references (in the DAC) needs to be implemented with finer and finer precision.

One proposed solution to this problem is to use types of dynamic-element



(a) Higher-order quantization. **(b)** Overcomplete DC.

Figure 9.1: Visualization of the control task for the local DC using higher-order quantizers compared to the overcomplete DC.

matching or mismatch shaping, cf. [25]. Dynamic-element matching effectively averages out imperfections but this comes with significant overhead since it requires many more circuit components (to allow averaging) and additional control logic to administrate it.

As an alternative to using higher-order quantizers, we propose the overcomplete DC principle. This DC aims at increasing complexity while maintaining a relative mismatch sensitivity. Additionally, the DC should be composed of many independently operated scalar DCs to promote a simpler hardware implementation.

9.1 Overlapping Reach

Before further describing the overcomplete DC, we will consider how a local control using a higher-order quantizer divides the control task. In Figure 9.1a, two AS states are shown together with uniform thresholds from a higher-order quantizer. Since the local control controls each state independently, the higher-order quantization thresholds form a square grid where the blue lines correspond to the local control of $x_1(t)$, and similarly, the red ones correspond to $x_2(t)$. Equivalently, higher-order quantization results in thresholds that can be viewed as inner products with affine hyperplanes. As previously mentioned, a bottleneck when using higher-order quantizers is to implement these offsets with sufficient precision. In practice, it is the DAC, and not the quantizer, that is the bottleneck. Regardless, the same analogy applies to the DAC.

The overcomplete DC divides the state space in a different way, as illustrated in Figure 9.1b. Specifically, the corresponding threshold hyperplanes are not affine, as there are no offsets. Furthermore, they are inherently not local, since essentially every physical dimension can be projected into every individual DC's control hyperplane. The same thing can be said for an arbitrary signal dimension. Subsequently, almost every element of the control signal takes part in the control task of every state. As a result, the control task is now divided among many control signals, and thereby the importance of a single control contribution becomes less critical. This results from the cumulative control effort.

When describing the overcomplete DC it is neither the elements of the control signal $\boldsymbol{s}[k]$ vector nor the control contribution $\boldsymbol{s}(t)$ vector that are the focus. In fact, these operate identically as in the local DC, i.e., independently and in synchronization with a global clock. Instead, it is the control input matrix

$$\boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{\gamma}_1, & \ldots, & \boldsymbol{\gamma}_M \end{pmatrix} \in \mathbb{R}^{N \times M} \tag{9.1}$$

and the control observation matrix

$$\tilde{\boldsymbol{\Gamma}} = \begin{pmatrix} \tilde{\boldsymbol{\gamma}}_1, & \ldots, & \tilde{\boldsymbol{\gamma}}_M \end{pmatrix} \in \mathbb{R}^{N \times M}, \tag{9.2}$$

that defines the overcomplete DC. Furthermore, the vectors from (9.1) and (9.2) are N-dimensional column vectors. Figure 9.2 illustrates the general structure of a overcomplete DC.

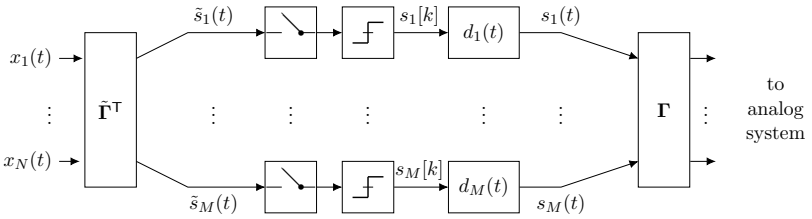


Figure 9.2: An overcomplete DC where we have more independent DC paths M compared to the number of states of the AS N.

The first thing to note is that for an overcomplete DC $M > N$, i.e., we have more independent controls than states of the AS. Also as is shown in Figure 9.2, we only use 1-bit quantizers and thereby 1-bit DACs. Furthermore, $\boldsymbol{\gamma}_\ell$ and $\tilde{\boldsymbol{\gamma}}_\ell$ need not to be distributed as in

Figure 9.1b. However, the column vectors of $\mathbf{\Gamma}$ or $\tilde{\mathbf{\Gamma}}$ are required to form an overcomplete set.

Definition 9.1.1. An overcomplete set of vectors $\mathcal{X}$ is such that, for any subset $\mathcal{Y} = \mathcal{X} \setminus \{\boldsymbol{x}\}$, where $\boldsymbol{x} \in \mathcal{X}$, the span of both $\mathcal{X}$ and $\mathcal{Y}$ are identical.

A trivial example would be to take any of the previously suggested control input matrices $\mathbf{\Gamma}_{\text{old}}$ and duplicate its column vectors. Note that every column vector of $\mathbf{\Gamma}_{\text{old}}$ needs to be duplicated for this to become an overcomplete set of column vectors. The proposed approach certainly battles mismatch as the combined control contribution now will be the average of all duplicates. However, the performance is unchanged as duplication does not affect the control's capabilities in terms of maintaining bounds.

To increase performance, we want each column of $\mathbf{\Gamma}$ to be unique. In fact, the more dissimilar each column vector of $\mathbf{\Gamma}$ is, the better since this divides the reach and overlap of each control dimension. This inspires uniformly spaced polar angles, in polar coordinates, as in Figure 9.1b. Similarly, for a third-order AS, the same could be done by partitioning polar and azimuthal angles uniformly. However, for an N-th order system, it is less clear how an overcomplete DC would most advantageously partition the AS state space or, equivalently, divide the control task.

Finding an Overcomplete Control Set

To find a good partitioned set of overcomplete N-dimensional vectors, we propose using the algorithm given in Algorithm 1. The algorithm starts with a non-empty set of vectors $\mathcal{G}$ and recursively adds vector elements until the set has a cardinality $|\mathcal{G}| = M$.

This algorithm is quite idealistic since the optimization problem from row 6 is a non-convex problem as soon as $\text{rank}(\boldsymbol{T}) = N$. Instead, the solution space has many local minimums, and finding a global optimum quickly becomes unrealistic. In this thesis, this was practically managed by gradient type algorithms with many restarts. In retrospect, the practical implementation results in quite poor numerical precision, and, especially for $M \approx N$, the partitioning ended up far from optimal. This did not seem to be of great importance as for a large M and significant mismatch, having good enough separation between the control vectors gave satisfactory results. This will be evident from the simulations in

Function `OverCompleteSet`($\mathcal{G}$, M)**:**
  input : $\mathcal{G}$ - set of initial vectors.
    M - sought cardinality of $\mathcal{G}$.
  output : overcomplete set of vectors with cardinality M
  while $|\mathcal{G}| < M$ **do**
    `// create matrix from` $\mathcal{G}$
    $\boldsymbol{T} \leftarrow \left(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{|\mathcal{G}|}\right) \in \mathbb{R}^{N \times |\mathcal{G}|}, \boldsymbol{\gamma}_\ell \in \mathcal{G}$
    `// find an unit norm vector that has the largest Euclidean distance to the vectors in` $\mathcal{G}$.
    $\boldsymbol{\gamma}_{\text{new}} \leftarrow \text{argmin}_{\tilde{\boldsymbol{\gamma}}} \frac{\|\boldsymbol{T}^{\mathsf{T}} \tilde{\boldsymbol{\gamma}}\|_2}{\|\tilde{\boldsymbol{\gamma}}\|_2}$
    $\mathcal{G} \leftarrow \mathcal{G} \cup \{\boldsymbol{\gamma}_{\text{new}}\}$
  end
  return $\mathcal{G}$
end

Algorithm 1: Finding an Overcomplete Set

Section 9.4.

9.2 Effective Digital Control

Since the overcomplete DC is no longer local to each AS state vector element, we cannot apply the same recursive procedure, where each node is considered in successive order. Instead, by design, each independent control signal vector element $s_\ell[k]$ is involved in the control of each state of the AS, which makes ensuring an effective control very complicated for large M.

It is clear that since increasing M results in many overlapping control contributions, the magnitude of each column vector of $\boldsymbol{\Gamma}$ must be scaled down accordingly to avoid instability for a given AS. To describe this scaling we denote

$$\boldsymbol{\Gamma} = \kappa \boldsymbol{T} \tag{9.3}$$

where κ is a general scaling and $\boldsymbol{T} \in \mathbb{R}^{N \times |\mathcal{G}|}$ contains the previously mentioned hyperplanes corresponding to each independent DC. Normalizing $\boldsymbol{T}$ as

$$\boldsymbol{T} \leftarrow \left(\boldsymbol{T}\boldsymbol{T}^{\mathsf{T}}\right)^{-\frac{1}{2}} \boldsymbol{T} \tag{9.4}$$

and scaling the global magnitudes as

$$\kappa = \frac{1}{M} \cdot \beta \tag{9.5}$$

has empirically shown to result in effective DC.

Similarly, the control observation matrix $\tilde{\mathbf{\Gamma}}^{\mathsf{T}}$ can be constructed from $\mathbf{\Gamma}$ as

$$\tilde{\mathbf{\Gamma}}^{\mathsf{T}} = \frac{1}{\|\mathbf{\Gamma}\|_2} \cdot \mathbf{\Gamma}^{\mathsf{T}} \tag{9.6}$$

where the matrix norm refers to the largest singular value of $\mathbf{\Gamma}$.

As the scaling above have no stability guarantees whatsoever, extensive simulations have to be conducted to ensure that bounded AS states are maintained. As an example, Figure 9.3 shows the estimated probability density function of the L_∞ norm of the AS state vector for a Hadamard ADC, as in Chapter 8. The columns of $\mathbf{\Gamma}$ are chosen in accordance with Algorithm 1 and the input matrix and the control observation matrix are scaled as in (9.5). Furthermore, the input signal is a sinusoidal signal of the largest permissible amplitude $b_{\boldsymbol{u}}$. The L_∞ norm essentially picks the largest element of the state vector, and the state vector is evaluated at the end of each control period T. From the figure, it is clear that the suggested scaling is overly pessimistic since no element of the AS state vector has any support close to the AS state bound $b_{\boldsymbol{x}}$. Also, in comparison with the default Hadamard ADC configuration, this scaling appears more restrictive.

In Figure 9.3, we considered the worst-case scenario by estimating the probability density function of the L_∞ norm of the AS state vector. Repeating the same setup but estimating with respect to the L_2 norm results in Figure 9.4. The L_2 norm density figure provides additional insights since the density, and thereby also the averaged power consumed by the control signals, concentrates at $\|\boldsymbol{x}(t)\|_2/b_{\boldsymbol{x}} = 0.11$, for the given scaling.

To conclude this section: We have seen a way of scaling the control input and observation matrix, which by a heuristic approach, has been determined to ensure an effective DC. In contrast to all other DCs presented so far, this approach has at this time no known theoretical guarantees.

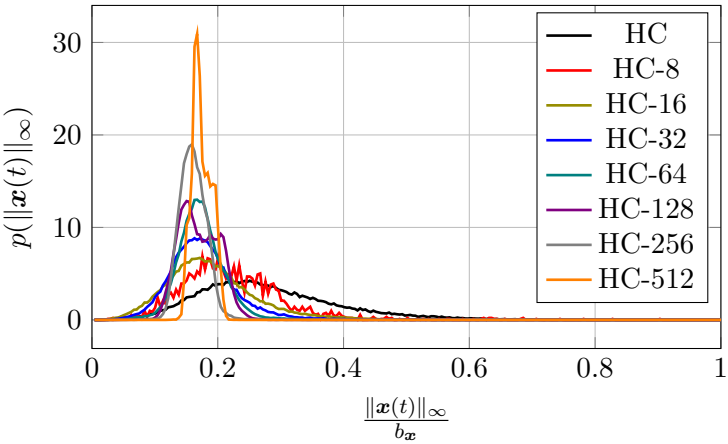


Figure 9.3: Estimated probability density function of $\|\boldsymbol{x}(t)\|_\infty$. Given a full-scale input signal and where t is evaluated at the end of each control period T. HC refers to the default case of a $N = 4$ Hadamard ADC as in Section 8.4. HC-M refers to the same converter using a M-th order overcomplete DC.

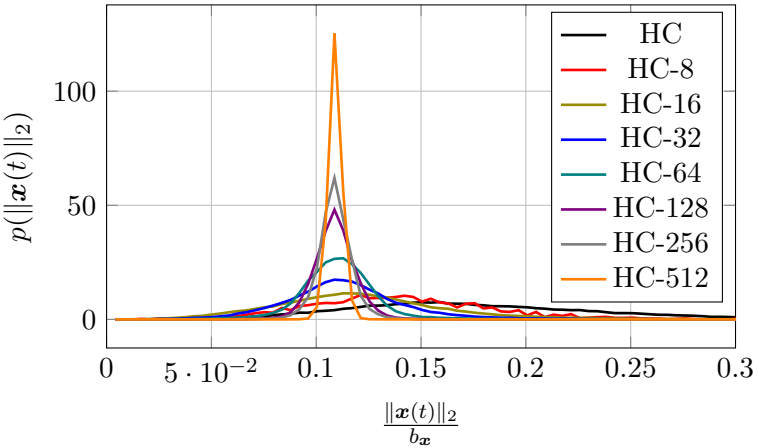


Figure 9.4: Estimated probability density function as in Figure 9.3 but with respect to the L_2 norm.

9.3 Digital Estimator

The overcomplete DC changes the DE filter coefficients as both $\boldsymbol{B}_f$ and $\boldsymbol{B}_b$ from (4.63) and (4.64) are dependent on $\boldsymbol{\Gamma}$. In terms of computational complexity, we repeat the findings of Section 4.3.4, where we concluded that the offline parallelized batch Algorithm 3 from Appendix E utilizes

- $\mathcal{O}(N)$ real-valued scalar multiplications,
- $\mathcal{O}\left(\frac{N(L+M)}{L}\right)$ real-valued scalar additions,
- and requires $\frac{M}{L}$ bits and $4\frac{N}{L}+2$ real-valued scalar values to be kept in memory

per estimated scalar sample. We remind ourselves that in these expressions: L is the number of inputs, N is the number of analog states in the AS, and M is the number of independent scalar controls.

9.4 Mismatch Simulations

The proposed scaling in Section 9.2 has heuristically been determined to give approximately 3 dB increased SNR at each doubling of M. This, in turn, is inferior to the 6 dB, which is expected if the same amount of bits are invested in a higher-order quantizer combined with a local DC. Heuristic experimentation has shown that, when the involved parameters are tuned carefully, SNR improvements above 6 dB per doubling of M can be sustained. However, any such generalized parameter tuning has failed to scale for an arbitrary M and will, therefore, not be reported specifically here.

Regardless of the inferior nominal performance scaling, to fully appreciate the overcomplete DC structure we must consider a mismatch scenario as in Figure 9.5. Here we essentially repeat the mismatch simulations from Figure 8.7, i.e., each component of $\boldsymbol{\Gamma}$ are randomly distorted such that they could differ up to 1% from their nominal values. Figure 9.5, demonstrate that, even though we have substantial component mismatch, the nominal 3 dB SNR improvement is approximately sustained as we double the number of independent DCs M.

Admittedly, the scaling is not completely without problems, as is clear from the harmonics that rise from the noise spectrum as $M > 32$. These could perhaps be addressed by improving the implementation of $\boldsymbol{\Gamma}$ as

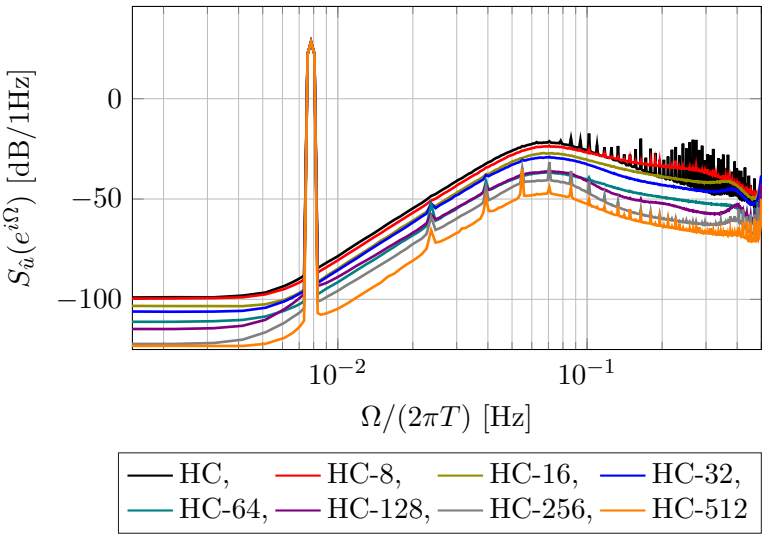


Figure 9.5: PSD of the estimate of $\hat{u}(kT)$ as in Figure 8.7 (HC) where the elements of $\mathbf{\Gamma}$ are subject to mismatch. Additionally, the same AS is equipped with an overcomplete DC using M independent controls (HC-M) that are simulated using components of the same level of imperfection.

done in Figure 8.6 for the Hadamard matrix. More importantly, note that the general conversion noise floor consistently shrinks as we increase the complexity of DC. This is a result of the distributed nature of the overcomplete DC, where, as we increase M, each imperfection gives less influence to the overall control contribution and thereby also the estimate.

Finally we summarize the impact of the M parameter on the resources associated with the AS, DC, and DE. Starting with the DC we recognize that the M parameter determines the number of independent scalar controls. As for the AS M determines the number of columns in $\mathbf{\Gamma}$ and $\tilde{\mathbf{\Gamma}}$, see (9.1) and (9.2). Finally, the computational complexity of the DE is given in Section 9.3 but essentially boils down to a linear increase in the number of scalar additions per estimated scalar sample.

9.5 Controlling a Subspace

In all examples given above, $M > N$, which guarantees overcompleteness for nontrivial cases of $\mathbf{\Gamma}$ as the column rank would exceed the the number of rows. However, it is not necessary for the columns of $\mathbf{\Gamma}$ to span the whole AS state space. The overcomplete concept can equally well be applied to cases where the rank of $\mathbf{\Gamma}$ is less than N. One such example would be for scenarios where $M \leq N$ and the column vectors of $\mathbf{\Gamma}$ still form an overcomplete set. Fundamentally, when the rank of $\mathbf{\Gamma}$ is less than N, an overcomplete DC would imply that the DC only controls a subspace of the AS state space in an overcomplete way.

Particularly, interesting scenarios for controlling only a subspace would be the multi-input case which will be the topic of Chapter 10. Specifically, for a given class of multi-dimensional input signals that mainly excite a subspace of the AS state space (think of correlated input signal dimensions). The DC's $\mathbf{\Gamma}$ matrix could be optimized such that its column vectors span only the mentioned subspace. This would then enable a comparatively more precise DC and possibly reduce the required number of control bits for a given target specification. Another, interesting example would be a DC that could reconfigure its control input matrix $\mathbf{\Gamma}$ and thereby dynamically optimize its control interactions with the AS as the AS state vector's statistics would change over time. Note that extending the DE to compute statistics over $\boldsymbol{x}(t)$ only requires minor modifications to the given DE filter.

Chapter 10

Multi-Input Analog-to-Digital Converters

The examples of the control-bounded converters presented so far were all scalar input ADCs. In this chapter, we will explore multi-input ADCs.

Already in Chapter 4 the generalized control-bounded ADC was described with a multi-channel input, i.e. $\boldsymbol{u}(t) \in \mathbb{R}^L$ for $L > 1$. Any of the previously given examples could be extended to the multi-input scenario where signal dimensions are separately divided among the different inputs. This multi-input scenario is not particularly interesting since it would nominally be the same as converting them using independent ADCs.

However, the overcomplete control from Chapter 9 in combination with the Hadamard type AS from Chapter 8, brings yet another dimension to multi-input ADC conversion.

10.1 Shared Analog System & Digital Control

Imagine an AS with the state space parametrization

$$\boldsymbol{A} = \frac{1}{N}\boldsymbol{H}_N \begin{pmatrix} \boldsymbol{A}_1 & & \\ & \ddots & \\ & & \boldsymbol{A}_L \end{pmatrix} \boldsymbol{H}_N^{\mathsf{T}} \in \mathbb{R}^{N\times N}, \tag{10.1}$$

$$\boldsymbol{B} = \frac{1}{N}\boldsymbol{H}_N \begin{pmatrix} \boldsymbol{B}_1 & & \\ & \ddots & \\ & & \boldsymbol{B}_L \end{pmatrix} \in \mathbb{R}^{N\times L}, \tag{10.2}$$

where $\boldsymbol{A}_\ell \in \mathbb{R}^{N_\ell \times N_\ell}$ and $\boldsymbol{B}_\ell \in \mathbb{R}^{N_\ell}$ are AS parametrizations of converters as presented in previous chapters and $N = \sum_{\ell=1}^{L} N_\ell$. Furthermore, $\boldsymbol{\Gamma}$ and $\tilde{\boldsymbol{\Gamma}}$ are determined by an overcomplete set as in Section 9.1 where $M \gg N$.

The proposed AS essentially has L ASs stacked in parallel in the transformed state space, but with an overcomplete DC. The idea of this approach is that the different input channels share the state space and can, therefore, allocate a larger portion of the AS state space conditioned on the other input channels being less active. A larger allocated AS state space means greater amplification and thus larger dynamical range for the same AS state control bound. Additionally, the overcomplete DC jointly controls the AS state space by its overlapping reach.

This idea promotes adapting amplification and control bounds of the overall ADC with respect to an average multi-channel signal activity, as opposed to designing each respective ADC towards their worst case. An example, for four jointly converter input signals ($L = 4$), is given in Figure 10.1. The figure shows the PSD of the estimated input(s) for a forth-order, $N = 4$, Hadamard ADC with a $M = 32$ overcomplete DC (marked HC-32 in the figure) in comparison to a multi-input signal where four of the mentioned converters ($L = 4$) are combined as described above. Furthermore, three of the inputs, are fixed as $u_2(t) = u_3(t) = u_4(t) = 0$ thereby allowing $u_1(t)$ to allocate a larger portion of the state space. This means $u_1(t)$ can increase its amplitude by a factor $\sqrt{L}$ without affecting the corresponding control bound. From the SNR definition in (3.8), we

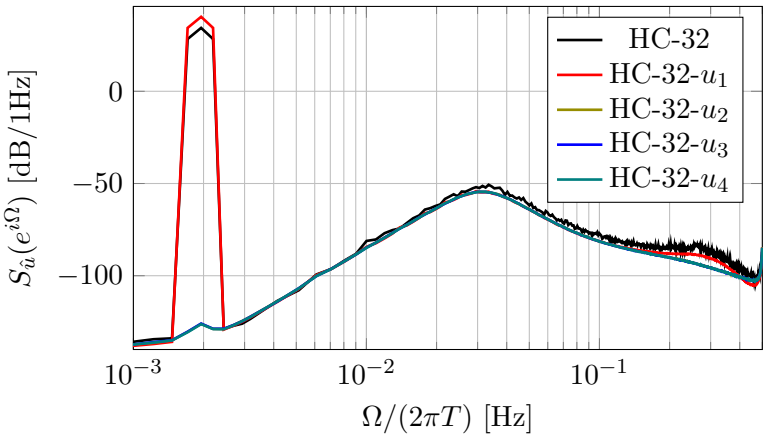


Figure 10.1: PSD of the estimated input(s) $\hat{\boldsymbol{u}}(kT)$ for a Hadamard ADC as in Figure 9.5 in comparison with a multi-input control-bounded ADC. Note that the estimates of HC-32-u_2, HC-32-u_3, and HC-32-u_4 completely overlay each other in the figure and are therefore indistinguishable.

realize that increasing the amplitude by a factor a amounts to a SNR increase of $10\log(a)$ dB.

This can be confirmed from Figure 10.1 as HC-32-l_1 is 6 dB taller than the default overcomplete Hadamard ADC. In other words, the multi-input overcomplete Hadamard ADC can allow a $10\log(L)$ dB larger dynamic range for each input conditioned on the others being zero, i.e.,

$$\text{SNR} \propto L. \tag{10.3}$$

For a large L this means that the amplification and control bounds can be adapted towards $\mathrm{E}[\|\boldsymbol{u}(t)\|_2]$ instead of $L \cdot \max_t \|\boldsymbol{u}(t)\|_\infty$ as long as $\{\|\boldsymbol{u}\|_\infty > \sqrt{L} \cdot b_{\boldsymbol{x}}\}$ remains sufficiently unlikely.

10.2 Adaptive Beamforming ADC

An obvious application of this scaling would be an adaptive beamforming application where the signal lies in a subspace of the input channels and changes over time. For this case, individually converting each input means

that each channel needs to be dimensioned for the worst-case scenario of all signal power residing in a single channel, i.e.chain-of-integrators ADC

$$\|\boldsymbol{u}(t)\|_\infty \leq b_{\boldsymbol{u}} \tag{10.4}$$

for any $t \in \mathbb{R}$. In contrast, the multi-input control-bounded converter only needs to be bounded as

$$\|\boldsymbol{u}(t)\|_2 \leq \sqrt{L} b_{\boldsymbol{u}} \tag{10.5}$$

which results in a L increase in SNR. To illustrate this, consider an input signal

$$\boldsymbol{u}(t) = u(t) \cdot \boldsymbol{v} \in \mathbb{R}^L \tag{10.6}$$

where $u(t)$ is a scalar input as before and the amplitude vector $\boldsymbol{v} = \big(v_1, \ldots, v_L\big)^{\mathsf{T}} \in \mathbb{R}^L$ is normalized, depending on the two different cases described above.

Furthermore, as a post-processing step the ADC estimate $\hat{\boldsymbol{u}}[k]$ is combined into a scalar estimate by projecting it onto $\boldsymbol{v}$ as

$$\hat{u}[k] = \frac{\langle \boldsymbol{v}, \hat{\boldsymbol{u}}[k] \rangle}{\|\boldsymbol{v}\|_2^2}. \tag{10.7}$$

The proposed beamforming setup is simulated with an $\boldsymbol{A}$ and $\boldsymbol{B}$ as in Section 10.1, and all subsystems are identical $N_\ell = 4$ Hadamard ADCs. Furthermore, $L = 8$ and $\boldsymbol{v}$ is randomly generated and scaled according to (10.5). The overcomplete control is designed with $M = 128$, i.e., $M_\ell = 16$ if all channels where converted individually. In comparison, the individual A/D conversion case requires a more restrictive amplification scaling since the max signal power could, however unlikely, be contained in a single converter. The results are shown in Figure 10.2. As can be seen from the figure there is a substantial benefit, in this case ≈ 21 dB increase in SNR, when doing joint A/D conversion compared to converting each channel individually. Note that Figure 10.2 is the averaged result of more than a 100 random vector realizations $\boldsymbol{v}$.

10.3 Mismatch Sensitivity

A fair concern with the proposed multi-input ADC is its sensitivity to mismatch. Specifically, as for this ADC conversion strategy, multiple

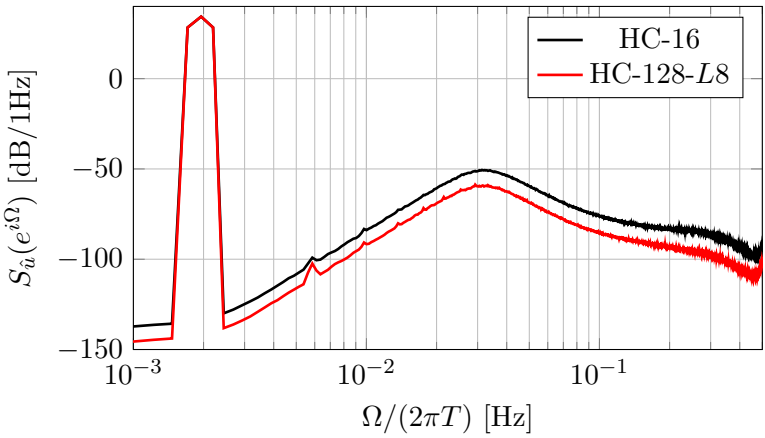


Figure 10.2: PSD of $\hat{u}(kT)$ for a beamformed signal, as in (10.7). The figure shows the relative advantage of converting multiple input channels jointly in comparison with individual conversion. Note that both these simulations use the same amount of independent DCs and AS states per scalar input. However, as the number of scalar additions in the DE scales quadratically with M HC-128-$L8$ is more computationally demanding for the DE than HC-16 per scalar input.

input signals share the same circuit components, non-ideal circuitry might cause performance degradation as the different signal components "leak" into each other. On the other hand, both the Hadamard ADC architecture and the overcomplete DC were motivated by, among other things, their robustness to component mismatch. To resolve this matter, a mismatch simulation is given in Figure 10.3. The mismatch simulation is conducted with four full-scale sinusoidal input signals with four different frequencies and a 1% tolerance on the components corresponding to $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{\Gamma}$ for a circuit setup similar to that of Figure 8.4 but with an overcomplete DC.

The figure confirms signal leakage between the input channels as small peaks rise in the PSD of each input at the frequencies of the other inputs signals. On the other hand, the overall mismatch performance, including the mentioned leakage, is still relatively good. For example, consider

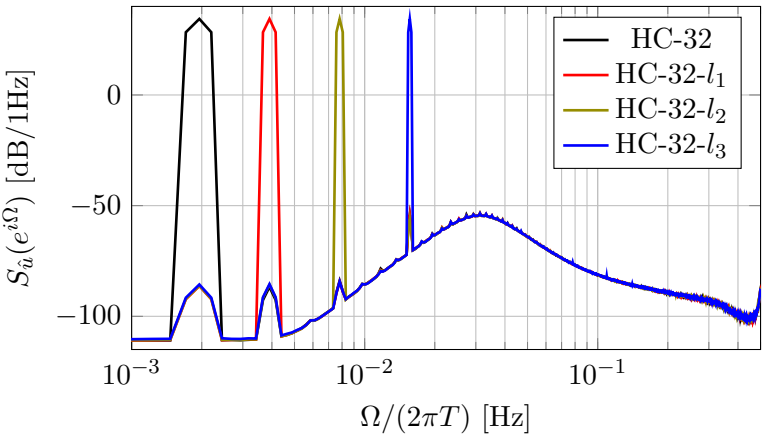


Figure 10.3: Mismatch simulation with a 1% variation of the components in the corresponding circuit components of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{\Gamma}$ for a ADC setup as in Figure 10.1.

the PSD for a chain-of-integrators ADC simulated with the same 1% mismatch as in Figure 8.7.

The severity of the inter-channel leakage in the estimate also depends on the application. As an example, for an adaptive beamforming application, as in Section 10.2, the increased dynamic range would not be heavily affected by the mismatch. This is due to the fact that any such mismatch can be seen as a distortion to the amplitude vector $\boldsymbol{v}$, see (10.6). Subsequently, the distortion is accounted for as this vector is estimated in a digital post-processing step.

In summary, we conclude that the multi-input ADC is an interesting concept in terms of efficiently distributing the conversion problem and resources. However, due to the increased sensitivity to mismatch-induced leakage between estimation channels, it might not be advantageous for every A/D conversion application.

10.4 Fundamental Resource Scaling

The multi-input ADC enables circuit resource sharing as multiple, otherwise independent, conversion processes are combined. Next, we summa-

rize how this impacts the resources used by the AS, DC, and DE. The number of integrators, or equivalently AS states, remains constant per scalar input signal. However, the number of connections, the $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ matrix, grows quadratically in the number of states. The same applies to the input vector $\boldsymbol{B} \in \mathbb{R}^{N\times L}$. The DC's number of independent controls also remains constant per scalar input signal. However, the number of elements in the control contribution matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{N\times M}$, and the control observation matrix $\tilde{\boldsymbol{\Gamma}} \in \mathbb{R}^{M\times N}$ might substantially increase. We remind the reader that the mentioned increase in connections above does not necessarily make the system more power-consuming as the many connections are normalized in an $\mathcal{L}_2$ sense.

Finally, the computational complexity, per scalar input signal, of the DE, increases as follows from the analysis in Section 4.3.4. Specifically, the number of scalar additions scales quadratically in M and therefore as M^2/L, per scalar input signal, when we combine multiple converters into one. Interestingly, the number of scalar multiplications remains constant per scalar input signal. It is essential to note that this additional computational complexity did not account for the fact that the multi-input ADC also potentially enables massive gains, see Section 10.2. Therefore, the actual cost per benefit would be application-specific and a topic that requires more careful consideration.

Chapter 11

Reciprocal Problem

The control-bounded A/D conversion concept that has been the focus throughout all previous chapters can be adapted for the purpose of D/A conversion.

11.1 Control-Bounded Digital-to-Analog Conversion

It is already clear from previous examples that the AS is capable of creating complex analog waveforms via the interaction of a DC. Furthermore, the more complex these ASs become, the richer the variety of the analog signals they produce, and this motivates the use of such ASs, combined with DC, for the process of D/A conversion. By adapting the DC and DE we can transform the previously proposed control-bounded ADCs into control-bounded DACs. The general system description is given in Figure 11.1.

The control-bounded DAC is operated by reversing the control-bounded ADC concept. Specifically, based on a sequence of samples $\boldsymbol{u}[k]$, we estimate what the AS state vector must be such that the continuous-time output $\hat{\boldsymbol{u}}(t)$ approaches $\boldsymbol{u}[k]$ at the given sample times $kT_s, (k+1)T_s, ...$ The estimated AS state vector is realized using a DC which ensures a bounded error between the actual state vector and the sought trajectory. The concept is illustrated in Figure 11.1. We briefly summarize the DE,

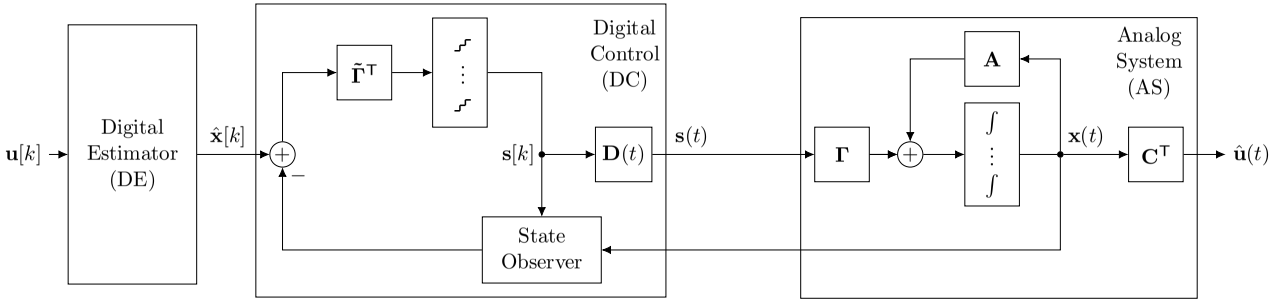


Figure 11.1: The control-bounded DAC setup where for a given sequence samples $\boldsymbol{u}[k]$ a continuous-time analog version $\hat{\boldsymbol{u}}(t)$ is created. Note that the state observer, inside of the DC, might optionally contain an ADC as is further discussed in Section 11.3.

DC, and AS role as:

- The DE converts the target samples $\boldsymbol{u}[k]$ into a corresponding AS state vector $\boldsymbol{x}(t)$ evaluated at the end of each control period T. The state vector is the target trajectory for the DC.
- The DC compares the target AS state vector to an estimated version. Consequently, the DC decides on a control signal $\boldsymbol{s}[k]$ that minimizes the error for each control period.
- The AS takes the control contributions and thereby excites the AS state vector such that continuous-time analog output approximates the specified trajectory. Specifically, the AS output $\hat{\boldsymbol{u}}(t)$ approximates the target samples at the sampling times.

11.2 Digital Estimator

The task of the DE is to solve an inversion problem where we assume a fictional continuos-time input signal $\boldsymbol{y}(t)$ being fed into the AS. The input signal is such that the resulting fictional output is an approximation of the target samples $\boldsymbol{u}[k]$.

For this estimation problem, it is not the fictional input signal $\boldsymbol{y}(t)$ that is of primary interest but instead the resulting state trajectories $\boldsymbol{x}(t)$. In a later step, the DC will control the AS such that it follows these state trajectories. Furthermore, as $\boldsymbol{x}(t)$ is a continuous-time object, we sample it at the end of each control period T. In other words, the output of the DE is the sampled estimated state trajectories.

The estimation task can be written as an optimization problem

$$\begin{aligned}\underset{\boldsymbol{x}(t)}{\operatorname{argmax}} \sum_{k\in\mathbb{Z}} \big(\boldsymbol{u}(kT_s) + \boldsymbol{C}^{\mathsf{T}}\boldsymbol{x}(kT_s)\big)^{\mathsf{H}} \big(\boldsymbol{u}(kT_s) + \boldsymbol{C}^{\mathsf{T}}\boldsymbol{x}(kT_s)\big) \\ + \eta^2 \int_{-\infty}^{\infty} \boldsymbol{y}(\tau)^{\mathsf{T}}\boldsymbol{y}(\tau)\,\mathrm{d}\tau \end{aligned} \tag{11.1}$$

such that

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{y}(t). \tag{11.2}$$

Estimating $\boldsymbol{x}(t)$ implicitly also entails estimating the fictional input signal $\boldsymbol{y}(t)$. Furthermore, the optimization problem can be solved, at specific times $\ldots, (k-1)T, kT, \ldots$, using a Kalman smoothing algorithm. Specifically, we will use the BIFM factor graph message passing rules,

cf. [34], given in Appendix D.2. This algorithm is closely related to the one presented in Section 4.3.2. In particular, it is in the same computational complexity class. Furthermore, the algorithm is parameterized by some model assumptions: The input $\boldsymbol{y}(t)$ is modeled as a zero-mean stationary stochastic process where the process, integrated over a unit time step, results in the covariance matrix

$$\boldsymbol{\Sigma}_{\boldsymbol{u}} = \frac{1}{\eta} \cdot \boldsymbol{I}_L. \tag{11.3}$$

where L is the dimension of the fictional input signal, i.e., $\boldsymbol{y}(t) : \mathbb{R} \to \mathbb{R}^L$.

Similarly, the input samples $\boldsymbol{u}[k]$ are modeled as being noisy observations where the noise is assumed i.i.d. zero-mean Gaussian random variables with a covariance matrix

$$\boldsymbol{\Sigma}_{\boldsymbol{z}} = \eta \cdot \boldsymbol{I}_M \tag{11.4}$$

where M is the dimension of the samples, i.e., $\boldsymbol{u}(kT_s) \in \mathbb{R}^M$.

Estimating the posterior mean of the state vector $\boldsymbol{x}(kT_s)$ reduces to a linear filter split up in a forward and backward recursion as

$$\boldsymbol{\xi}[k] = \boldsymbol{A}_b \boldsymbol{\xi}[k+1] + \boldsymbol{B}_b \boldsymbol{y}[k] \tag{11.5}$$

$$\boldsymbol{x}(kT_s) = \boldsymbol{A}_f \boldsymbol{x}((k-1)T_s) + \boldsymbol{B}_f \boldsymbol{\xi}[k] \tag{11.6}$$

where the $\boldsymbol{A}_f$, $\boldsymbol{A}_b$, $\boldsymbol{B}_f$, and $\boldsymbol{B}_b$ follow from Appendix D.2 as

$$\boldsymbol{A}_b = e^{\boldsymbol{A}^\mathsf{T} T_s} \left(\boldsymbol{I}_N - \boldsymbol{W} \left(\boldsymbol{\Sigma}_{\boldsymbol{u},T_s}^{-1} + \boldsymbol{W} \right)^{-1} \right) \in \mathbb{R}^{N \times N} \tag{11.7}$$

$$\boldsymbol{A}_f = \boldsymbol{A}_b^\mathsf{T} \in \mathbb{R}^{N \times N} \tag{11.8}$$

$$\boldsymbol{B}_b = \boldsymbol{C}^\mathsf{T} \boldsymbol{\Sigma}_{\boldsymbol{z}}^{-1} \in \mathbb{R}^{N \times M} \tag{11.9}$$

$$\boldsymbol{B}_f = \left(\boldsymbol{\Sigma}_{\boldsymbol{u},T_s}^{-1} + \boldsymbol{W} \right)^{-1} \in \mathbb{R}^{N \times N} \tag{11.10}$$

and $\boldsymbol{W}$ is the solution to the discrete-time algebraic Riccati equation

$$\boldsymbol{W} = e^{\boldsymbol{A}^\mathsf{T} T_s} \boldsymbol{W} e^{\boldsymbol{A} T_s} - e^{\boldsymbol{A}^\mathsf{T} T_s} \boldsymbol{W} \left(\boldsymbol{\Sigma}_{\boldsymbol{u},T_s}^{-1} + \boldsymbol{W} \right)^{-1} \boldsymbol{W} e^{\boldsymbol{A} T_s} + \boldsymbol{C} \boldsymbol{\Sigma}_{\boldsymbol{z}}^{-1} \boldsymbol{C}^\mathsf{T}. \tag{11.11}$$

The expression above originates from the work presented in [4, 5, 18, 35]. A brief derivation from the underlying statistical estimation problem is given in Appendix D.2.

Signal Transfer Function

The continuous-time input estimation enforces a certain type of PSD on the final estimate; as the message passing reduces to a Wiener filter we can formulate a transfer function between the target samples $\boldsymbol{u}[k]$ and the estimated fictional input signal $\boldsymbol{y}(t)$. Namely,

$$\boldsymbol{Y}(\omega) = \underbrace{\tilde{\boldsymbol{G}}(\omega)^{\mathsf{H}} \left(\tilde{\boldsymbol{G}}(\omega)\tilde{\boldsymbol{G}}(\omega)^{\mathsf{H}} + \eta^2 \boldsymbol{I}_N\right)^{-1}}_{\boldsymbol{H}_{\mathrm{DAC}}(\omega)} \boldsymbol{U}(e^{i\omega T}) \tag{11.12}$$

where

$$\tilde{\boldsymbol{G}}(\omega) \triangleq \sum_{\ell \in \mathbb{Z}} \boldsymbol{G}\left(\omega - \ell \frac{2\pi}{T}\right) \tag{11.13}$$

as shown in [6] (Section 5.3) and $\boldsymbol{Y}(\omega)$ is the continuous-time Fourier transform of the estimated fictional input signal $\hat{\boldsymbol{y}}$. Note that the ATF matrix $\boldsymbol{G}(\omega)$ of the AS is defined as previously stated in (4.7). Additionally, we see the implicit aliasing resulting from mapping the discrete-time samples into a continuous-time version. The effect of aliasing in the estimate is visible in later simulations, cf. Figure 11.3.

The fictional input signal will appear at the AS output $\hat{\boldsymbol{u}}(t)$ as a filtered version. Specifically, the filter

$$\mathrm{STF}(\omega) = \boldsymbol{G}(\omega)\boldsymbol{H}_{\mathrm{DAC}}(\omega) \tag{11.14}$$

determines the transfer function between the discrete-time target sample signal $\boldsymbol{u}[k]$ and the continuous-time output of the AS $\hat{\boldsymbol{u}}(t)$.

This insight suggest that the AS can be seen as an interpolation kernel that the DE uses to interpolate the discrete-time samples into a continuous-time version. Furthermore, the DC realizes the interpolation by using a digital control loop and control contributions.

11.3 Digital Control

As for the control-bounded ADC, the control-bounded D/A converter's DC operates in a discrete-time setting. Additionally, as shown in Figure 11.1, the DC can be recognized as a conventional control system from control theory. Specifically, for a given estimated target trajectory $\hat{\boldsymbol{x}}[k]$ given by the DE, the control produces a control signal $\boldsymbol{s}[k]$ which tries to minimize the error between the target and the observed state vector.

A significant difference to the previously seen DCs is the state observer. The objective of the state observer is to produce an estimated version of the AS state vector. This is a well-known concept in control-theory, and traditionally involves an additional Kalman estimation step. One extreme would be to completely base the state estimate on the previously applied control signals $\boldsymbol{s}[k]$ together with the prior knowledge of the system parameters of the AS. This approach is attractive from a computational point of view, but also has its caveats. Specifically, as it leaves the AS as an open-loop system, minor modeling errors might result in AS instabilities. These issues could partially be resolved by ensuring the AS to be stable.

Alternatively, the state observer could additionally incorporate conventional ADCs to observe the actual AS state vector. The latter approach stabilizes the AS by a control loop, but this results in a significantly more complex DC. Furthermore, using AS state observation would require the DC to incorporate a conventional ADC as well as a more computationally demanding state observer algorithm.

Except for the state observer, the DC operates much like in the ADC counterpart. Specifically, after subtracting the state estimate from the sought state trajectory, multiple simple independent controls determines the control signal $\boldsymbol{s}[k]$.

The Conversion Error

A key feature of the proposed DE scheme is that the fictional input signal $\tilde{\boldsymbol{u}}(t)$ is substantially smaller than any target samples $\boldsymbol{u}[k]$, given $\|\boldsymbol{G}(\omega)\|_2 \gg 1$ in the frequency band of interest. This can be confirmed from (11.12) as for a scalar input and output system

$$\|\boldsymbol{H}_{\mathrm{DAC}}(\omega)\|_2^2 \approx \frac{1}{\|\tilde{\boldsymbol{G}}(\omega)\|_2^2}. \tag{11.15}$$

Based on this insight, the DC can be designed with much smaller amplification, i.e., the scaling of the $\boldsymbol{\Gamma}$ matrix. Specifically, for a sequence of target samples $\boldsymbol{u}[k]$, bounded by $b_{\boldsymbol{u}}$, the fictional input signal $\tilde{\boldsymbol{u}}(t)$ will be bounded as

$$b_{\tilde{\boldsymbol{u}}} = \|\boldsymbol{H}_{\mathrm{DAC}}(\omega_{\mathrm{crit}})\|_\infty \cdot b_{\boldsymbol{u}} \tag{11.16}$$

where ω_{crit} is the frequency with the smallest amplification $\|\boldsymbol{G}(\omega)\|_2$ in the frequency band of interest. To demonstrate this, assume we would

apply the local DC from Section 5.3. Subsequently, the largest state error

$$\boldsymbol{x}_\epsilon(t) = \hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t) \tag{11.17}$$

would also be bounded as $\|\boldsymbol{x}_\epsilon(t)\|_\infty \leq b_{\tilde{\boldsymbol{u}}}$.

For the control-bounded DAC, this illustrates how the conversion performance manifests itself as the bound of the state error is approximately inversely proportional to the overall AS amplification.

11.4 Analog System

The AS from Figure 11.1 resembles the AS from Figure 4.2. However, the control-bounded DAC lacks the input matrix $\boldsymbol{B}$, since this is a conceptual quantity only relevant to the DE. Furthermore, for the DAC the signal observation matrix $\boldsymbol{C}^\mathsf{T}$ is an actual matrix, and is a part of the hardware of the AS.

Additionally, as the AS input has changed roles with the AS output, the signal observation matrix $\boldsymbol{C}^\mathsf{T} \in \mathbb{R}^{L\times N}$ has a different role since it essentially mixes the N analog states into L analog output signals.

11.5 Performance Measure

The performance measure of the control-bounded DAC is closely related to the ADC one from Section 4.4. Specifically the signal power can be determined as in (4.84) when considering the STF from (11.14).

Similarly, the conversion error follows as

$$\begin{aligned}\boldsymbol{P}_{\boldsymbol{\epsilon}} &\triangleq \mathrm{E}\big[\boldsymbol{\epsilon}(t)^2\big] && (11.18)\\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \boldsymbol{C}^\mathsf{T}\boldsymbol{S}_{\boldsymbol{x}_\epsilon\boldsymbol{x}_\epsilon^\mathsf{T}}(\omega)\boldsymbol{C}\,\mathrm{d}\omega && (11.19)\end{aligned}$$

where $\boldsymbol{x}_\epsilon(t)$ is assumed a stationary stochastic process with PSD matrix

$$\boldsymbol{S}_{\boldsymbol{x}_\epsilon\boldsymbol{x}_\epsilon^\mathsf{T}}(\omega) \triangleq \int_{-\infty}^{\infty} \mathrm{E}\big[\boldsymbol{x}_\epsilon(t)\boldsymbol{x}_\epsilon(t+\tau)^\mathsf{T}\big]\, e^{-i\omega\tau}\, d\tau. \tag{11.20}$$

For the sake of analysis we once more make a bandlimited assumption on the spectrum of the AS state vector error $\boldsymbol{x}_\epsilon(t)$, i.e,

$$\boldsymbol{P}_\epsilon = \frac{1}{2\pi}\int_{\omega\in\mathcal{B}} \boldsymbol{C}^\mathsf{T}\boldsymbol{S}_{\boldsymbol{x}_\epsilon\boldsymbol{x}_\epsilon^\mathsf{T}}(\omega)\boldsymbol{C}\,\mathrm{d}\omega \tag{11.21}$$

where $\mathcal{B}$ is defined as in (3.10). Additionally, we assume a flat spectrum of $\boldsymbol{S}_{\boldsymbol{x}_\epsilon \boldsymbol{x}_\epsilon^\mathsf{T}}$ in the frequency band of interest $\mathcal{B}$, i.e.,

$$\boldsymbol{S}_{\boldsymbol{x}_\epsilon \boldsymbol{x}_\epsilon^\mathsf{T}}(\omega) \approx \frac{\sigma^2_{b_{\tilde{\boldsymbol{u}}}|\mathcal{B}}}{\|\boldsymbol{G}(\omega_{\mathrm{crit}})\|_2^2} \boldsymbol{I}_M \tag{11.22}$$

where we have indicated the impact of the much smaller DC amplification. It follows that

$$\boldsymbol{P}_\epsilon \approx \frac{\sigma^2_{b_{\tilde{\boldsymbol{u}}}|\mathcal{B}}|\mathcal{B}|}{2\pi\|\boldsymbol{G}(\omega_{\mathrm{crit}})\|_2^2} \boldsymbol{C}^\mathsf{T}\boldsymbol{C} \tag{11.23}$$

which confirms that the combination of an AS generating large amplification in the frequency band of interest in combination with a DC structure that ensures a tight bound for the state vector results in better DAC performance. By comparing (4.94) to (11.23) we can also see that the D/A conversion error behaves similarly as in the case of the control-bounded ADC.

11.6 Chain-of-Integrators Digital-to-Analog Converter

Next, we will demonstrate an example of a control-bounded DAC. Specifically, we will adapt the chain-of-integrators ADC from Chapter 5 into a control-bounded DAC. The local nature of the chain-of-integrators is maintained as each AS state is controlled individually by a local DC based on the DE state estimate. Furthermore, we can repeat the recursive bounded-input bounded-output concept from Section 5.3.1, to ensure an effective DC.

To demonstrate this, we assume an AS as in Figure 5.1 with a

$$\boldsymbol{A}_{\mathrm{DA}} = \begin{pmatrix} 0 & & & \\ \beta & 0 & & \\ & \ddots & \ddots & \\ & & \beta & 0 \end{pmatrix} \in \mathbb{R}^{N\times N} \tag{11.24}$$

$$\boldsymbol{B}_{\mathrm{DA}} = \begin{pmatrix} \beta & 0 & \cdots & 0 \end{pmatrix}^\mathsf{T} \in \mathbb{R}^{N\times 1} \tag{11.25}$$

$$\boldsymbol{\Gamma}_{\mathrm{DA}} = \begin{pmatrix} \beta\kappa & & & \\ & \ddots & & \\ & & \beta\kappa \end{pmatrix} \in \mathbb{R}^{N\times N} \tag{11.26}$$

$$\boldsymbol{C}^{\mathsf{T}} = \begin{pmatrix} 0 & \dots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{1\times N}. \tag{11.27}$$

As the given parameter settings imply a scalar input and output, we will refer to the input signal as $\hat{u}(t)$ and the ATF matrix as $G(\omega)$ to indicate their scalar nature.

Furthermore, we assume the target samples $u[k] \in [-1, 1]$ and as indicated by the expression above $\beta_1 = \cdots = \beta_N = \beta$, $\rho_1 = \cdots = \rho_N = 0$, and $\kappa_1 = \cdots = \kappa_N = \kappa$. For an output target bounded between ± 1, the fictional input signal $y(t)$ must be substantially smaller. This follows from the Wiener filter perspective as

$$\begin{aligned} Y(\omega) &= \boldsymbol{H}_{\mathrm{DAC}}(\omega) U(e^{i\omega T_s}) && (11.28)\\ &= \frac{\bar{\tilde{G}}(\omega)}{|\tilde{G}(\omega)|^2 + \eta^2} U(e^{i\omega T_s}) && (11.29)\\ &\approx \frac{1}{G(\omega)} U(e^{i\omega T_s}) && (11.30)\\ &= \left(\frac{i\omega}{\beta}\right)^N U(e^{i\omega T_s}) && (11.31) \end{aligned}$$

where we have assumed $\|\tilde{G}(\omega)\|_2^2 \gg \eta^2$ and approximated $\tilde{G}(\omega) = 0$ for $\omega \geq 2\pi/T$.

The DC could bound this input at the first node of the chain, using a $|\kappa| \approx 1/|G(\omega_{\mathrm{crit}})|$. The chosen control amplification, in combination with a stability margin chosen as before, results in $b_{\boldsymbol{x}} = \kappa$.

As in the case for the chain-of-integrators ADC, each node of the chain can then be bounded recursively, resulting in a bound on the last AS state error. For the given $\boldsymbol{C}$, the same bound can be sustained for the error on the signal observation $\hat{u}(t)$, which then is bounded by $b_{\boldsymbol{x}}$. In other words, the control-bounded DAC can maintain an error signal bounded by $b_{\boldsymbol{x}} \approx b_{\boldsymbol{u}}/|G(\omega_{\mathrm{crit}})|$ around the given state trajectory.

Simulation

To demonstrate this principle we simulate the chain-of-integrators DAC for $N = 5$, $\beta T = 0.5$, $\mathrm{OSR} = 16$, and $\kappa = -1/G(\omega_{\mathrm{crit}})$. The simulation results are shown in Figure 11.2. From the figure, we see the full-scale input sinusoidal and the noise floor for both the estimated and simulated PSD. Interestingly, the estimated PSD reveals aliasing terms at higher frequencies of the spectrum. These aliasing terms originates from the

estimation task in the DE, and have nothing to do with the DC or AS operation. This can be confirmed by plotting the NTF and STF as in (11.12) and (11.14) for the chain-of-integrators DAC. The results is shown in Figure 11.3. From this figure, we recognize aliasing at multiples of the sampling frequency.

11.7 Control-Bounded Transceivers

Another interesting application is that of a communication system. In this setting the objective is to transmit a sequence of digital representations to another digital domain using an communication channel. As the communication channel is inherently analog, the sequence of digital representations first needs to be converted into the analog domain, transmitted over some channel and then converted back into its digital form. It turns out that for this specific application, the A/D and D/A conversion performance can be improved by matching a control-bounded ADC to a control-bounded DAC.

A basic communication setting is outlined in Figure 11.4. As opposed to before, it is not the analog signals, $\hat{\boldsymbol{y}}(t)$ and $\breve{\boldsymbol{y}}(t)$, that are the prime focus here. Instead, the combined A/D and D/A conversion task is to represent $\boldsymbol{u}[k]$ (using a waveform encoder and an ADC) as an analog signal that is easily converted back into the digital domain (by the DAC and waveform decoder) as shown in Figure 11.4.

The control-bounded converter principle allows us to incorporate the waveform encoder and decoder into the conversion process.

It turns out that a joint conversion scheme, i.e., when the ASs of the ADC and DAC are matched, results in better conversion performance in comparison to two independent conversion schemes. To see this, consider two ASs. The first one belongs to the DAC and is defined by the system of ODEs

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{\mathrm{DAC}}\boldsymbol{x}(t) + \boldsymbol{B}_{\mathrm{DAC}}\tilde{\boldsymbol{u}}(t) + \boldsymbol{\Gamma}_{\mathrm{DAC}}\boldsymbol{s}(t) \tag{11.32}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}_{\mathrm{DAC}}^{\mathsf{T}}\boldsymbol{x}(t). \tag{11.33}$$

Similarly, the second one belongs to the ADC and can be defined by the ODEs

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{\mathrm{ADC}}\boldsymbol{x}(t) + \boldsymbol{B}_{\mathrm{ADC}}\breve{\boldsymbol{y}}(t) + \boldsymbol{\Gamma}_{\mathrm{ADC}}\boldsymbol{s}(t) \tag{11.34}$$

$$\hat{\breve{\boldsymbol{y}}}(t) = \boldsymbol{C}_{\mathrm{ADC}}^{\mathsf{T}}\boldsymbol{x}(t). \tag{11.35}$$

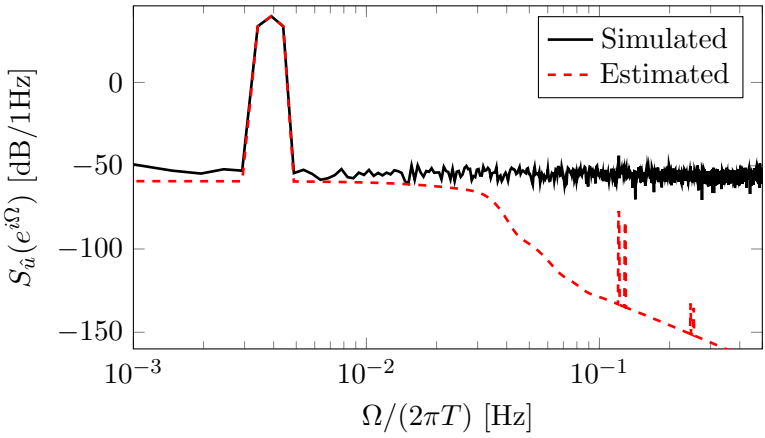


Figure 11.2: The PSD of $\hat{u}(kT)$ for a simulated chain-of-integrators DAC. The simulated $u(kT)$ refers to the sampled output of AS as a result of the control contributions $\boldsymbol{s}(t)$. Similarly, the dashed line corresponds to the estimated output by the DE.

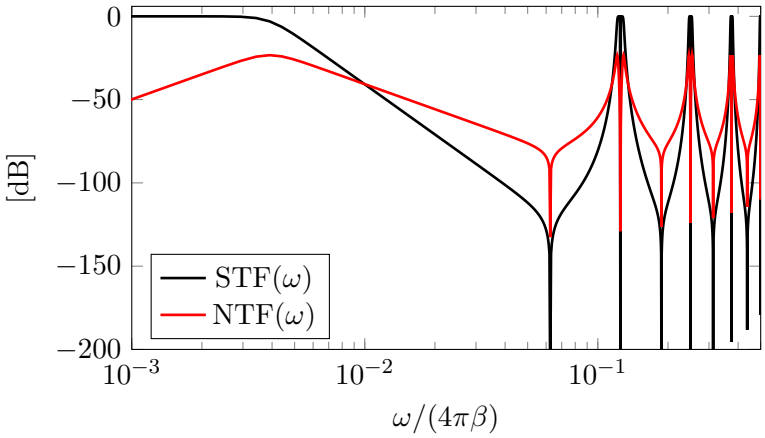


Figure 11.3: NTF and STF of chain-of-oscillator DAC for an OSR = 16.

Figure 11.4: A simplified view of a communication scenario describing the steps involved to transmit a digital signal over an analog domain.

We match the two ASs by setting

$$\boldsymbol{A}_{\mathrm{DAC}} = -\boldsymbol{A}_{\mathrm{ADC}}^{\mathsf{T}} \tag{11.36}$$

$$\boldsymbol{B}_{\mathrm{DAC}} = \boldsymbol{C}_{\mathrm{ADC}} \tag{11.37}$$

$$\boldsymbol{C}_{\mathrm{DAC}}^{\mathsf{T}} = -\boldsymbol{B}_{\mathrm{ADC}}^{\mathsf{T}} \tag{11.38}$$

The control input matrices $\boldsymbol{\Gamma}_{\mathrm{ADC}}$ and $\boldsymbol{\Gamma}_{\mathrm{DAC}}$ are such that we can ensure an effective DC for both the ADC and DAC as we have seen from previous examples.

The corresponding transfer functions can be written as

$$\boldsymbol{G}_{\mathrm{ADC}}(\omega) = \boldsymbol{C}_{\mathrm{ADC}}^{\mathsf{T}} \left(i\omega \boldsymbol{I}_N - \boldsymbol{A}_{\mathrm{ADC}}\right)^{-1} \boldsymbol{B}_{\mathrm{ADC}} \tag{11.39}$$

and

$$\begin{aligned} \boldsymbol{G}_{\mathrm{DAC}}(\omega) &= -\boldsymbol{B}_{\mathrm{ADC}}^{\mathsf{T}} \left(i\omega \boldsymbol{I}_N - (-\boldsymbol{A}_{\mathrm{ADC}}^{\mathsf{T}})\right)^{-1} \boldsymbol{C}_{\mathrm{ADC}} && (11.40)\\ &= \left(\boldsymbol{C}_{\mathrm{ADC}}^{\mathsf{T}} \left(i\omega \boldsymbol{I}_N - \boldsymbol{A}_{\mathrm{ADC}}\right)^{-1} \boldsymbol{B}_{\mathrm{ADC}}\right)^{\mathsf{H}} && (11.41)\\ &= \boldsymbol{G}_{\mathrm{ADC}}(\omega)^{\mathsf{H}}. && (11.42) \end{aligned}$$

As seen from (11.42), the two ATF matrices are matched; they are each other's Hermitian transpose. Furthermore, the relation from (11.36) has the additional side effect that for a stable ADC AS, the DAC's AS will be unstable and vice versa. However, as we have seen from previous chapters, the stability of each system is managed by the corresponding DC.

In the following analysis, we focus on the digital-to-digital conversion process as shown in Figure 11.5. In this setting we have replaced the channel by an additive white Gaussian noise term $\boldsymbol{z}(t)$. Furthermore, the waveform encoder and decoder are not explicitly mentioned in the setup. We exclude these two steps since the focus is on the digital-to-analog and analog-to-digital aspects of the communication channel. Naturally,

any additional errors caused by these operations will add to the overall conversion error. Furthermore, we note that the waveform encoder is essentially a part of the DAC process as described in Section 11.2 and the waveform decoder can be included in a post-processing filtering step of the ADC. Therefore, we will assume an already generated fictional $\tilde{\boldsymbol{y}}(t)$ and cover the conversion between $\tilde{\boldsymbol{y}}(t)$ and the digital representation of $\hat{\tilde{\boldsymbol{y}}}(t)$.

Additional A/D conversion performance can be attained by incorporating prior knowledge of the D/A conversion process. Specifically, since the ADC's DE knows of the parametrization of the DAC it includes the DAC's AS matrix in its DE. This means that the ADC forms an estimate of $\tilde{\boldsymbol{y}}(t)$ rather than $\breve{\boldsymbol{y}}(t)$. The NTF of the resulting digital estimation filter can then be written as

$$\boldsymbol{H}(\omega) = \boldsymbol{G}(\omega)\boldsymbol{G}(\omega)^{\mathsf{H}} \left(\boldsymbol{G}(\omega)^{\mathsf{H}}\boldsymbol{G}(\omega)\boldsymbol{G}(\omega)\boldsymbol{G}(\omega)^{\mathsf{H}} + \eta^2 \boldsymbol{I}_N\right)^{-1} \quad (11.43)$$

For the sake of tractable analysis, we next assume a single transmission channel and a single waveform, thus making both $\boldsymbol{B}, \boldsymbol{C} \in \mathbb{R}^N$ column vectors. Naturally, any prior knowledge of the communication channels transfer function can be incorporated, into the DE of both the A/D and D/A conversion steps. The Fourier transform of the estimate can then be written as

$$\begin{aligned} \hat{\tilde{Y}}(\omega) = &\frac{\|G(\omega)\|_2^4}{\|G(\omega)\|_2^4 + \eta^2} \tilde{Y}(\omega) \\ &+ \frac{\|G(\omega)\|_2^3}{\|G(\omega)\|_2^4 + \eta^2} G(\omega) \left(Z(\omega) + \epsilon_1(\omega)\right) \\ &+ \frac{\|G(\omega)\|_2^2}{\|G(\omega)\|_2^4 + \eta^2} \epsilon_2(\omega) \end{aligned} \quad (11.44)$$

where $\hat{\tilde{Y}}(\omega)$, $\tilde{Y}(\omega)$, $Z(\omega)$, $\epsilon_1(\omega)$, and $\epsilon_2(\omega)$ are the continuous-time Fourier transforms of their continuous-time signal counterpart.

Notably, the resulting expressions in (11.44) are similar to those when performing A/D and D/A conversion individually. However, the A/D magnitude of the conversion error $|\epsilon_2(\omega)|$ is approximately $\frac{1}{\|G(\omega)\|_2}$ smaller than for individual conversion. Also the same applies to $\epsilon_1(\omega)$. The difference can be illustrated further by applying the white noise analysis as in Section 4.4 and Section 11.5. Subsequently, the total conversion

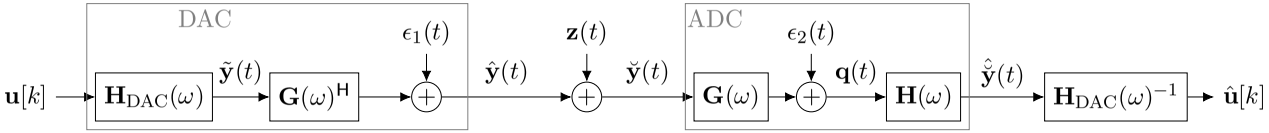


Figure 11.5: Transfer function view of the control-bounded digital-to-digital conversion process.

error can be written as

$$\mathrm{P}_{\epsilon|\mathcal{B}} \approx \frac{1}{2\pi} \int_{\omega \in \mathcal{B}} \left(\frac{\sigma^2_{\epsilon_2|\mathcal{B}}}{\|G(\omega)\|_2^4} + \frac{\sigma^2_{b_{\tilde{y}}|\mathcal{B}} \boldsymbol{C}^{\mathsf{T}}_{\mathrm{DAC}} \boldsymbol{C}_{\mathrm{DAC}}}{\|G(\omega_{\mathrm{crit}})\|_2^2 \|G(\omega)\|_2^2} + \frac{\sigma^2_{z|\mathcal{B}}}{\|G(\omega)\|_2^2} \right) \mathrm{d}\omega \tag{11.45}$$

From (11.45) we recognize that both the A/D and D/A conversion errors (the first two expressions in the parenthesis) are lowered by a factor $\frac{1}{\|G(\omega)\|_2^2}$ compared to the errors when conversion is performed individually. This means that for a matched ADC and DAC, each individual conversion error specification can be relaxed for a fixed target performance.

Naturally, the channel noise term's impact on the conversion error remains unchanged for the joint conversion method compared to individually performing D/A and A/D conversion.

Chapter 12

Conclusions & Outlook

Throughout this thesis, we have seen multiple examples demonstrating the capabilities and features enabled by the control-bounded converter concept. The common theme among these examples was the DE that allowed an alternative interface between analog and digital and thereby new designs of ASs and DCs. The new approach fundamentally changed the converter design principle into designing an AS, using continuous-time analysis and analog circuitry while enforcing stability using a DC. The design task differed substantially between the two components as the AS performance is determined by concepts familiar to the analog circuit designer, such as amplification and frequency filtering. At the same time, the DC reduced to a low complexity control problem without direct concern for signal paths and transfer function analysis.

While the control-bounded ADC is conceptually different, in terms of how we think of A/D conversion, it shares many commonalities with $\Delta\Sigma$ modulators. In fact, the control-bounded ADC could be viewed as a generalization of the $\Delta\Sigma$ modulator concept that allows more flexible AS and DC interactions and thereby designs.

As the control-bounded conversion concept still is an emerging concept, several aspects still require further research. In particular, as the field of A/D conversion already is a much-refined art, many best practices regarding circuit design must be adapted before a competitive control-bounded ADC could realistically be realized in a modern circuit technology. Therefore, a competitive control-bounded converter would most likely involve

refinements to the ASs and DCs presented in this thesis. This is expected as the examples are intended to demonstrates concepts rather than optimized implementations. Due to the DE's flexibility, such changes to both the AS and the DC are typically accounted for by minor modifications to the DE without affecting its computational complexity or operating principle.

As has been repeatedly stated, the DE is a central part of the control-bounded conversion concept and also the enabler resulting in the extended AS and DC design spaces. However, as with all good things, this comes at a cost. For the control-bounded converters, this is the computational complexity of DE. As was discussed in Section 4.3.4 and Section 4.3.5, the increase in computational complexity is not overwhelming as we remain within a linear complexity class. However, the proposed DE cannot compete with scalar decimation filters in terms of the absolute number of multiplications and additions per scalar estimate. Regardless, we believe that this additional cost is motivated as it essentially moves complexity and, in our view, unnecessary design constraints from the analog part of the ADC into the digital domain. Something that also resonates with how circuit technology has evolved over the last decades. By "moving complexity" from analog to digital, we do not suggest that all things are better done in digital. On the contrary, the core idea of the control-bounded converter is to divide the conversion task according to each domain's strengths. In other words, the control-bounded converter concept promotes an analog frontend, unconstrained by digital constraints and considerations, where analog operations enhance the acquisition process as part of the digitalization. Furthermore, with the flexibility in choosing sampling patterns in the DE, this means that the control-bounded converter concept not only is an A/D and D/A converter in the conventional sense but also serves as a design framework for complex analog signal processing and advanced sampling techniques when converting signals between the analog and digital domain.

Even though the DE is conceptually more complicated than conventional postprocessing of a $\Delta\Sigma$ modulator's bitstream, the AS and DC's design task is not. The AS design task resembles that of analog filter design, which is a classical analog circuit discipline. Also, designing the DC proves straightforward as it is nothing else than a low complexity control problem.

Among the examples given in this thesis, the overcomplete DC, from

Chapter 9, is perhaps the one whose appearance most distinguish itself from state of the art ADCs. In particular, the overcomplete DC challenges the way we think of higher-order quantization and instead reuse these resources to control overlapping subspaces of the AS's state space. Conceptually, it is much harder to think of the relation between these overlapping DCs and the sought input signal. However, for the DE, the complex interactions are irrelevant as the computational complexity and general operation is maintained. We know that the overcomplete DC has much potential as it fundamentally distributes the control task and enables tighter control bounds by many overlapping, and thus cumulative, control tasks. However, this does present us with one of the major missing pieces in this work, namely how to dimension the overcomplete DC. Therefore, the overcomplete DC concept, is a topic that remains open for much further research. It is clear that determining overcomplete DC that are also effective, i.e., guarantee stability for a bounded input, is considerably more involved than for a local DC, as the one in the chain-of-integrators architecture from Chapter 5. Additionally, the significance of stability guarantees and the ability to handle the worst possible adversarial input signals can be questioned as encountering such signals might be extremely unlikely. As an example, for $\Delta\Sigma$-modulators, stability is often determined by testing the converter against a set of test input signals. In a similar setting, one could envision that dimensioning the overcomplete DC could be done by data-driven approaches, and the resulting DC would then be tailored for a specific class of input signals.

In Chapter 10, we saw examples of multi-channel input conversion. This is another topic that is a natural extension of the control-bounded scalar ADC. In essence, for multi-input signal scenarios there are potentially large performance gains as for many applications the corresponding scalar input signals are not well captured by an jointly independent assumption on their underlying distributions. In such scenarios, the AS and DC resources can be better combined to enhance overall conversion performance. Chapter 10 shows an example of this that utilized the concepts from Chapter 8 and Chapter 9. However, this topic deserves more consideration as it potentially is very application dependent. Furthermore, multi-input A/D conversion requires us to rethink how we quantify performance measures as the conversion objective is fundamentally changed. Regardless, in our view, multi-input A/D conversion holds massive potential for many of today's and emerging applications and is, therefore a most interesting direction for future research.

12.1 Summary

In this work, we have demonstrated the general principle and several individual aspects and features of the control-bounded conversion concept. In particular, we have seen that the control-bounded ADC can be seen as a generalization of the continuous-time $\Delta\Sigma$ modulator. Specifically, the digital estimator enables greater freedom in terms of the analog and the digital circuit architecture.

We have shown several modular designs where conversion performance can be attained by combining multiple smaller systems. One such example was the chain-of-oscillators ADC, which demonstrates how to make A/D conversion at higher frequency bands. Another example was the Hadamard converter, which transformed the physical signal representation and demonstrated how we could robustify the circuit implementation. Furthermore, the overcomplete DC shows how we can increase the complexity of the DC in a robust way, without component mismatch becoming a conversion performance bottleneck.

All these examples represent fundamental building blocks that can be combined into a very versatile A/D or D/A converter structure. We have also shown the possible benefits of considering multi-input conversion as circuit resources can be shared among multiple conversion processes. Furthermore, we argued that such a converter has the benefit of being dimensioned towards an average rather than a worst-case input signal.

12.2 Outlook

During the process of developing the content of this thesis, many exciting extensions have presented themselves. Several of them did not materialize into this thesis, mainly due to time constraints. We will next discuss some promising and interesting extensions with the intent to inspire future work.

12.2.1 Calibrated Digital Estimator

Both hardware and software calibration are standard tools for refining precision circuitry. However, for control-bounded ADCs, there are some aspects that make this particularly interesting. Firstly, the purpose of calibrating the partially unknown circuit components is to suppress the

effects of mismatch, which is often the limiting factor for a high-resolution ADC.

For control-bounded ADC, we propose a software calibration of the DE. We do not consider hardware calibration since component variations do not have a substantial impact on the AS ability to amplify the input signal, nor the DC ability to remain effective. In other words, we can have a significant mismatch in both the AS and DC without substantially effecting the potential conversion performance.

Another aspect that seems promising is that the control-bounded ADC is by itself a data acquisition device. Therefore, for a given test input signal, the ADC can measure itself without the need for additional circuitry.

Finally, for an overcomplete DC, not every control dimension is strictly needed to maintain an effective control. Alternatively, we could iteratively fix some elements of the control signal, thus creating a test input signal, while controlling the AS state vector using the remaining elements of the control signal. In other words, we can implicitly create test signals by small changes to the DC's operating principle.

In summary, the control-bounded converters are, with minor adaptation, capable of calibrating themselves. Furthermore, the calibration task is essentially a textbook example of a system identification problem where standard approaches, as in [16], would apply.

12.2.2 Clock-Jitter Estimation

The DC of the control-bounded ADC works in synchronization with a global clock. In a practical circuit, the global clock will suffer from timing jitter, meaning variation in the length of each clock cycle. As in the case of component mismatch, the mentioned clock jitter does not have a significant impact on the potential conversion performance. Specifically, a well-designed DC can sustain a substantial amounts of clock jitter without jeopardizing the stability or compromise the amplification of the AS. On the other hand, if the DE does not account for such timing jitter, the estimate is undoubtedly affected.

One way of addressing this issue is to estimate the clock jitter in addition to the converted input signal. Again this idea builds on the fact that the control-bounded ADC is a data acquisition device. In other words, extending the input signal and AS to manifest the clock jitter as a measured signal, for example with an additional dimension containing

the clock pulse (or a zero input), gives the control-bounded ADC the possibility of observing and inferring additional timing information of the clock. Furthermore, there are multiple ways that the DE can be extended to also estimate and thereby suppress the effect of clock jitter. Clearly, such an estimator would require additional computational resources, possibly changing the filters overall computational complexity class.

12.2.3 Multi-Band Frequency A/D Conversion

The multi-channel input A/D conversion of Chapter 10 can also be applied to broadband signals. In particular, for an application where a narrowband signal resides in an unknown sub-band of some broadband frequency range, as in Xampling [10], several chain-of-oscillators ADCs, using the Hadamard conversion principle Chapter 8, could share the AS state space and thereby have a significantly higher conversion performance compared to converting each narrowband separately.

By the same principle, we can also imagine building general broadband A/D converters with shared circuit components.

12.2.4 Configurable ADCs

Using configurable control-bounded ADCs, we can imagine multi-channel input estimation scenarios where, by alterations to the AS and or the DC, the effective resolution is reassigned between the multiple input channel on demand.

Additionally, this could also amount to a power-saving ADC that "turns off" certain parts of the AS and corresponding DC on demand. For low-power applications, this is of particular interest as the A/D converter could be using a subset of the converter (AS states and independent DCs) in an idle state and, upon signal detection, adjust effective resolution and power consumption by an adaptive principle.

12.2.5 General Filter Design

When designing $\Delta\Sigma$ modulators, there are excellent optimization tools for developing the loop-filter transfer function utilizing zero and pole placement. The control-bounded AS design steps would benefit from such tools. However, some additional constraints concerning the largest value of each AS state and the necessary adaptations to the accompanying DC

need to be incorporated. Adapting these tools for the control-bounded ADC design procedure would be a significant contribution to the design procedure.

Appendix A

Wiener-Hopf Equations

The statistical estimation problem in (4.30) resulted, via the orthogonality principle, in the following conditions

$$\mathrm{E}\big[((\boldsymbol{h} * \boldsymbol{q})(t) - \boldsymbol{u}(t))\, \boldsymbol{q}(t+\tau)^{\mathsf{T}}\big] = \mathbf{0}_{1\times N} \tag{A.1}$$

for any $\tau \in \mathbb{R}$. This can be rewritten as

$$\mathrm{E}\big[(\boldsymbol{h} * \boldsymbol{q})(t)\boldsymbol{q}(t-\tau)^{\mathsf{T}}\big] = \mathrm{E}\big[\boldsymbol{u}(t)\boldsymbol{q}(t-\tau)^{\mathsf{T}}\big] \tag{A.2}$$

$$\mathrm{E}\left[\int_{-\infty}^{\infty} \boldsymbol{h}(v)\boldsymbol{q}(t-v)\boldsymbol{q}(t-\tau)^{\mathsf{T}}\,\mathrm{d}v\right] = \mathrm{E}\big[\boldsymbol{u}(t)\boldsymbol{q}(t-\tau)^{\mathsf{T}}\big] \tag{A.3}$$

$$\int_{-\infty}^{\infty} \boldsymbol{h}(v)\mathrm{E}\big[\boldsymbol{q}(t-v)\boldsymbol{q}(t-\tau)^{\mathsf{T}}\big]\,\mathrm{d}v = \mathrm{E}\big[\boldsymbol{u}(t)\boldsymbol{q}(t-\tau)^{\mathsf{T}}\big] \tag{A.4}$$

$$\int_{-\infty}^{\infty} \boldsymbol{h}(v)\mathrm{E}\big[\boldsymbol{q}(t)\boldsymbol{q}(t-\tau+v)^{\mathsf{T}}\big]\,\mathrm{d}v = \mathrm{E}\big[\boldsymbol{u}(t)\boldsymbol{q}(t-\tau)^{\mathsf{T}}\big] \tag{A.5}$$

$$\int_{-\infty}^{\infty} \boldsymbol{h}(v)\boldsymbol{R}_{\boldsymbol{q}\boldsymbol{q}^{\mathsf{T}}}(v-\tau)\,\mathrm{d}v = \boldsymbol{R}_{\boldsymbol{u}\boldsymbol{q}^{\mathsf{T}}}(-\tau) \tag{A.6}$$

$$\int_{-\infty}^{\infty} \boldsymbol{h}(v)\boldsymbol{R}_{\boldsymbol{q}\boldsymbol{q}^{\mathsf{T}}}(\tau-v)\,\mathrm{d}v = \boldsymbol{R}_{\boldsymbol{u}\boldsymbol{q}^{\mathsf{T}}}(-\tau) \tag{A.7}$$

$$(\boldsymbol{h} * \boldsymbol{R}_{\boldsymbol{q}\boldsymbol{q}^{\mathsf{T}}})(\tau) = \boldsymbol{R}_{\boldsymbol{u}\boldsymbol{q}^{\mathsf{T}}}(-\tau) \tag{A.8}$$

where in (A.7) we have made use of the fact that the autocovariance function is symmetric for wide sense stationary stochastic processes. Note that (A.8) is commonly known as the Wiener-Hopf equation. Furthermore,

the autocovariance and cross-covariance functions follows as

$$\boldsymbol{R}_{\boldsymbol{q}\boldsymbol{q}^{\mathsf{T}}}(\tau) \triangleq \mathrm{E}\big[\boldsymbol{q}(t)\boldsymbol{q}(t+\tau)^{\mathsf{T}}\big] \tag{A.9}$$

$$\boldsymbol{R}_{\boldsymbol{u}\boldsymbol{q}^{\mathsf{T}}}(\tau) \triangleq \mathrm{E}\big[\boldsymbol{u}(t)\boldsymbol{q}(t+\tau)^{\mathsf{T}}\big]\,. \tag{A.10}$$

By taking the Fourier transform on both sides of (A.8) we obtain

$$\boldsymbol{H}(\omega)\left(\boldsymbol{G}(\omega)\boldsymbol{S}_{\boldsymbol{u}\boldsymbol{u}^{\mathsf{T}}}(\omega)\boldsymbol{G}(\omega)^{\mathsf{H}} + \boldsymbol{S}_{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}}(\omega)\right) = \boldsymbol{S}_{\boldsymbol{u}\boldsymbol{u}^{\mathsf{T}}}(\omega)\boldsymbol{G}(\omega)^{\mathsf{H}} \tag{A.11}$$

where $\boldsymbol{H}(\omega)$ is the element-wise Fourier transform of $\boldsymbol{h}(t)$ and the PSD

$$\boldsymbol{S}_{\boldsymbol{u}\boldsymbol{u}^{\mathsf{T}}}(\omega) \triangleq \int_{-\infty}^{\infty} \mathrm{E}\big[\boldsymbol{u}(t)\boldsymbol{u}(t+\tau)^{\mathsf{T}}\big]\, e^{-i\omega\tau}\,\mathrm{d}\tau \tag{A.12}$$

$$\boldsymbol{S}_{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}}(\omega) \triangleq \int_{-\infty}^{\infty} \mathrm{E}\big[\boldsymbol{y}(t)\boldsymbol{y}(t+\tau)^{\mathsf{T}}\big]\, e^{-i\omega\tau}\,\mathrm{d}\tau. \tag{A.13}$$

The left hand side of (A.11) follows from

$$\int_{-\infty}^{\infty} (\boldsymbol{h} * \boldsymbol{R}_{\boldsymbol{q}\boldsymbol{q}^{\mathsf{T}}})(\tau) e^{-i\omega\tau}\,\mathrm{d}\tau \tag{A.14}$$

$$= \boldsymbol{H}(\omega)\int_{-\infty}^{\infty} \mathrm{E}\big[\boldsymbol{q}(t)\boldsymbol{q}(t+\tau)^{\mathsf{T}}\big]\, e^{-i\omega\tau}\,\mathrm{d}\tau \tag{A.15}$$

$$= \boldsymbol{H}(\omega)\int_{-\infty}^{\infty} \mathrm{E}\Big[\big(\breve{\boldsymbol{y}}(t) - \boldsymbol{x}(t)\big)\big(\breve{\boldsymbol{y}}(t+\tau) - \boldsymbol{x}(t+\tau)\big)^{\mathsf{T}}\Big]\, e^{-i\omega\tau}\,\mathrm{d}\tau \tag{A.16}$$

$$= \boldsymbol{H}(\omega)\int_{-\infty}^{\infty} \left(\mathrm{E}\big[\breve{\boldsymbol{y}}(t)\breve{\boldsymbol{y}}(t+\tau)^{\mathsf{T}}\big] + \mathrm{E}\big[\boldsymbol{x}(t)\boldsymbol{x}(t+\tau)^{\mathsf{T}}\big]\right) e^{-i\omega\tau}\,\mathrm{d}\tau \tag{A.17}$$

$$= \boldsymbol{H}(\omega)\left(\boldsymbol{G}(\omega)\boldsymbol{S}_{\boldsymbol{u}\boldsymbol{u}^{\mathsf{T}}}(\omega)\boldsymbol{G}(\omega)^{\mathsf{H}} + \boldsymbol{S}_{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}}(\omega)\right) \tag{A.18}$$

where (A.16) follows from (4.24), and (A.17) from the independence between $\boldsymbol{u}(t)$ and $\boldsymbol{x}(t)$. Furthermore, we have used

$$\int_{-\infty}^{\infty} \mathrm{E}\big[\breve{\boldsymbol{y}}(t)\breve{\boldsymbol{y}}(t+\tau)^{\mathsf{T}}\big]\, e^{-i\omega\tau}\,\mathrm{d}\tau \tag{A.19}$$

$$= \int_{-\infty}^{\infty} \mathrm{E}\big[\breve{\boldsymbol{y}}(t)\breve{\boldsymbol{y}}(t-\tau)^{\mathsf{T}}\big]\, e^{-i\omega\tau}\,\mathrm{d}\tau \tag{A.20}$$

$$= \int_{-\infty}^{\infty} \mathrm{E}\big[(\boldsymbol{g} * \boldsymbol{u})(t)(\boldsymbol{g} * \boldsymbol{u})(t-\tau)^{\mathsf{T}}\big]\, e^{-i\omega\tau}\,\mathrm{d}\tau \tag{A.21}$$

$$= \iiint \boldsymbol{g}(\tau_2)\mathrm{E}\big[\boldsymbol{u}(t-\tau_2)\boldsymbol{u}(t-\tau-\tau_3)^{\mathsf{T}}\big]\,\boldsymbol{g}(\tau_3)^{\mathsf{T}}e^{-i\omega\tau}\,\mathrm{d}\tau_2\,\mathrm{d}\tau_3\,\mathrm{d}\tau \tag{A.22}$$

$$= \iiint \boldsymbol{g}(\tau_2)\mathrm{E}\big[\boldsymbol{u}(t-\tau_2)\boldsymbol{u}(t-\tau_2+\tau_4)^{\mathsf{T}}\big]\,\boldsymbol{g}(\tau_3)^{\mathsf{T}}e^{-i\omega(\tau_2-\tau_3-\tau_4)}\,\mathrm{d}\tau_2\,\mathrm{d}\tau_3\,\mathrm{d}\tau_4 \tag{A.23}$$

$$= \boldsymbol{G}(\omega)\boldsymbol{S}_{\boldsymbol{u}\boldsymbol{u}^{\mathsf{T}}}(\omega)\boldsymbol{G}(\omega)^{\mathsf{H}} \tag{A.24}$$

where (A.20) follows from the fact that wide sense stationary stochastic process must have a symmetric autocovariance function, (A.21) follows from (4.23), (A.22) from rearranging convolution and expectation, and (A.23) from the variable transformation $\tau_4 \triangleq \tau_2 - \tau_3 - \tau$. Similarly, the right hand side of (A.11) follows from

$$\int_{-\infty}^{\infty} \boldsymbol{R}_{\boldsymbol{u}\boldsymbol{q}^{\mathsf{T}}}(-\tau)e^{-i\omega\tau}\,\mathrm{d}\tau \tag{A.25}$$

$$= \int_{-\infty}^{\infty} \mathrm{E}\big[\boldsymbol{u}(t)\boldsymbol{q}(t-\tau)^{\mathsf{T}}\big]\,e^{-i\omega\tau}\,\mathrm{d}\tau \tag{A.26}$$

$$= \int_{-\infty}^{\infty} \mathrm{E}\Big[\boldsymbol{u}(t)\left(\breve{\boldsymbol{y}}(t-\tau)-\boldsymbol{x}(t-\tau)\right)^{\mathsf{T}}\Big]\,e^{-i\omega\tau}\,\mathrm{d}\tau \tag{A.27}$$

$$= \int_{-\infty}^{\infty} \mathrm{E}\big[\boldsymbol{u}(t)\left(\boldsymbol{g}*\boldsymbol{u}\right)(t-\tau)^{\mathsf{T}}\big]\,e^{-i\omega\tau}\,\mathrm{d}\tau \tag{A.28}$$

$$= \iint \mathrm{E}\big[\boldsymbol{u}(t)\boldsymbol{u}(t-\tau-\tau_2)^{\mathsf{T}}\big]\,\boldsymbol{g}(\tau_2)^{\mathsf{T}}e^{-i\omega\tau}\,\mathrm{d}\tau\,\mathrm{d}\tau_2 \tag{A.29}$$

$$= \iint \mathrm{E}\big[\boldsymbol{u}(t)\boldsymbol{u}(t+\tau_3)^{\mathsf{T}}\big]\,\boldsymbol{g}(\tau_2)^{\mathsf{T}}e^{i\omega(\tau_2+\tau_3)}\,\mathrm{d}\tau_2\,\mathrm{d}\tau_3 \tag{A.30}$$

$$= \boldsymbol{S}_{\boldsymbol{u}\boldsymbol{u}^{\mathsf{T}}}(\omega)\boldsymbol{G}(\omega)^{\mathsf{H}} \tag{A.31}$$

where (A.27) follows from (4.24), (A.28) is due to the independence between $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ and (4.23), (A.29) comes from the definition of convolution, and (A.30) follows from the variable transformation $\tau_3 \triangleq -\tau - \tau_2$.

Appendix B

Continuous-Time & Discrete-Time Fourier Transformations

The results in this section are common knowledge in the signal processing community, see [10]. However, for sake of completeness, we restate them here.

Theorem 2. *For a signal*

$$x(t) = \sum_{k\in\mathbb{Z}} y[k]h(t - kT_s) \tag{B.1}$$

where $y : \mathbb{Z} \to \mathbb{R}$ *and* $h : \mathbb{R} \to \mathbb{R}$*, the continuous-time Fourier transform is:*

$$X(\omega) = Y(e^{i\omega T_s})H(\omega). \tag{B.2}$$

Proof. From the definition of the continuous-time Fourier transform

$$X(i\omega) = \int \sum_{k\in\mathbb{Z}} y[k]h(t - kT_s)e^{-i\omega t}\,\mathrm{d}t \tag{B.3}$$

$$= \sum_{k\in\mathbb{Z}} y[k] \int h(t - kT_s)e^{-i\omega t}\,\mathrm{d}t \tag{B.4}$$

$$= \sum_{k\in\mathbb{Z}} y[k]e^{-i\omega kT_s} \int h(t)e^{-i\omega t}\,\mathrm{d}t \tag{B.5}$$

$$= Y(e^{i\omega T_s})H(\omega) \tag{B.6}$$

□

Similarly,

Theorem 3. *A continuous-time signal $x(t)$ sampled uniformly with a sample period T_s has the spectrum*

$$\tilde{X}(e^{i\omega T_s}) = \frac{1}{T_s}\sum_{k\in\mathbb{Z}} X\left(\omega - 2\pi k/T_s\right), \tag{B.7}$$

where $\tilde{X}$ is the discrete-time Fourier transform of T_s spaced samples of $x(t)$.

Proof. The sampled signal can be expressed as:

$$x[k] \triangleq x(kT_s) \tag{B.8}$$

for all k. We are going to show that (B.7) is a necessary condition for (B.8) to hold. Subsequently, for

$$x[k] = \frac{1}{2\pi}\int_0^{2\pi} \tilde{X}(e^{i\Omega})e^{i\Omega k}\,\mathrm{d}\Omega \tag{B.9}$$

$$= \frac{1}{2\pi}\int_0^{2\pi} \frac{1}{T_s}\sum_{n\in\mathbb{Z}} X\left(\frac{\Omega - 2\pi n}{T_s}\right) e^{i\Omega k}\,\mathrm{d}\Omega \tag{B.10}$$

$$= \frac{1}{2\pi T_s}\sum_{n\in\mathbb{Z}} \int_0^{2\pi} X\left(\frac{\Omega - 2\pi n}{T_s}\right) e^{i\Omega k}\,\mathrm{d}\Omega \tag{B.11}$$

$$= \frac{1}{2\pi}\sum_{n\in\mathbb{Z}} e^{-2\pi nk} \int_{2\pi k/T_s}^{2\pi(k+1)/T_s} X\left(\tilde{\omega}\right) e^{i\tilde{\omega}kT_s}\,\mathrm{d}\tilde{\omega} \tag{B.12}$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} X\left(\tilde{\omega}\right) e^{i\tilde{\omega}kT_s}\,\mathrm{d}\tilde{\omega} \tag{B.13}$$

$$= x(kT_s) \tag{B.14}$$

where (B.10) follows from plugging in the definition in Theorem 3, (B.11) is a result from Fubini's theorem where we assume that $X(\omega)$ is square integrable, the variable transformation $\tilde{\omega} \triangleq \frac{\Omega - 2\pi n}{T_s}$ leads to (B.12), and finally (B.13) and (B.14) is the fundamental theorem of calculus and the definition of the continuous Fourier transform. □

Using Theorem 2 and 3 the discrete-time Fourier transform of the output of the $\Delta\Sigma$ modulator in Figure 3.4 can be written as

$$\begin{aligned} S(e^{i\omega T_s}) = & Z\left(e^{i\omega T_s}\right) \\ & + \frac{1}{T_s}\sum_{k\in\mathbb{Z}} G\left(\omega - 2\pi k/T_s\right) U\left(\omega - 2\pi k/T_s\right) \\ & + \frac{1}{T_s}\sum_{k\in\mathbb{Z}} G\left(\omega - 2\pi k/T_s\right) D\left(\omega - 2\pi k/T_s\right) S\left(e^{i\omega T_s}\right) \end{aligned} \tag{B.15}$$

where

$$U(\omega) \triangleq \int u(t)e^{-i\omega t}\,\mathrm{d}t \tag{B.16}$$

$$Z(e^{i\omega T_s}) \triangleq \sum_{k\in\mathbb{Z}} z[k]e^{-i\omega T_s k} \tag{B.17}$$

$$S(e^{i\omega T_s}) \triangleq \sum_{k\in\mathbb{Z}} s[k]e^{-i\omega T_s k}, \tag{B.18}$$

and $G(\omega)$ and $D(\omega)$ are the transfer functions of the filter and DAC respectively.

From (B.15) we can also write the transfer functions as

$$S\left(e^{i\omega T_s}\right) = \frac{Z\left(e^{i\omega T_s}\right) + \frac{1}{T_s}\sum_{k\in\mathbb{Z}} G\left(\omega - 2\pi k/T_s\right) U\left(\omega - 2\pi k/T_s\right)}{1 + \frac{1}{T_s}\sum_{k\in\mathbb{Z}} G\left(\omega - 2\pi k/T_s\right) D\left(\omega - 2\pi k/T_s\right)} \tag{B.19}$$

Further assuming, both a bandlimited input signal $u(t)$ and system $G(\omega)$, the expression simplifies as:

$$S\left(e^{i\omega T_s}\right) = \frac{Z\left(e^{i\omega T_s}\right) + L_0(e^{i\omega T_s})\tilde{U}\left(e^{i\omega T_s}\right)}{1 + L_1(e^{i\omega T_s})} \tag{B.20}$$

where

$$\tilde{U}(e^{i\omega T_s}) = \frac{1}{T_s}\sum_{k\in\mathbb{Z}} U(\omega - 2\pi k/T_s) \tag{B.21}$$

$$L_0(e^{i\omega T_s}) \triangleq \sum_{k\in\mathbb{Z}} G(\omega - 2\pi k/T_s) \tag{B.22}$$

$$L_1(e^{i\omega T_s}) \triangleq \frac{1}{T_s}\sum_{k\in\mathbb{Z}} G\left(\omega - 2\pi k/T_s\right) D\left(\omega - 2\pi k/T_s\right). \tag{B.23}$$

Appendix C

Rotation Matrices

Chapter 7 uses several commonly known properties of rotation matrix. For convenience they are restated here. Firstly, a rotation matrix is defined as

$$\boldsymbol{\Theta}(\phi) \triangleq \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix}. \tag{C.1}$$

Furthermore, we recognize that the matrix exponential of a feedback structure as each node in the chain-of-oscillators Figure 7.1 results in a rotation matrix as

$$\begin{aligned} \exp\left(\begin{pmatrix} 0 & -\phi \\ \phi & 0 \end{pmatrix}\right) &= \boldsymbol{I}_2 + \begin{pmatrix} 0 & -\phi \\ \phi & 0 \end{pmatrix} - \frac{1}{2!}\begin{pmatrix} -\phi^2 & 0 \\ 0 & -\phi^2 \end{pmatrix} \\ &\quad + \frac{1}{3!}\begin{pmatrix} 0 & \phi^3 \\ -\phi^3 & 0 \end{pmatrix} + \frac{1}{4!}\begin{pmatrix} \phi^4 & 0 \\ 0 & \phi^4 \end{pmatrix} \\ &\quad - \frac{1}{5!}\begin{pmatrix} 0 & -\phi^5 \\ \phi^5 & 0 \end{pmatrix} + \dots \qquad \text{(C.2)} \\ &= \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} \qquad \text{(C.3)} \\ & \qquad \text{(C.4)} \end{aligned}$$

where we have identified the Taylor series

$$\cos(\phi) = 1 + \frac{1}{2!}\phi^2 + \frac{1}{4!}\phi^4 + \dots \tag{C.5}$$

$$\sin(\phi) = \phi - \frac{1}{3!}\phi^3 + \frac{1}{5!}\phi^5 - \dots \tag{C.6}$$

Secondly, when considering products of rotation matrices we recognize that

$$\boldsymbol{\Theta}(\phi_1)\boldsymbol{\Theta}(\phi_2) = \begin{pmatrix} \cos(\phi_1) & -\sin(\phi_1) \\ \sin(\phi_1) & \cos(\phi_1) \end{pmatrix} \begin{pmatrix} \cos(\phi_2) & -\sin(\phi_2) \\ \sin(\phi_2) & \cos(\phi_2) \end{pmatrix} \tag{C.7}$$

$$= \begin{pmatrix} \cos(\phi_1+\phi_2) & -\sin(\phi_1+\phi_2) \\ \sin(\phi_1+\phi_2) & \cos(\phi_1+\phi_2) \end{pmatrix} \tag{C.8}$$

$$= \boldsymbol{\Theta}(\phi_1+\phi_2) \tag{C.9}$$

where we have used the trigonometric identities

$$\cos(\phi_1+\phi_2) = \cos(\phi_1)\cos(\phi_2) - \sin(\phi_1)\sin(\phi_2) \tag{C.10}$$

$$\sin(\phi_1+\phi_2) = \cos(\phi_1)\sin(\phi_2) + \sin(\phi_1)\cos(\phi_2). \tag{C.11}$$

(C.8) also shows that rotation matrices commute, i.e. $\boldsymbol{\Theta}(\phi_1)\boldsymbol{\Theta}(\phi_2) = \boldsymbol{\Theta}(\phi_2)\boldsymbol{\Theta}(\phi_1)$. Another immediate result is that a rotation matrix's inverse matrix is itself with a negative rotation, i.e.

$$\boldsymbol{\Theta}(\phi)\boldsymbol{\Theta}(-\phi) = \boldsymbol{\Theta}(-\phi)\boldsymbol{\Theta}(\phi) = \boldsymbol{I}_2. \tag{C.12}$$

The rotation matrix is also an orthogonal matrix since $\boldsymbol{\Theta}(\phi)^\mathsf{T} = \boldsymbol{\Theta}(-\phi)$.

Yet another interesting property of rotation matrices is that its derivative with respect to its argument

$$\frac{d}{\mathrm{d}\phi}\boldsymbol{\Theta}(\phi) = \begin{pmatrix} -\sin(\phi) & -\cos(\phi) \\ \cos(\phi) & -\sin(\phi) \end{pmatrix} \tag{C.13}$$

$$= \begin{pmatrix} \cos(\phi+\frac{\pi}{2}) & -\sin(\phi+\frac{\pi}{2}) \\ \sin(\phi+\frac{\pi}{2}) & \cos(\phi+\frac{\pi}{2}) \end{pmatrix} \tag{C.14}$$

$$= \boldsymbol{\Theta}\left(\phi+\frac{\pi}{2}\right) \tag{C.15}$$

$$= \boldsymbol{\Theta}\left(\frac{\pi}{2}\right)\boldsymbol{\Theta}\left(\phi\right) \tag{C.16}$$

can be written as a $\pi/2$ rotation of itself.

Furthermore, the inverse matrix from (7.31), can be written as the eigendecomposition matrix

$$\begin{pmatrix} i\omega & -\omega_\ell \\ \omega_\ell & i\omega \end{pmatrix} = \underbrace{\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}}_{\boldsymbol{Q}} \underbrace{\begin{pmatrix} i(\omega-\omega_\ell) & 0 \\ 0 & i(\omega+\omega_\ell) \end{pmatrix}}_{\boldsymbol{\Lambda}} \underbrace{\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}\frac{1}{\sqrt{2}}}_{\boldsymbol{Q}^{-1}} \tag{C.17}$$

Additionally, the rotation matrix and the matrix (C.17) commute, i.e.

$$\begin{aligned}\boldsymbol{\Theta}(\phi)\begin{pmatrix} a & -b \\ b & a \end{pmatrix} &= \begin{pmatrix} a\cos(\phi) - b\sin(\phi) & -b\cos(\phi) - a\sin(\phi) \\ b\cos(\phi) + a\sin(\phi) & a\cos(\phi) - b\sin(\phi) \end{pmatrix} \\ &= \begin{pmatrix} a & -b \\ b & a \end{pmatrix}\boldsymbol{\Theta}(\phi) \end{aligned} \tag{C.18}$$

Appendix D

Factor Graphs and Gaussian Message Passing

D.1 A/D Digital Estimation Filter

In this appendix, we give a brief and condensed derivation of the algorithm of Section 4.3.2. This derivation was stated in [20] and is repeated here for convenience. Furthermore,

We first observe that the filter (4.41) is formally a multivariate extension of the continuous-time Wiener filter [1] that estimates a multivariate zero-mean white Gaussian noise "signal" $\boldsymbol{U}(t)$ from the signal

$$\tilde{\boldsymbol{Y}}(t) \triangleq (\boldsymbol{g} * \boldsymbol{U})(t) + \boldsymbol{Z}(t), \tag{D.1}$$

where $\boldsymbol{Z}(t)$ is m-dimensional zero-mean white Gaussian noise that is independent of $\boldsymbol{U}(t)$. In this statistical model, the average

$$\tilde{\boldsymbol{U}}(t, \Delta) \triangleq \frac{1}{\Delta} \int_{t-\Delta}^{t} \boldsymbol{U}(\tau)\, d\tau \tag{D.2}$$

(for $\Delta > 0$) is a L-dimensionalzero-mean Gaussian random variable with covariance matrix $\frac{\sigma_U^2}{\Delta} \boldsymbol{I}_K$. The covariance matrix $\frac{\sigma_Z^2}{\Delta} \boldsymbol{I}_m$ of $\boldsymbol{Z}(t)$ is defined analogously.

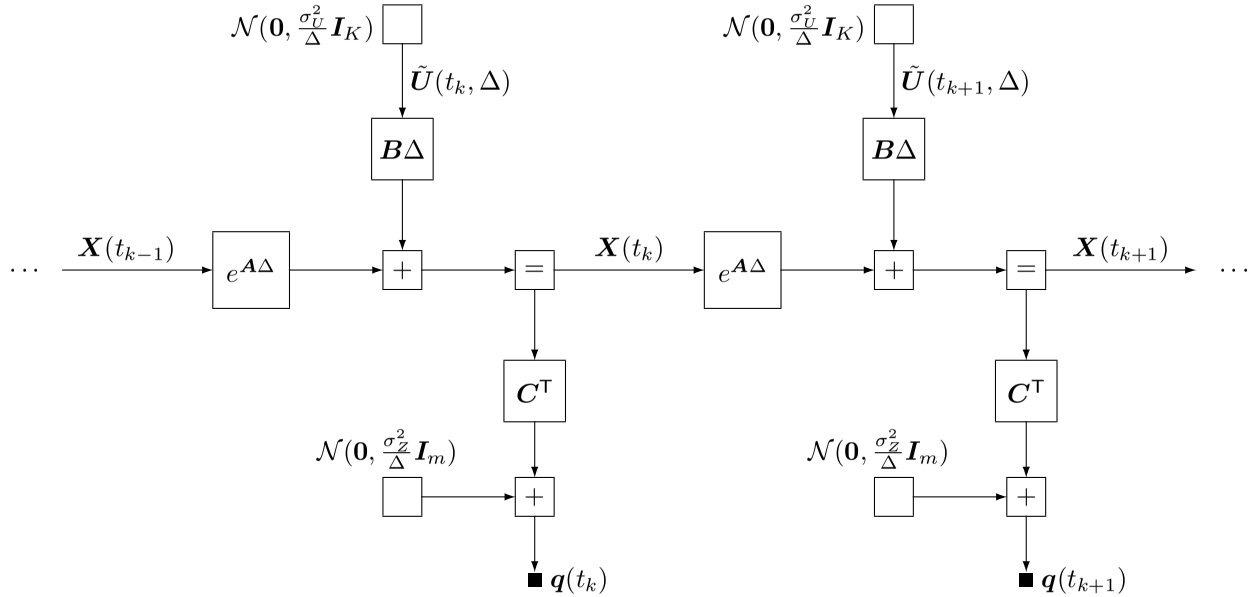


Figure D.1: Two sections of the factor graph of the (uncontrolled) state space model. The total factor graph consists of many such sections; perhaps with initial and final conditions, which we can ignore in this paper. A box labeled "$\mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$" represents a multivariate Gaussian density with mean vector $\boldsymbol{m}$ and covariance matrix $\boldsymbol{\Sigma}$, $\mathbf{0}$ refers to an all zero vector of appropriate dimensions, and a small filled box represents a known quantity; all other boxes represent linear equations. This factor graph representation is exact only in the limit $\Delta = t_k - t_{k-1} \to 0$.

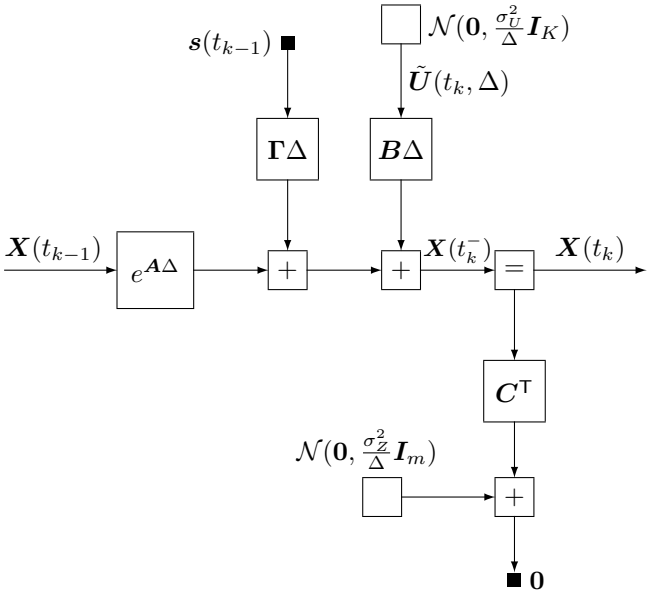


Figure D.2: One section of the factor graph of the state space model with plugged-in digital control signals $\boldsymbol{s}(t)$. The total factor graph consists of many such sections. The representation is exact only in the limit $\Delta = t_k - t_{k-1} \to 0$, where $e^{\boldsymbol{A}\Delta} \to \boldsymbol{I}_n + \boldsymbol{A}\Delta$.

By "estimating $\boldsymbol{U}(t)$", we really mean to estimate the random variable(s) (D.2) for any fixed t, and then taking the limit $\Delta \to 0$ [5]. In this setting, the MAP estimate, the MMSE estimate, and the LMMSE estimate agree and equal the mean of the posterior distribution of $\tilde{\boldsymbol{U}}(t, \Delta)$ conditioned on the observation of $\tilde{\boldsymbol{Y}}(t)$. The Wiener filter computes this estimate (for $\Delta \to 0$) as

$$\hat{\boldsymbol{U}}(t) = (\boldsymbol{h} * \tilde{\boldsymbol{Y}})(t) \tag{D.3}$$

where the Fourier transform of $\boldsymbol{h}(t)$ is (4.41) with

$$\eta^2 = \sigma_Z^2 / \sigma_U^2. \tag{D.4}$$

Applying this Wiener filter to the signal $\boldsymbol{q}(t)$ as in (4.25) means that we solve the statistical estimation problem for the observation $\tilde{\boldsymbol{Y}}(t) = \boldsymbol{q}(t)$.

The same statistical estimation problem can also be solved by a variation of Kalman smoothing. In contrast to the Wiener filter, the Kalman approach is based on the state space equations (4.4) and (4.6), which leads to recursive estimation algorithms. We will use a discrete-time approximation of the state space model with discrete times[1] $t_1, t_2, \ldots$ and fixed $t_k - t_{k-1} = \Delta > 0$; our continuous-time results will then be obtained by taking the limit $\Delta \to 0$.

From now on, we will use factor graphs as in [18], which allow to compose recursive estimation algorithms from lookup tables of "local" computations. A factor graph of (the discrete-time approximation of) our statistical model in state space form is shown in Figure D.1. Note that Figure D.1 represents the uncontrolled analog system with the observations $\tilde{\boldsymbol{Y}}(t_k) = \boldsymbol{q}(t_k)$.

Now we plug in the (known and piecewise constant) control signals $\boldsymbol{s}(t) = (s_1(t), \ldots, s_n(t))$ into the state space model. We thus obtain the factor graph of Figure D.2, where all the observed signals are now zero. This second factor graph is easy to work with and then to take the limit $\Delta \to 0$ to continuous time.

Using the notation of [18], we now consider the quantities $\overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t)}$ and $\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t)}$ as well as $\overleftarrow{\boldsymbol{m}}_{\boldsymbol{X}(t)}$ and $\overleftarrow{\boldsymbol{V}}_{\boldsymbol{X}(t)}$. The former denote the mean vector and the covariance matrix, respectively, of the forward sum-product

[1] The discrete times $t_1, t_2, \ldots$ in this appendix (with $t_k - t_{k-1} = \Delta \to 0$) are unrelated to the discrete time steps in Section 4.3.2.

message, which equals the Gaussian probability density of the time-t state $\boldsymbol{X}(t)$ given past observations (up to a scale factor); the latter denote the mean vector and the covariance matrix, respectively, of the backward sum-product message, which equals the likelihood of the (given) future observations conditioned on $\boldsymbol{X}(t)$ (up to a scale factor).

From Figure D.2, we determine these quantities using Tables II–IV of [18] as follows. From (III.1) and (II.7) of [18], we have

$$\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_k^-)} = e^{\boldsymbol{A}\Delta}\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_{k-1})}(e^{\boldsymbol{A}\Delta})^{\mathsf{T}} + \sigma_U^2\Delta\boldsymbol{B}\boldsymbol{B}^{\mathsf{T}}, \tag{D.5}$$

and from (IV.2) and (IV.3) of [18], we have

$$\begin{aligned}\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_k)} = \overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_k^-)} \\ -\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_k^-)}\boldsymbol{C}\left(\frac{\sigma_Z^2}{\Delta}\boldsymbol{I}_o + \boldsymbol{C}^{\mathsf{T}}\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_k^-)}\boldsymbol{C}\right)^{-1}\boldsymbol{C}^{\mathsf{T}}\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_k^-)}\end{aligned} \tag{D.6}$$

For $\Delta \approx 0$, we have

$$e^{\boldsymbol{A}\Delta} \approx \boldsymbol{I}_n + \Delta\boldsymbol{A}; \tag{D.7}$$

thus (D.5) becomes

$$\begin{aligned}\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_k^-)} \approx \overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_{k-1})} \\ + \Delta\left(\boldsymbol{A}\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_{k-1})} + (\boldsymbol{A}\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_{k-1})})^{\mathsf{T}} + \sigma_U^2\boldsymbol{B}\boldsymbol{B}^{\mathsf{T}}\right)\end{aligned} \tag{D.8}$$

and (D.6) becomes

$$\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_k)} \approx \overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_k^-)} - \frac{\Delta}{\sigma_Z^2}\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_k^-)}\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}}\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t_k^-)}. \tag{D.9}$$

Combining (D.8) and (D.9) yields Equations (4.56)-(4.58) as the steady-state condition for

$$\overrightarrow{\boldsymbol{V}} \triangleq \overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t)}/\sigma_U^2 \tag{D.10}$$

in the limit $\Delta \to 0$.

The derivation of (4.59) is essentially identical except that the matrix $e^{\boldsymbol{A}\Delta}$ is replaced by its inverse, which amounts to a sign change in $\boldsymbol{A}$.

As for $\overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t)}$, we have

$$\overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t_k^-)} = e^{\boldsymbol{A}\Delta}\overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t_k)} + \boldsymbol{\Gamma}\boldsymbol{s}(t_{k-1})\Delta \tag{D.11}$$

from (III.2) and (II.9) of [18], and

$$\begin{aligned}\overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t_k)} = \overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t_k^-)} \\ -\overrightarrow{\boldsymbol{V}}_{X(t_k^-)}\boldsymbol{C}\left(\frac{\sigma_Z^2}{\Delta}\boldsymbol{I}_o + \boldsymbol{C}^{\mathsf{T}}\overrightarrow{\boldsymbol{V}}_{X(t_k^-)}\boldsymbol{C}\right)^{-1}\boldsymbol{C}^{\mathsf{T}}\overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t_k^-)}\end{aligned} \tag{D.12}$$

from (IV.1) and (IV.3) of [18]. For $\Delta \approx 0$, we obtain with (D.7)

$$\begin{aligned}\overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t_k)} = \overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t_{k-1})} + \Delta\Big(\boldsymbol{A}\overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t_{k-1})} \\ + \boldsymbol{\Gamma}\boldsymbol{s}(t_{k-1}) - \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}}\overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t_{k-1})}\Big),\end{aligned} \tag{D.13}$$

where we have used the normalized stationary covariance matrix (D.10). Note that (D.13) is exact in the limit $\Delta \to 0$ and amounts to the differential equation

$$\frac{d}{dt}\overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t)} = \left(\boldsymbol{A} - \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}}\right)\overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t)} + \boldsymbol{\Gamma}\boldsymbol{s}(t). \tag{D.14}$$

The solution of this differential equation (for $t > 0$) is

$$\overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(t)} = e^{\tilde{\boldsymbol{A}}t}\overrightarrow{\boldsymbol{m}}_{\boldsymbol{X}(0)} + e^{\tilde{\boldsymbol{A}}t}\int_0^t e^{-\tilde{\boldsymbol{A}}\tau}\boldsymbol{\Gamma}\boldsymbol{s}(\tau)\,d\tau \tag{D.15}$$

with $\tilde{\boldsymbol{A}} \triangleq \boldsymbol{A} - \overrightarrow{\boldsymbol{V}}\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}}/\eta^2$. This solution applies to any interval between t_k and t_{k+1} in Section 4.3.2 and yields (4.53) with (4.61) and (4.63).

The derivation for $\overleftarrow{\boldsymbol{m}}_{\boldsymbol{X}(t)}$ is essentially identical except for a sign change in both $\boldsymbol{A}$ and $\boldsymbol{\Gamma}$, where the latter is due to (II.10) of [18].

Finally, we use the result from [4] that the MAP/MMSE/LMMSE estimate of $U(t)$ (i.e., the posterior mean of (D.2) for $\Delta \to 0$) is given by

$$\hat{\boldsymbol{u}}(t) = \sigma_U^2\boldsymbol{B}^{\mathsf{T}}\tilde{\boldsymbol{W}}(t)\left(\overleftarrow{\boldsymbol{m}}_{X(t)} - \overrightarrow{\boldsymbol{m}}_{X(t)}\right) \tag{D.16}$$

with

$$\tilde{\boldsymbol{W}}(t) \triangleq \left(\overrightarrow{\boldsymbol{V}}_{\boldsymbol{X}(t)} + \overleftarrow{\boldsymbol{V}}_{\boldsymbol{X}(t)}\right)^{-1}, \tag{D.17}$$

which yields (4.55) and (4.60). Note that (D.16) and (D.17) may also be obtained directly from Figure D.2 using (II.12), (III.8), and (III.9) of [18] and then taking the limit $\Delta \to 0$.

D.2 D/A Digital Estimation Filter

The D/A digital estimation task is a variation from the A/D digital estimation task shown in Appendix D.1. Specifically, the observations are not continuous-time observations. Instead these correspond to a sequence of M-dimensional samples. In the following we will assume these to be uniformly spaced with a time period T_s. In contrast, to the discrete-time observations the input signal is a continuous-time estimate as defined in (D.2). The estimation setup is illustrated in Figure D.3. In particular, if we let $\Delta \to 0$ the input contribution between two samples can be expressed as an additive zero mean multi-variate Gaussian random variable with a covariance matrix

$$\boldsymbol{\Sigma}_{\boldsymbol{u},T_s} = \int_0^{T_s} e^{\boldsymbol{A}\tau} \boldsymbol{B} \boldsymbol{\Sigma}_{\boldsymbol{u}} \boldsymbol{B}^{\mathsf{T}} e^{\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\tau}} \, \mathrm{d}\tau \in \mathbb{R}^{N \times N} \tag{D.18}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{u}}$ is the instantaneous input covariance matrix.

The resulting factor graph is a discrete-time factor graph as illustrated in Figure D.4. The posterior mean of the state vector $\boldsymbol{X}(kT_s) \triangleq \boldsymbol{X}[k]$ can then be computed using standard Gaussian messaging as proposed in [18]. In the following we use a particular version called the backward information filter, forward marginal (BIFM) as in [35]. Furthermore, we assuming that the recursions reach a steady state, i.e.,

$$\boldsymbol{W} = e^{\boldsymbol{A}^{\mathsf{T}}T_s} \boldsymbol{W} e^{\boldsymbol{A}T_s} - e^{\boldsymbol{A}^{\mathsf{T}}T_s} \boldsymbol{W} \left(\boldsymbol{\Sigma}_{\boldsymbol{u},T_s}^{-1} + \boldsymbol{W} \right)^{-1} \boldsymbol{W} e^{\boldsymbol{A}T_s} + \boldsymbol{C} \boldsymbol{\Sigma}_{\boldsymbol{z}}^{-1} \boldsymbol{C}^{\mathsf{T}}. \tag{D.19}$$

Were we recognize the expression from (D.19) as an algebraic Riccati equation. Acquiring a steady state $\boldsymbol{W} \in \mathbb{R}^{N \times N}$ that satisfy (D.19) is a standard problem in control theory.

The recursion can be written as

$$\boldsymbol{\xi}[k] = \boldsymbol{A}_b \boldsymbol{\xi}[k+1] + \boldsymbol{B}_b \boldsymbol{y}[k] \tag{D.20}$$

$$\boldsymbol{x}[k] = \boldsymbol{A}_f \boldsymbol{x}[k-1] + \boldsymbol{B}_f \boldsymbol{\xi}[k] \tag{D.21}$$

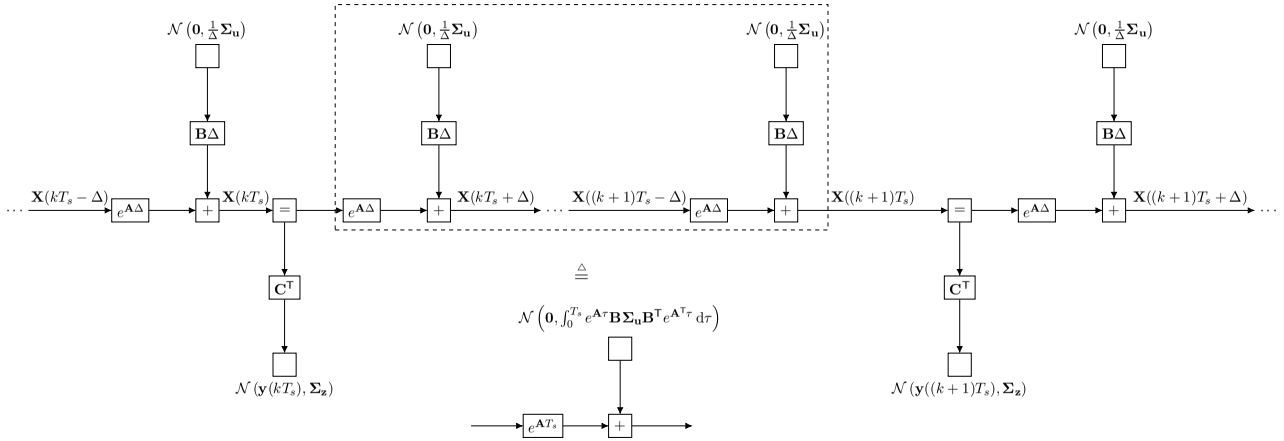


Figure D.3: Factor graph describing a continuous-time input process in between discrete-time observations.

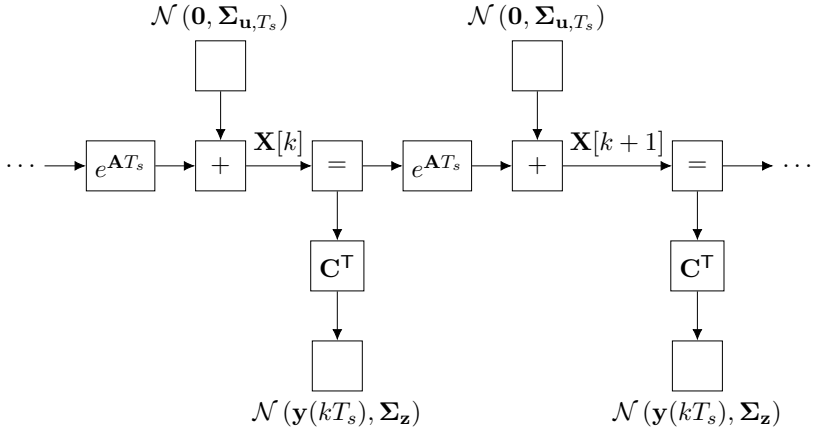


Figure D.4: The discrete-time factor graph of a state space model with random variables as inputs and outputs.

where

$$\boldsymbol{A}_b = e^{\boldsymbol{A}^\mathsf{T} T_s} \left(\boldsymbol{I}_N - \boldsymbol{W} \left(\boldsymbol{\Sigma}_{\boldsymbol{u},T_s}^{-1} + \boldsymbol{W} \right)^{-1} \right) \in \mathbb{R}^{N \times N} \tag{D.22}$$

$$\boldsymbol{A}_f = \boldsymbol{A}_b^\mathsf{T} \in \mathbb{R}^{N \times N} \tag{D.23}$$

$$\boldsymbol{B}_b = \boldsymbol{C}^\mathsf{T} \boldsymbol{\Sigma}_{\boldsymbol{z}}^{-1} \in \mathbb{R}^{N \times M} \tag{D.24}$$

$$\boldsymbol{B}_f = \left(\boldsymbol{\Sigma}_{\boldsymbol{u},T_s}^{-1} + \boldsymbol{W} \right)^{-1} \in \mathbb{R}^{N \times N}. \tag{D.25}$$

Similarly to the A/D estimation algorithm from Appendix D.1 the Equations (D.22)-(D.25) can be computed offline. The $\boldsymbol{x}[k]$ in (D.21) is the posterior mean vector of the state evaluated at kT_s. This is also the output of the DAC's digital estimation filter.

In the expressions above the posterior means are computed at the sample times dictated by T_s. We note that any sample in between the given sample can be computed as proposed in [5].

Finally, a sensible initialization is to set the initial $\boldsymbol{x}[0] = \boldsymbol{\xi}[\infty] = \mathbf{0}_N$.

Appendix E

Digital Estimation Filter Implementation

Several alternative DE implementations were suggested in Section 4.3.4 and Section 4.3.5. In this appendix we will describe them in more detail.

E.1 Offline Estimation

In an offline setting the objective is to estimate a batch of input signal samples $\{\hat{\boldsymbol{u}}(K_0T), \ldots, \hat{\boldsymbol{u}}((K_0+K_1-1)T)\}$ where we are given a batch of control signals $\{\boldsymbol{s}[K_0], \ldots, \boldsymbol{s}[K_0+K_1-1]\}$ as well as the precomputed filter coefficients.

E.1.1 Digital Estimation Filter

The plain vanilla version uses the recursions

$$\overrightarrow{\boldsymbol{m}}_{k+1} \triangleq \boldsymbol{A}_f \overrightarrow{\boldsymbol{m}}_k + \boldsymbol{B}_f \boldsymbol{s}[k], \tag{E.1}$$

$$\overleftarrow{\boldsymbol{m}}_{k-1} \triangleq \boldsymbol{A}_b \overleftarrow{\boldsymbol{m}}_k + \boldsymbol{B}_b \boldsymbol{s}[k-1], \tag{E.2}$$

and

$$\hat{\boldsymbol{u}}(t_k) \triangleq \boldsymbol{W}^{\mathsf{T}} \left(\overleftarrow{\boldsymbol{m}}_k - \overrightarrow{\boldsymbol{m}}_k \right) \tag{E.3}$$

which are described in Section 4.3.2 and repeated here for convenience. Note that the filter coefficients, corresponding to the matrices $\boldsymbol{A}_f$, $\boldsymbol{B}_f$, $\boldsymbol{A}_b$, $\boldsymbol{B}_b$, and $\boldsymbol{W}$, are all precomputed. A pseudo code version of these computations is given in Algorithm 2.

Function `BatchEstimator(`$\boldsymbol{s}$, K_1`):`
  input : $\boldsymbol{s}$ - a batch of control signals of dimensions $M \times K_1$.
    K_1 - the number of samples in the batch.
  output : A batch of estimates $\hat{\boldsymbol{u}}$ of dimensions $L \times K_1$.
  `// initalize data vectors`
  $\boldsymbol{M} \leftarrow \mathbf{0}_{N\times(K_1+1)}$
  $\tilde{\boldsymbol{M}} \leftarrow \mathbf{0}_N$
  $\hat{\boldsymbol{u}} \leftarrow \mathbf{0}_{L\times K_1}$
  `// compute the forward recursion`
  for $k_1 \leftarrow 1$ **to** $K_1 - 1$ **do**
    $\boldsymbol{M}[k_1] \leftarrow \boldsymbol{A}_f\boldsymbol{M}[k_1-1] + \boldsymbol{B}_f\boldsymbol{s}[k_1-1]$
  end
  `// compute the backward recursion and estimate`
  for $k_2 \leftarrow K_1$ **to** 1 **do**
    $\tilde{\boldsymbol{M}} \leftarrow \boldsymbol{A}_b\boldsymbol{M}[k_2] + \boldsymbol{B}_b\boldsymbol{s}[k_2-1]$
    $\hat{\boldsymbol{u}}[k_2-1] \leftarrow \boldsymbol{W}^{\mathsf{T}}\left(\tilde{\boldsymbol{M}} - \boldsymbol{M}[k_2-1]\right)$
    $\boldsymbol{M}[k_2-1] \leftarrow \tilde{\boldsymbol{M}}$
  end
  return $\hat{\boldsymbol{u}}$
end

Algorithm 2: Estimating a batch of samples using the filter recursions from Section 4.3.2.

Furthermore, we remind ourselves that the matrix

$$\mathbf{0}_{N\times K_1} \triangleq \begin{pmatrix}\mathbf{0}_N & \dots & \mathbf{0}_N\end{pmatrix} \in \mathbb{R}^{N\times K_1} \tag{E.4}$$

and

$$\mathbf{0}_N \triangleq \begin{pmatrix}0 & \dots & 0\end{pmatrix}^{\mathsf{T}} \in \mathbb{R}^N. \tag{E.5}$$

Also, $\boldsymbol{M}[k]$ refers to the k-th column vector of the matrix $\boldsymbol{M}$ where our indexing starts from 0. These conventions will be used throughout this appendix.

E.1.2 Parallel Digital Estimation Filter

An alternative offline estimator uses the parallelized filter recursions

$$\overrightarrow{\tilde{m}}_{k+1,n} \triangleq \overrightarrow{\tilde{\lambda}}_n \overrightarrow{\tilde{m}}_{k,n} + \overrightarrow{\tilde{f}}_n(\boldsymbol{s}[k]) \tag{E.6}$$

$$\overleftarrow{\tilde{m}}_{k-1,n} \triangleq \overleftarrow{\tilde{\lambda}}_n \overleftarrow{\tilde{m}}_{k,n} + \overleftarrow{\tilde{f}}_n(\boldsymbol{s}[k-1]) \tag{E.7}$$

and

$$\hat{u}_\ell[k] = \sum_{n=1}^{N} \overrightarrow{\tilde{w}}_{n,\ell} \overrightarrow{\tilde{m}}_{k,n} + \overleftarrow{\tilde{w}}_{n,\ell} \overleftarrow{\tilde{m}}_{k,n} \tag{E.8}$$

as described in Section 4.3.3. Furthermore, note that the filter coefficients $\overrightarrow{\tilde{\lambda}}_n$, $\overrightarrow{\tilde{f}}_n(\cdot)$, $\overleftarrow{\tilde{\lambda}}_n$, $\overleftarrow{\tilde{f}}_n(\cdot)$, $\overrightarrow{\tilde{w}}_{n,\ell}$, and $\overleftarrow{\tilde{w}}_{n,\ell}$ are precomputed. The algorithm is expressed using pseudo code in Algorithm 3.

Note that for this version all computations are expressed as scalar operations. As the filter coefficients are typically complex, this filter requires complex arithmetics. We also observe that many of the computations in this algorithm allows parallelization as is highlighted by the *do in parallel* statements in the for loops.

E.2 Online Estimator

In a conventional ADC the estimated samples are not computed offline in a subsequent processing step but rather incorporated into the continuous stream of samples. Next we discuss modifications to the previously proposed offline estimators (Section E.1) such that they generate streams of batches. For the online estimators we imagine an endless stream of control signals $\boldsymbol{s}[k]$ denoted the *inStream* that we can access in a sequential order. Similarly, the objective of the estimator is to gradually populate an *outStream* with estimated signal samples $\hat{\boldsymbol{u}}[k]$.

The online version of the offline estimator from Section E.1.1 is given in Algorithm 4.

In addition to the general program we recognize that the reading and writing operations from and to the *inStream* and *outStream* are given in Algorithm 5 and Algorithm 6 respectively.

Additionally, the mod operator from row 31 in Algorithm 4 refers to the modulus operator and represents the use of a circular buffer for the control signal buffer.

Function `ParallelBatchEstimator(`$\boldsymbol{s}$, K_1`):`
  input : $\boldsymbol{s}$ - a batch of control signals of dimensions $M \times K_1$.
  K_1 - the number of samples in the batch.
  output : A batch of estimates $\hat{\boldsymbol{u}}$ of dimensions $L \times K_1$.
  `// initalize data vectors`
  $\overrightarrow{\boldsymbol{M}} \leftarrow \mathbf{0}_{N\times(K_1+1)}$
  $\overleftarrow{\boldsymbol{M}} \leftarrow \mathbf{0}_{N\times(K_1+1)}$
  $\hat{\boldsymbol{u}} \leftarrow \mathbf{0}_{L\times K_1}$
  `// compute the forward and backward recursion`
  for $n \leftarrow 1$ **to** N **do** in parallel
    for $k_2 \leftarrow 1$ **to** K_1 **do**
      $\overrightarrow{\boldsymbol{M}}[n, k_2] \leftarrow \overrightarrow{\lambda}_n \overrightarrow{\boldsymbol{M}}[n, k_2 - 1] + \overrightarrow{f}_n(\boldsymbol{s}[k_2 - 1])$
      $k_3 \leftarrow K_1 - k_2 + 1$
      $\overleftarrow{\boldsymbol{M}}[n, k_3] \leftarrow \overleftarrow{\lambda}_n \overleftarrow{\boldsymbol{M}}[n, k_3 + 1] + \overleftarrow{f}_n(\boldsymbol{s}[k_3])$
    end
  end
  `// compute estimates`
  for $\ell \leftarrow 1$ **to** L **do** in parallel
    for $k_4 \leftarrow 1$ **to** K_1 **do** in parallel
      $\hat{\boldsymbol{u}}[\ell, k_4 - 1] \leftarrow \sum_{n=1}^{N} \overrightarrow{w}_{n,\ell} \overrightarrow{\boldsymbol{M}}[n, k_4] + \overleftarrow{w}_{n,\ell} \overleftarrow{\boldsymbol{M}}[n, k_4]$
    end
  end
  return $\hat{\boldsymbol{u}}$
end

Algorithm 3: Estimating a batch of samples using the parallel recursions from Section 4.3.3.

Function `OnlineEstimator(`*inStream, outStream,* K_1*,* K_2`):`
  input : *inSteam* - providing M dimensional samples $\boldsymbol{s}[k]$.
  outStream - accepting L dimensional estimates $\hat{\boldsymbol{u}}[k]$.
  K_1 - batch size.
  K_2 - lookahead size.
  `// intialize data buffers and auxilary variable`
  $\boldsymbol{M} \leftarrow \boldsymbol{0}_{N\times(K_1+1)}$
  $\tilde{\boldsymbol{M}} \leftarrow \boldsymbol{0}_N$, $\tilde{\boldsymbol{M}}_0 \leftarrow \boldsymbol{0}_N$
  $\hat{\boldsymbol{u}} \leftarrow \boldsymbol{0}_{L\times K_1}$
  $\boldsymbol{s} \leftarrow \boldsymbol{0}_{M\times(K_1+K_2+1)}$
  $K_3 \leftarrow K_1 + K_2$
  `// start online algorithm`
  repeat
    `// retrive` K_1 `new samples as in Algorithm 5`
    $\boldsymbol{s} \leftarrow$ `RetriveSamplesFromStream(`$\boldsymbol{s}$*, inStream,* K_1*,* K_2`)`
    `// compute lookahead mean vector`
    for $k_1 \leftarrow K_3$ **to** $K_1 + 1$ **do**
      $\boldsymbol{M}[K_1+1] \leftarrow \boldsymbol{A}_b\boldsymbol{M}[K_1+1] + \boldsymbol{B}_b\boldsymbol{s}[k_1]$
    end
    `// compute batch of estimates`
    for $k_2 \leftarrow 1$ **to** $K_1 - 1$ **do**
      $\boldsymbol{M}[k_2] \leftarrow \boldsymbol{A}_f\boldsymbol{M}[k_2-1] + \boldsymbol{B}_f\boldsymbol{s}[k_2-1]$
    end
    $\tilde{\boldsymbol{M}}_0 \leftarrow \boldsymbol{M}[K_1-1]$
    for $k_3 \leftarrow K_1$ **to** 1 **do**
      $\tilde{\boldsymbol{M}} \leftarrow \boldsymbol{A}_b\boldsymbol{M}[k_3] + \boldsymbol{B}_b\boldsymbol{s}[k_3-1]$
      $\hat{\boldsymbol{u}}[k_3-1] \leftarrow \boldsymbol{W}^{\mathsf{T}}\left(\tilde{\boldsymbol{M}} - \boldsymbol{M}[k_3-1]\right)$
      $\boldsymbol{M}[k_3-1] \leftarrow \tilde{\boldsymbol{M}}$
    end
    `// set inital mean for next batch`
    $\boldsymbol{M}[0] \leftarrow \tilde{\boldsymbol{M}}_0$
    `// reset lookahead mean vector`
    $\boldsymbol{M}[K_1+1] \leftarrow \boldsymbol{0}_N$
    `// shift control signal buffer`
    $\boldsymbol{s}[k] \leftarrow \boldsymbol{s}[\mathrm{mod}(k+K_1, K_3)]$
    `// write batch to` ***outStream*** `as in Algorithm 6`
    `SubmitEstimatesToStream(`$\hat{\boldsymbol{u}}$*, outStream,* K_1`)`
  until inStream *closes*
end

Algorithm 4: Online filter computations.

```
1 Function RetriveSamplesFromStream(s, stream, K_1, K_2):
      input : s - buffer to write samples to.
              stream - input stream to read samples from.
              K_1 - batch size.
              K_2 - offset.
2     // retrive K_1 new samples
3     for ℓ ← 1 to K_1 do
4         s[K_2 + ℓ] ← stream
5     end
6     return s
7 end
```

Algorithm 5: Auxiliary function for reading from a stream.

```
1 Function SubmitEstimatesToStream(û, stream, K_1):
      input : û - buffer of estimates to be written.
              stream - output stream to write samples to.
              K_1 - batch size.
2     // write new batch of samples to stream
3     for ℓ ← 0 to K_1 − 1 do
4         stream ← û[ℓ]
5     end
6 end
```

Algorithm 6: Auxiliary function for writing to a stream.

Similarly, the online version of the parallelized two-way filtering is given in Algorithm 7.

Function `ParOnlineEstimator(`*inStream, outStream,* K_1*,* K_2`):`
  input : *inSteam* - providing M dimensional samples $\boldsymbol{s}[k]$.
  outStream - accepting L dimensional estimates $\hat{\boldsymbol{u}}[k]$.
  K_1 - batch size.
  K_2 - lookahead size.
  `// intialize data buffers and auxilary variable`
  $\overrightarrow{\boldsymbol{M}} \leftarrow \mathbf{0}_{N\times(K_1+1)}$; $\overleftarrow{\boldsymbol{M}} \leftarrow \mathbf{0}_{N\times(K_1+1)}$; $\hat{\boldsymbol{u}} \leftarrow \mathbf{0}_{L\times K_1}$;
  $\boldsymbol{s} \leftarrow \mathbf{0}_{M\times(K_1+K_2+1)}$; $K_3 \leftarrow K_1 + K_2$
  `// start online algorithm`
  repeat
    `// retrive` K_1 `new samples as in Algorithm 5`
    $\boldsymbol{s} \leftarrow$ `RetriveSamplesFromStream(`$\boldsymbol{s}$*, inStream,* K_1*,* K_2`)`
    for $n \leftarrow 1$ **to** N **do** in parallel
      `// compute lookahead mean vector`
      for $k_1 \leftarrow K_3$ **to** $K_1 + 1$ **do**
        $\overleftarrow{\boldsymbol{M}}[n, K_1+1] \leftarrow \overleftarrow{\tilde{\lambda}}_n \overleftarrow{\boldsymbol{M}}[n, K_1+1] + \overleftarrow{\tilde{f}}_n(\boldsymbol{s}[k_1])$
      end
      `// compute batch of estimates`
      for $k_2 \leftarrow 1$ **to** K_1 **do**
        $\overrightarrow{\boldsymbol{M}}[n, k_2] \leftarrow \overrightarrow{\tilde{\lambda}}_n \overrightarrow{\boldsymbol{M}}[n, k_2-1] + \overrightarrow{\tilde{f}}_n(\boldsymbol{s}[k_2-1])$
        $k_3 \leftarrow K_1 - k_2 + 1$
        $\overleftarrow{\boldsymbol{M}}[n, k_3] \leftarrow \overleftarrow{\tilde{\lambda}}_n \overleftarrow{\boldsymbol{M}}[n, k_3+1] + \overleftarrow{\tilde{f}}_n(\boldsymbol{s}[k_3])$
      end
    end
    for $\ell \leftarrow 1$ **to** L **do** in parallel
      for $k_4 \leftarrow 1$ **to** K_1 **do** in parallel
        $\hat{\boldsymbol{u}}[\ell, k_4-1] \leftarrow \sum_{n=1}^{N} \overrightarrow{\tilde{w}}_{n,\ell} \overrightarrow{\boldsymbol{M}}[n, k_4] + \overleftarrow{\tilde{w}}_{n,\ell} \overleftarrow{\boldsymbol{M}}[n, k_4]$
      end
    end
    `// prepare mean vectors for next batch`
    $\overrightarrow{\boldsymbol{M}}[0] \leftarrow \overrightarrow{\boldsymbol{M}}[K_1-1]$; $\overleftarrow{\boldsymbol{M}}[K_1+1] \leftarrow \mathbf{0}_N$
    `// shift control signal buffer`
    $\boldsymbol{s}[k] \leftarrow \boldsymbol{s}[\mathrm{mod}(k+K_1, K_3)]$
    `// write batch to` *outStream* `as in Algorithm 6`
    `SubmitEstimatesToStream(`$\hat{\boldsymbol{u}}$*, outStream,* K_1`)`
  until inStream *closes*
end

Algorithm 7: Parallel online filter computations.

Bibliography

[1] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. New York: Prentice Hall, 1979.

[2] J. Biveroni, "On A/D converters with low-precision analog circuits and digital post-correction," Ph.D. dissertation, ETH Zurich, 2012.

[3] L. Bolliger, H. Loeliger, and C. Vogel, "Simulation, MMSE estimation, and interpolation of sampled continuous-time signals using factor graphs," in *2010 Information Theory & Applications Workshop (ITA)*, 2010, pp. 1–4.

[4] L. Bolliger, H. Loeliger, and C. Vogel, "LMMSE estimation and interpolation of continuous-time signals from discrete-time samples using factor graphs," 2013, arXiv: 1301.4793 [cs.IT].

[5] L. Bruderer and H.-A. Loeliger, "Estimation of sensor input signals that are neither bandlimited nor sparse," in *2011 Information Theory & Application Workshop (ITA)*, San Diego, CA, USA, Feb. 9-14 2014.

[6] L. Bruderer, "Input estimation and dynamical system identification: New algorithms and results," Ph.D. dissertation, ETH Zurich, 2015.

[7] P. M. Chopp and A. A. Hamoui, "A 1-V 13-mW single-path frequency-translating $\Delta\Sigma$ modulator with 55-dB SNDR and 4-MHz bandwidth at 225-MHz," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 2, pp. 473–486, 2013.

[8] J. M. de la Rosa, "Sigma-Delta modulators: Tutorial overview, design guide, and state-of-the-art survey," *IEEE Transactions on Circuits & Systems I*, vol. 58, no. 1, pp. 1–21, 2011.

[9] J. M. de la Rosa, R. Schreier, K.-P. Pun, and S. Pavan, "Next-generation Delta-Sigma converters: Trends and perspectives," *IEEE Journal of Emerging and Selected Topics in Circuits & Systems*, vol. 5, no. 4, pp. 484–489, 2015.

[10] Y. E. Eldar, *Sampling Theory: Beyond Bandlimited Systems*, 1st ed. Cambridge, UK: Cambridge University Press, 2015.

[11] S. Haykin, *Communicaton Systems*, 4th ed. John Wiley & Sons, Inc., 2001.

[12] K. Hosseini and M. P. Kennedy, "Maximum sequence length MASH digital Delta-Sigma modulators," *IEEE Transactions on Circuits & Systems I*, vol. 54, no. 12, pp. 2628–2638, 2007.

[13] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. New York: Prentice Hall, 2000.

[14] A. Kipnis, Y. C. Eldar, and A. J. Goldsmith, "Analog-to-digital compression: A new paradigm for converting signals to bits," *IEEE Signal Processing Magazine*, vol. 35, no. 3, pp. 16–39, May 2018.

[15] H. Landau, "Sampling, data transmission, and the Nyquist rate," *Proc. IEEE*, vol. 55, no. 10, pp. 1701–1706, 1967.

[16] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Prentice Hall, 1999.

[17] H.-A. Loeliger, L. Bolliger, G. Wilckens, and J. Biveroni, "Analog-to-digital conversion using unstable filters," in *2011 Information Theory & Application Workshop (ITA)*, UCSD, La Jolla, CA, USA, Feb. 6-11 2011.

[18] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.

[19] H.-A. Loeliger and G. Wilckens, "Control-based analog-to-digital conversion without sampling and quantization," in *2011 Information Theory & Application Workshop (ITA)*, San Diego, CA, USA, Feb. 1-6 2015.

[20] H.-A. Loeliger, H. Malmberg, and G. Wilckens, "Control-bounded analog-to-digital conversion: Transfer function analysis, proof of concept, and digital filter implementation," 2020, arXiv: 2001.05929.

[21] S. Mallat, *A wavelet tour of signal processing*, 3rd ed. San Diego, USA: University Press, 2008.

[22] M. Ortmanns and F. Gerfers, *Continuous-Time Sigma-Delta A/D Conversion: Fundamentals, Performance Limits and Robust Implementations*, 1st ed. Berlin: Springer, 2005.

[23] S. Pamarti and I. Galton, "Lsb dithering in MASH Delta-Sigma D/A converters," *IEEE Transactions on Circuits & Systems I*, vol. 54, no. 4, pp. 779–790, 2007.

[24] S. Pamarti, J. Welz, and I. Galton, "Statistics of the quantization noise in 1-bit dithered single-quantizer digital Delta-Sigma modulators," *IEEE Transactions on Circuits & Systems I*, vol. 54, no. 3, pp. 492–503, 2007.

[25] S. Pavan, R. Schreier, and G. C. Temes, *Understanding Delta-Sigma Data Converters*, 2nd ed. New York: Wiley, 2017.

[26] R. Schreier and B. Zhang, "Delta-Sigma modulators employing continuous-time circuitry," *IEEE Transactions on Circuits & Systems I*, vol. 43, no. 4, pp. 324–332, 1996.

[27] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[28] J. Song and I.-C. Park, "Spur-free MASH Delta-Sigma modulation," *IEEE Transactions on Circuits & Systems I*, vol. 57, no. 9, pp. 2426–2437, 2010.

[29] H. Tao and J. M. Khoury, "A 400-Ms/s frequency translating bandpass Sigma-Delta modulator," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 12, pp. 1741–1752, 1999.

[30] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22–38, 1999.

[31] M. Unser and P. D. Tafti, *An introduction to Sparse Stochastic Processes*, 1st ed. Cambridge, UK: Cambridge University Press, 2014.

[32] R. J. van de Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*, 2nd ed. Boston: Kluwer Academic Publishers, 2003.

[33] G. Venturini, "python-deltasigma," [Online]. Avaliable: http://www.python-deltasigma.io, 2016.

[34] F. Wadehn, L. Bruderer, J. Dauwels, V. Sahdeva, H. Yu, and H. Loeliger, "Outlier-insensitive Kalman smoothing and marginal message passing," in *2016 24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1242–1246.

[35] F. Wadehn, "State space methods with applications in biomedical signal processing," Ph.D. dissertation, ETH Zurich, 2019.

[36] P. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[37] E. T. Whittacker, "On the functions which are represented by the expansions of the interpolation theory," *Proc. Roy. Soc. Edinburgh*, vol. 35, pp. 181–194, 1915.

[38] G. Wilckens, "A new perspective on analog-to-digital conversion of continuous-time signals," Ph.D. dissertation, ETH Zurich, 2013.

Index

About the Author

Hampus Malmberg was born in Gothenburg, Sweden, on the 21st of July in 1988. In 2009 he enrolled in the electrical engineering program at Chalmers University of Technology in Gothenburg, Sweden, from which he completed his BSc degree in electrical engineering in 2012. During this time, he also did an internship for the biomedical company Unfors RaySafe AB. Subsequently, In 2014 he received his MSc degree in electrical engineering and information technology from ETH Zürich.

Since 2014, he has been a PhD candidate and research assistant at the Signal and Information Processing Laboratory (ISI) under the supervision of Prof. Hans-Andrea Loeliger.

Except for A/D and D/A conversion, his research interests also include sparse Bayesian learning, Gaussian message passing, machine learning, and electronics.

Furthermore, he has a profound interest in scientific computing, programming, and teaching.

Series in Signal and Information Processing

edited by Hans-Andrea Loeliger

Vol. 1: Hanspeter Schmid, **Single-Amplifier Biquadratic MOSFET-C Filters.** ISBN 3-89649-616-6

Vol. 2: Felix Lustenberger, **On the Design of Analog VLSI Iterative Decoders**. ISBN 3-89649-622-0

Vol. 3: Peter Theodor Wellig, **Zerlegung von Langzeit-Elektromyogrammen zur Prävention von arbeitsbedingten Muskelschäden**. ISBN 3-89649-623-9

Vol. 4: Thomas P. von Hoff, **On the Convergence of Blind Source Separation and Deconvolution.** ISBN 3-89649-624-7

Vol. 5: Markus Erne, **Signal Adaptive Audio Coding using Wavelets and Rate Optimization.** ISBN 3-89649-625-5

Vol. 6: Marcel Joho, **A Systematic Approach to Adaptive Algorithms for Multichannel System Identification, Inverse Modeling, and Blind Identification.** ISBN 3-89649-632-8

Vol. 7: Heinz Mathis, **Nonlinear Functions for Blind Separation and Equalization.** ISBN 3-89649-728-6

Vol. 8: Daniel Lippuner, **Model-Based Step-Size Control for Adaptive Filters.** ISBN 3-89649-755-3

Vol. 9: Ralf Kretzschmar, **A Survey of Neural Network Classifiers for Local Wind Prediction.** ISBN 3-89649-798-7

Vol. 10: Dieter M. Arnold, **Computing Information Rates of Finite State Models with Application to Magnetic Recording.** ISBN 3-89649-852-5

Vol. 11: Pascal O. Vontobel, **Algebraic Coding for Iterative Decoding.** ISBN 3-89649-865-7

Vol. 12: Qun Gao, **Fingerprint Verification using Cellular Neural Networks.** ISBN 3-89649-894-0

Vol. 13: Patrick P. Merkli, **Message-Passing Algorithms and Analog Electronic Circuits.** ISBN 3-89649-987-4

Vol. 14: Markus Hofbauer, **Optimal Linear Separation and Deconvolu-tion of Acoustical Convolutive Mixtures.** ISBN 3-89649-996-3

Vol. 15: Sascha Korl, **A Factor Graph Approach to Signal Modelling, System Identification and Filtering.** ISBN 3-86628-032-7

Vol. 16: Matthias Frey, **On Analog Decoders and Digitally Corrected Converters.** ISBN 3-86628-074-2

Vol: 17: Justin Dauwels, **On Graphical Models for Communications and Machine Learning: Algorithms, Bounds, and Analog Implementation.** ISBN 3-86628-080-7

Vol. 18: Volker Maximillian Koch, **A Factor Graph Approach to Model-Based Signal Separation.** ISBN 3-86628-140-4

Vol. 19: Junli Hu, **On Gaussian Approximations in Message Passing Al-gorithms with Application to Equalization.** ISBN 3-86628-212-5

Vol. 20: Maja Ostojic, **Multitree Search Decoding of Linear Codes.** ISBN 3-86628-363-6

Vol. 21: Murti V.R.S. Devarakonda, **Joint Matched Filtering, Decoding, and Timing Synchronization.** ISBN 3-86628-417-9

Vol. 22: Lukas Bolliger, **Digital Estimation of Continuous-Time Signals Using Factor Graphs.** ISBN 3-86628-432-2

Vol. 23: Christoph Reller, **State-Space Methods in Statistical Signal Processing: New Ideas and Applications.** ISBN 3-86628-447-0

Vol. 24: Jonas Biveroni, **On A/D Converters with Low-Precision Analog Circuits and Digital Post-Correction.** ISBN 3-86628-452-7

Vol. 25: Georg Wilckens, **A New Perspective on Analog-to-Digital Conversion of Continuous-Time Signals.** ISBN 3-86628-469-1

Vol. 26: Jiun-Hung Yu, **A Partial-Inverse Approach to Decoding Reed-Solomon Codes and Polynomial Remainder Codes.** ISBN 3-86628-527-2

Vol. 27: Lukas Bruderer, **Input Estimation and Dynamical System Identification: New Algorithms and Results.** ISBN 3-86628-533-7

Vol. 28: Sarah Neff, **A New Approach to Information Processing with Filters and Pulses.** ISBN 3-86628-575-2

Vol. 29: Christian Schürch, **On Successive Cancellation Decoding of Polar Codes and Related Codes.** ISBN 3-86628-580-9

Vol. 30: Nour Zalmaï, **A State Space World for Detecting and Estimating Events and Learning Sparse Signal Decompositions.** ISBN 978-3-86628-594-1

Vol. 31: Federico Wadehn, **State Space Methods with Applications in Biomedical Signal Processing.** ISBN 978-3-86628-640-5

Vol. 32: Reto A. Wildhaber, **Localized State Space and Polynomial Filters with Applications in Electrocardiography**. ISBN 978-3-86628-652-8

Hartung-Gorre Verlag, Konstanz – http://www.hartung-gorre.de