Control-Bounded ADC

Fredrik Feyling

December 2020

List of Symbols

Matrices and Vectors

a	a scalar value
$\boldsymbol{a}$	a column vector $(a_1 \ \cdots \ a_N)^\mathsf{T} \in \mathbb{R}^N$
$\boldsymbol{A}$	a matrix $\begin{pmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{M1} & \cdots & a_{MN} \end{pmatrix} \in R^{M \times N}$
$\mathbf{0}_N$	an all-zero column vector of length N
$\mathbf{0}_{M \times N}$	an M-by-N all-zero matrix
$\mathbf{1}_N$	a column vector of length N with all elements 1
$\mathbf{1}_{M \times N}$	an M-by-N matrix with all elements 1
$\boldsymbol{I}_N$	an N-by-N matrix with ones on the main diagonal and all other elements zero
$\boldsymbol{H}_2$	second order Hadamard matrix $\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
$\boldsymbol{H}_N$	Hadamard matrix of order N defined by $\boldsymbol{H}_2 \otimes \boldsymbol{H}_{N/2}$
$\boldsymbol{H}'_N$	modified Hadamard matrix $\begin{bmatrix} \boldsymbol{H}_{N/2} & \mathbf{0}_{N/2 \times N/2} \\ \mathbf{0}_{N/2 \times N/2} & \boldsymbol{H}_{N/2} \end{bmatrix}$
$\otimes$	Kronecker product
$()^\mathsf{T}$	transpose
$\lvert a \rvert$	absolute value
$\lvert\lvert \boldsymbol{b} \rvert\rvert_p$	p-norm $(\Sigma_i \lvert b_i \rvert^p)^{1/p}$
$\lvert\lvert \boldsymbol{c} \rvert\rvert_\infty$	max norm, equivalent to $\max(\lvert c_1 \rvert, \lvert c_2 \rvert, \cdots \lvert c_N \rvert)$

Sets

$\mathbb{R}$	the real numbers

Miscellaneous

$\dot{\boldsymbol{x}}$	elementwise time derivative $\frac{d}{dt}\boldsymbol{x}(t)$

Control-Bounded Conversion

L	input signal dimension
N_ℓ	system order corresponding to each input channel
N	total system order LN_ℓ
β	integrator gain
$\boldsymbol{A}$	system matrix
$\boldsymbol{B}$	input matrix
$\boldsymbol{C}$	signal observation (output) matrix
$\mathbf{\Gamma}$	control input matrix
$\tilde{\mathbf{\Gamma}}$	control observation matrix
$\boldsymbol{u}(t)$	input signal
$\boldsymbol{u}[k]$	input samples
$\hat{\boldsymbol{u}}[k]$	estimated input samples
$\boldsymbol{x}(t)$	state vector
$\boldsymbol{s}[k]$	control signal
$\boldsymbol{s}(t)$	control contribution
$\tilde{\boldsymbol{s}}(t)$	control observation
$\boldsymbol{y}(t)$	signal observation
$\breve{\boldsymbol{y}}(t)$	fictional signal observation
$\boldsymbol{G}(\omega)$	analog transfer function (ATF) matrix
$\boldsymbol{H}(\omega)$	noise transfer function (NTF) matrix
$\boldsymbol{G}(\omega)\boldsymbol{H}(\omega)$	signal transfer function (STF) matrix

Acronyms

AS	analog system
ATF	analog transfer function
DC	digital control
DE	digital estimator
NTF	noise transfer function
STF	signal transfer function

Chapter 1

Introduction

Chapter 2

Background Theory

2.1 Oversampling A/D Converters

It will become apparent that the control-bounded ADC shares some similarities with conventional oversampling converters, and in particular the continuous-time ΣΔ converter. In order to show where the control-bounded converter distinguishes from these architectures, a brief introduction to conventional oversampling ADCs is included in this section. The presented material is assumed well known to the reader, and is only included to establish a basis of comparison. For a proper introduction to the topic, the reader is referred to [1].

An oversampling ADC is based on sampling the input signal at a frequency much higher than the Nyquist rate. For an analog input signal that is bandlimited to f_0, we define the oversampling ratio as

$$\mathrm{OSR} \triangleq \frac{f_s}{2f_0} \tag{2.1}$$

where f_s is the sampling frequency of the ADC. Sampling at a higher frequency generates redundant information about the input signal, and a single estimate of the input signal is typically obtained by averaging several consecutive samples. The redundancy is this way utilized to give a higher resolution, or equivalently reduced requirements on the involved circuit components.

Straight forward oversampling will itself give an improved signal-to-noise ratio (SNR) of 3dB per doubling of OSR [1]. The performance of the oversampling converter is further improved by noise shaping of the quantization noise, through a feedback loop with a loop filter. Such a system is known as a ΣΔ ADC and the part of the system that performs the noise shaping is called a ΣΔ modulator. Such a system is illustrated in figure 2.1. In figure 2.1, the box labeled "S/H" performs the sample-and-hold operation, and passes this discrete-time signal to the ΣΔ modulator. The ΣΔ modulator

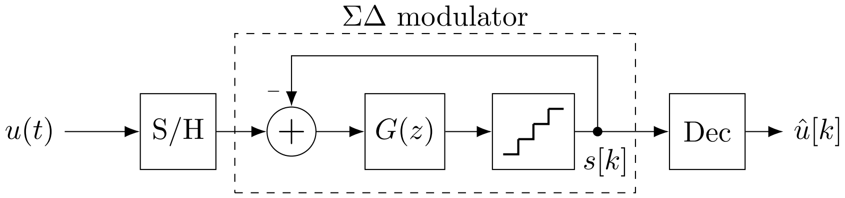

Figure 2.1: A discrete-time ΣΔ ADC.

performs a quantization of the signal, together with a noise shaping of the quantization noise. It is common to include an anti-aliasing filter in front of the S/H-operation.

The system shown in figure 2.1 is called a discrete-time ΣΔ ADC because the ΣΔ modulator has a discrete-time input. A continuous-time ΣΔ converter is achieved by including the sampling in the feedback loop, as shown in figure 2.2. In this case, an eventual anti-aliasing filter is part of loop filter $G(\omega)$.

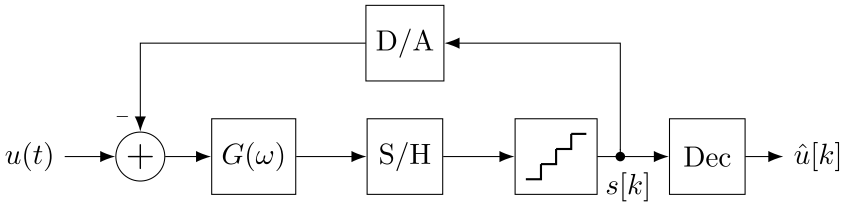

Figure 2.2: A continuous-time ΣΔ ADC.

2.1.1 Transfer Function Analysis

A transfer function analysis of the discrete-time ΣΔ modulator is obtained by evaluating the linearized model shown in figure 2.3. The analysis of the continuous-time modulator is similar. This model approximates the

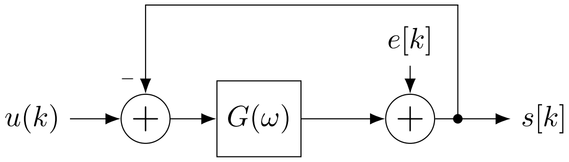

Figure 2.3: A simplified, linear model of the discrete-time ΣΔ ADC.

quantization error as a signal being independent of the input. This can of course not be strictly true, but is a useful approximation for the analysis.

Let $U(z), E(z)$ and $S(z)$ be the signals $u[k], e[k]$ and $s[k]$ after applying the z-transform. The modulator output is given by

$$S(z) = E(z) + G(z)[U(z) - S(z)], \tag{2.2}$$

and the signal- and noise transfer function can then be recognized by evaluating

$$S = \underbrace{\frac{1}{1+G}}_{\text{NTF}} E + \underbrace{\frac{G}{1+G}}_{\text{STF}} U. \tag{2.3}$$

as NTF $= \frac{1}{1+G}$ and STF $= \frac{G}{1+G}$. Because the input signal and the quantization noise experience different transfer functions, it is possible to shape the noise such that most of the quantization noise appears outside the frequency band of interest, while simultaneously leaving the actual signal unchanged. This is the effect known as noise shaping. Note here that the necessary condition for noise shaping is that the signal and the quantization error enters the system at different points in the signal flow.

Chapter 3

Control-Bounded ADC

3.1 History and Background

Control-bounded A/D conversion is a conceptually new approach to the problem of creating a digital representation of an analog signal. The conversion technique has developed quite recently over the last years, and the progress is mainly pushed forward by prof. Hans-Andrea Loeliger et al., from the Signal and Information Processing Laboratory (ISI), ETH Zürich. The concept was first introduced at the IEEE Information Theory & Applications Workshop (ITA), february 2011 [2]. In this paper, the main building blocks of a control-bounded ADC was presented, but no explicit example of such an ADC was given, and no behavioural analysis presented. The approach was further developed in [3], which was published for the same conference in 2015. In this paper, the conversion algorithm is improved and a limited transfer function analysis is presented. The latest publication on control-bounded conversion is from 2020 [4]. This is a longer paper with the goal of providing the sufficient information for analog designers to experiment with control-bounded ADCs. The paper provides a more details on the implementation and operation of the building blocks, together with a full transfer function analysis. Measurements on a proof-of-concept hardware prototype is also presented.

In addition to the mentioned papers, Hampus Malmberg, co-author of the latest paper [4], has recently defended his Ph.D. on Control-Bounded Converters. The author of this paper has been given early access to a draft of the thesis that is not yet published [5], and this draft serves as the main source of information on this topic. Malmberg also held a presentation at the 2020 IEEE International Symposium on Circuits and Systems (ISCAS) [6], where he presented the basic concept of control-bounded ADC together with the idea of the Hadamard ADC which is a basis for this work.

In this section, the operating principle of a control-bounded converter is described in detail, and we follow the notation established in [4]. The theoretical presentation given in this section will be very close to that of [5], but less general and limited to what is necessary for understanding the presented results.

3.2 Overview

The control-bounded ADC approaches the A/D conversion problem differently compared to conventional A/D converters. The conceptual difference lies in the view on sampling. In a control-bounded converter, the analog input signal is never sampled in the traditional way. The circuit that constitutes a control-bounded ADC still contains quantizers, but the quantized signals are never treated as a sampled version of the input. Instead, they are intermediate digital signals that only indirectly relates to the input, and they are used by a digital estimation filter to perform the digital estimate of the input signal. This way, rather than the process of performing accurate measurements of an analog signal using imperfect circuit components, sampling becomes the process of converting a redundant digital representation into an efficient one [5].

To clarify this, consider the general block diagram shown in figure 3.1.

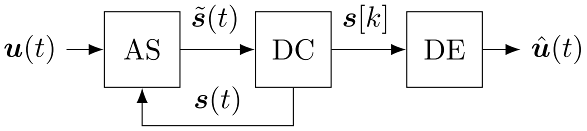

Figure 3.1: A block diagram of the control-bounded ADC.

As the figure indicates, the control-bounded ADC consists of three main building blocks; an analog system (AS), a digital control (DC) and a digital estimator (DE). The signals $\boldsymbol{u}(t), \hat{\boldsymbol{u}}(t), \tilde{\boldsymbol{s}}(t), \boldsymbol{s}[k]$ and $\boldsymbol{s}(t)$ are in general vector-valued functions. The analog system amplifies the input signal $\boldsymbol{u}(t)$, preferably with very high gain within the frequency band of interest. The digital control stabilizes the analog system by forcing the internal states of the system to stay within its bounds. The internal states are observed through the *control observation* $\tilde{\boldsymbol{s}}(t)$ and controlled through the *control contribution* $\boldsymbol{s}(t)$. The digital estimator takes the *control signal* $\boldsymbol{s}[k]$ as an input and forms the digital estimate $\hat{\boldsymbol{u}}(t)$ of $\boldsymbol{u}(t)$.

Note that the output of the digital estimator is denoted as a continuous-time estimate $\hat{\boldsymbol{u}}(t)$ instead of a discrete-time estimate $\hat{\boldsymbol{u}}[k]$. The digital estimator models the continuous-time dynamics of the analog system, and is thereby capable of estimating $\boldsymbol{u}(t)$ at arbitrary time instances. The actual estimates will obviously be computed at discrete time steps, but because the digital estimator itself imposes no criteria on this time interval, the output is denoted as a continuous time estimate.

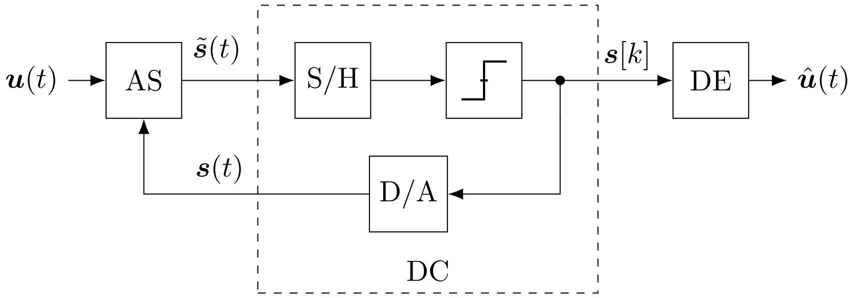

Figure 3.2: A control-bounded ADC with the DC box opened

Before going into detail on each of these building blocks, consider figure 3.2, which shows the same block diagram, but with the DC box opened. From this figure it is evident that the structure of the control-bounded ADC is very similar to that of the continuous-time ΣΔ modulator in figure 2.2. The control-bounded converter may in fact be viewed as a generalization of the continuous-time ΣΔ ADC. As mentioned, the main difference between these architectures arises from the different interpretation of the control signal $\boldsymbol{s}[k]$. In the ΣΔ ADC, this signal is viewed as a filtered, sampled and quantized version of the input signal and the digital output is obtained by averaging this signal through a decimation filter. In the control-bounded perspective the direct relation between $\boldsymbol{s}[k]$ and $\boldsymbol{u}(t)$ is ignored completely. Instead, we focus solely on the fact that $\boldsymbol{s}(t)$ is the contribution needed to stabilize the internal states of the analog system. This view leeds to a different estimation filter for the reconstruction of $\hat{\boldsymbol{u}}(t)$.

It should be noted that the contribution of the control-bounded ADC is not to provide an alternative decimation filter to already existing ΣΔ ADCs. As shown in section 2.1.1, the noise shaping of the ΣΔ modulator relies on the fact that the signal and the quantization noise enters the system at different points in the signal flow. This condition is a major restriction to the design space of ΣΔ modulators. The estimation filter of the control-bounded converter on the other hand, imposes no restrictions to the analog system. Hence, the ADC can be designed with combinations of analog system and digital control that have previously been unimaginable. The advantage of this will become more apparent when considering the Hadamard ADC in chapter 5.

3.3 Analog System

The analog system, here assumed to be a continuous time filter, sets the frequency response of the overall ADC, and is designed to amplify the frequency band of interest. As stability of the analog system is controlled digitally, the analog system itself need not be stable.

3.3.1 State Space Model

The dynamics of the analog system is described using a state space model notation, illustrated in figure 3.3. The multi-channel input signal $\boldsymbol{u}(t)$, the state-vector $\boldsymbol{x}(t)$ and the control contribution $\boldsymbol{s}(t)$ is related by the differential equation

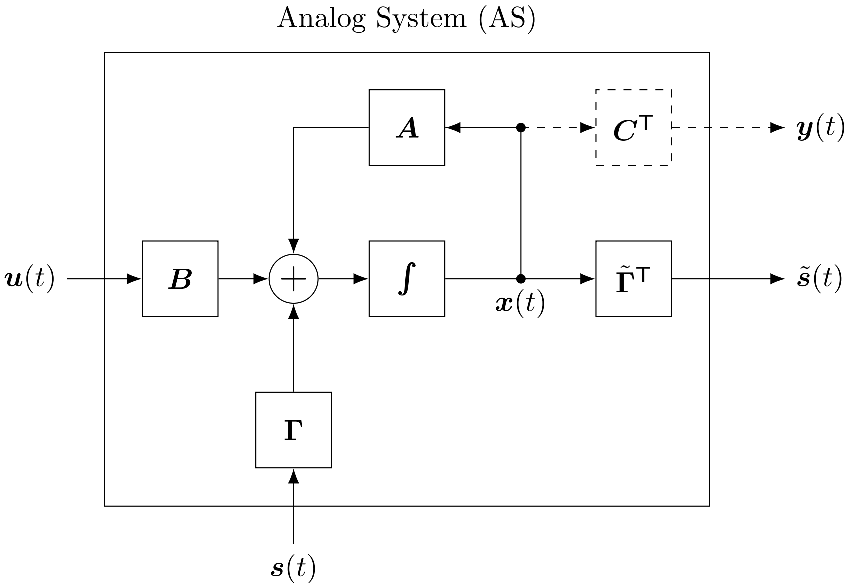

Figure 3.3: State space model of the AS. Figure from [5].

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{\Gamma}\boldsymbol{s}(t). \tag{3.1}$$

where

$$\boldsymbol{u}(t) \triangleq (u_1(t), ..., u_L(t))^\mathsf{T} \in \mathbb{R}^L, \tag{3.2}$$

$$\boldsymbol{x}(t) \triangleq (x_1(t), ..., x_N(t))^\mathsf{T} \in \mathbb{R}^N \tag{3.3}$$

and

$$\boldsymbol{s}(t) \triangleq (s_1(t), ..., s_M(t))^\mathsf{T} \in \mathbb{R}^M. \tag{3.4}$$

This system is said to have L inputs, M controls and N states. We will refer to $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, $\boldsymbol{B} \in \mathbb{R}^{N\times L}$ and $\boldsymbol{\Gamma} \in \mathbb{R}^{N\times M}$ as the *system matrix*, the *input matrix* and *the control input matrix* respectively.

The only physical output of the analog system is the control observation

$$\tilde{\boldsymbol{s}}(t) \triangleq \tilde{\boldsymbol{\Gamma}}^\mathsf{T}\boldsymbol{x}(t) \in \mathbb{R}^{\tilde{M}}, \tag{3.5}$$

which is used by the digital control to produce the control signal $\boldsymbol{s}[k]$. The control observation is a linear mapping of the internal state-vector, through the *control observation matrix* $\tilde{\mathbf{\Gamma}}^{\mathsf{T}} \in \mathbb{R}^{\tilde{M}\times N}$. The second output of the analog system is the purely conceptual signal

$$\boldsymbol{y}(t) \triangleq \boldsymbol{C}^{\mathsf{T}}\boldsymbol{x}(t) \in \mathbb{R}^{\tilde{N}}, \tag{3.6}$$

which is used by the digital estimator to produce the estimate $\hat{\boldsymbol{u}}(t)$. This signal has no physical meaning, and the *signal observation matrix* $\boldsymbol{C}^{\mathsf{T}} \in \mathbb{R}^{\tilde{N}\times N}$ is basically telling the digital estimation algorithm which of the internal states that could be treated as bounded. Thus $\boldsymbol{y}(t)$ and $\boldsymbol{C}^{\mathsf{T}}$ does only exist conceptually inside the digital estimator.

3.3.2 Transfer Function and Impulse Response Matrix

The transfer function of the analog system gives the frequency domain relation between the input $\boldsymbol{U}(\omega)$ and the output $\boldsymbol{Y}(\omega)$. Hence for the general case of L inputs and $\tilde{N}$ outputs, the analog transfer function (ATF) is a $\tilde{N}$-by-L matrix, defined by $\boldsymbol{Y}(\omega) = \boldsymbol{G}(\omega)\boldsymbol{U}(\omega)$. Each element $G_{i,j}(\omega)$ of $\boldsymbol{G}(\omega)$ is the transfer function from $U_j(\omega)$ to $Y_i(\omega)$. From (3.1) the ATF is obtained as

$$\boldsymbol{G}(\omega) = \boldsymbol{C}^{\mathsf{T}}\left(j\omega\boldsymbol{I}_N - \boldsymbol{A}\right)^{-1}\boldsymbol{B}, \tag{3.7}$$

and the derivation is given in appendix A. The analog impulse response matrix is then obtained from the inverse Laplace transform as

$$\boldsymbol{g}(t) = \boldsymbol{C}^{\mathsf{T}}\exp(\boldsymbol{A}t)\boldsymbol{B}, \tag{3.8}$$

where exp denotes the matrix exponential.

3.4 Digital Control

The digital control is a discrete time system which serves the purpose of stabilizing the analog system. It includes a sample-and-hold circuit, a one-bit quantizer and a D/A converter, as shown in figure 3.2. The control observation $\tilde{\boldsymbol{s}}(t)$ is sampled and quantized with a period T, resulting in the digital control signal $\boldsymbol{s}[k]$ which is passed on to the digital estimator. The D/A converter is a non-return to zero (NRZ) DAC generating the control contribution $\boldsymbol{s}(t)$.

The digital control is called effective if it manages to keep the state vector bounded, given a bounded input vector. The input vector $\boldsymbol{u}(t)$ is bounded if it satisfies

$$||\boldsymbol{u}(t)||_{\infty} \leq b_{\boldsymbol{u}} \quad \forall t. \tag{3.9}$$

Equivalently, the state vector $\boldsymbol{x}(t)$ is bounded if it satisfies

$$||\boldsymbol{x}(t)||_{\infty} \leq b_{\boldsymbol{x}} \quad \forall t. \tag{3.10}$$

In this paper, the input signal will always be assumed bounded, and the boundary $b_{\boldsymbol{u}}$ is assumed to be determined by an external circuit. The boundary for the state vector, $b_{\boldsymbol{x}}$, is a free variable and determines the magnitude of the state vector of the analog system.

A thorough analysis of the criteria for an effective control is found in [5]. The analysis is useful for the theoretical understanding of the system, but not necessary for the design process and is therefore beyond the scope of this paper. Intuitively, there are three quantities affecting the stability of the analog system. The sampling period T of the digital control, the unity gain frequency of the analog system and the boundary $b_{\boldsymbol{x}}$. Increasing the speed of the analog system would require a shorter sampling period to counteract the faster growth of the system states. Reducing the boundary $b_{\boldsymbol{x}}$ would require either reducing the speed of the analog system or increasing the sampling frequency, in order to maintain a tighter bound.

It will become apparent in the next section that the performance of the overall ADC is related to the digital controls ability to bound the state vector. Designing the ADC for a stability guarantee means that it is theoretically impossible for the state vector to grow beyond $b_{\boldsymbol{x}}$ at any point in time, given any valid input signal. This will of course result in a very large stability margin most of the time, which means that there is potential for increased performance not being utilized. The preferred way of tuning the stability of the system is therefore through simulations, and then to include the possibility of a full system reset if it happens to become unstable.

3.5 Digital Estimator

The digital estimator (DE) forms an estimate $\hat{\boldsymbol{u}}(t)$ of $\boldsymbol{u}(t)$ based on the control signals $\boldsymbol{s}[k]$ and the knowledge of the AS system parameters. The purpose of this section is to describe the digital estimation problem, and to derive the optimum linear estimation filter.

3.5.1 Statistical Estimation Problem and Transfer Functions

In the following analysis, the system described by (3.1) is assumed to be invariant and stable. This assumption only applies in the analysis of this section, where the goal is to describe the estimation problem and derive the analytic transfer function expressions. The actual estimation filter will not be limited by these assumptions.

The objective of the digital estimator is to construct a digital estimate $\hat{\boldsymbol{u}}(t)$ of $\boldsymbol{u}(t)$, based on the control signals $\boldsymbol{s}[k]$. As highlighted previously in this chapter, the direct relation between $\boldsymbol{s}[k]$ and $\boldsymbol{u}(t)$ is ignored completely by the digital estimator. Instead, $\boldsymbol{s}[k]$ is only treated as the signal needed to stabilize the analog system, when triggered by an input signal $\boldsymbol{u}(t)$.

To formalize this approach, let $\breve{\boldsymbol{y}}(t) \triangleq (\boldsymbol{g} * \boldsymbol{u})(t) \in \mathbb{R}^{\tilde{N}}$ be the signal that would have occurred at the output of the analog system in the absence of any digital control. Futhermore, let $\boldsymbol{q}(t)$ be the control contribution signal seen at the output of the analog system. Because the control contribution enters the analog system in an additive way, we can express the relation as

$$\boldsymbol{y}(t) = \breve{\boldsymbol{y}}(t) - \boldsymbol{q}(t). \tag{3.11}$$

The situation is illustrated in figure 3.4. In this figure, solid lines represent the physical components of the ADC, while dashed lines represents conceptual quantities that only exist inside the digital estimator. It is illustrated how $\boldsymbol{q}(t)$ relates to the control contribution $\boldsymbol{s}(t)$. Because the digital estimator knows the parametrization of the analog system, as well as the waveform of the D/A converter, $\boldsymbol{q}(t)$ is (in principle) known from the observation of $\boldsymbol{s}[k]$. Note that this illustration is only meant to illustrate the estimation problem of the digital estimator, not to show how the actual estimate is computed. We denote the frequency response of the digital estimation filter by $\boldsymbol{H}(\omega)$ and the continuous time estimate $\hat{\boldsymbol{u}}(t)$ of $\boldsymbol{u}(t)$ is obtained by

$$\hat{\boldsymbol{u}}(t) = (\boldsymbol{h} * \boldsymbol{q})(t) \in \mathbb{R}^{L}. \tag{3.12}$$

Because the objective of the analog system is to greatly amplify the sought frequency content of $\boldsymbol{u}(t)$, both $||\breve{\boldsymbol{y}}(t)||_\infty$ and $||\boldsymbol{q}(t)||_\infty$ will be very large compared to $||\boldsymbol{y}(t)||_\infty$, which is bounded due to (3.10). We can therefore approximate $\breve{\boldsymbol{y}}(t) \approx \boldsymbol{q}(t)$ which is equivalent to the approximation $\boldsymbol{y}(t) \approx \boldsymbol{0}$. Hence the estimate may be written as

$$\begin{aligned}
\hat{\boldsymbol{u}}(t) &= (\boldsymbol{h} * \boldsymbol{q})(t) && (3.13)\\
&= (\boldsymbol{h} * \breve{\boldsymbol{y}})(t) - (\boldsymbol{h} * \boldsymbol{y})(t) && (3.14)\\
&\approx (\boldsymbol{h} * \breve{\boldsymbol{y}})(t) && (3.15)\\
&= (\boldsymbol{h} * \boldsymbol{g} * \boldsymbol{u})(t) && (3.16)
\end{aligned}$$

It is evident that $\hat{\boldsymbol{u}}(t)$ could have been computed with arbitrary accuracy, if the output $\boldsymbol{y}(t)$ was known to the digital estimator. This if statement is obvious, but it illustrates an important point. Instead of relying on an inevitably inaccurate measurement of $\boldsymbol{y}(t)$, we choose to approximate this signal as constantly being zero. The accuracy of the estimate then relies on the validity of the approximation $\boldsymbol{y}(t) \approx \boldsymbol{0}$, rather than the precision of a direct measurement of $\boldsymbol{y}(t)$. This approximation is illustrated in figure 3.4, where it is indicated that the actual analog system output is disregarded and substituted with the fictional observation $\tilde{\boldsymbol{y}}(t) = \boldsymbol{0}$.

Any deviation of $\boldsymbol{y}(t)$ from $\boldsymbol{0}$ will result in a conversion error, meaning that $\boldsymbol{y}(t)$ is the conversion error signal seen at the output of the analog

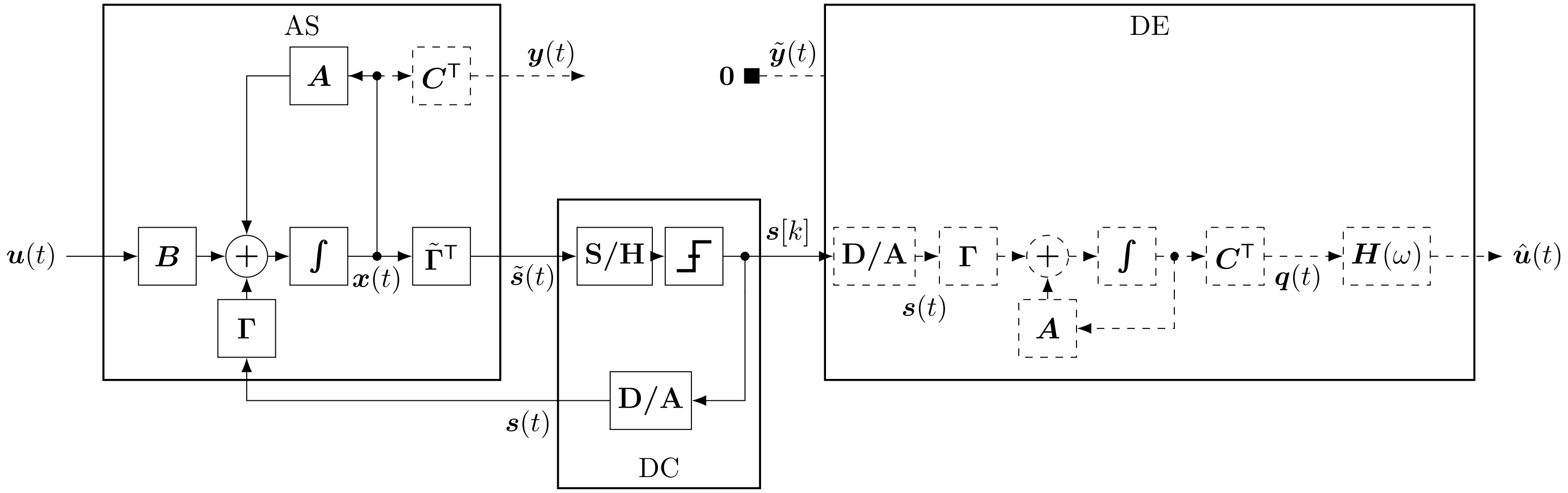

Figure 3.4: Block diagram of the complete control-bounded ADC, with the digital estimation problem visualized. The approximation $\boldsymbol{y}(t) \approx \boldsymbol{0}$ is indicated by the fixed observation of $\tilde{\boldsymbol{y}}(t) = \boldsymbol{0}$ outside the DE box.

system. This conversion error does not enter the estimate directly, but is filtered by $\boldsymbol{h}(t)$. From the Fourier transform of (3.16),

$$\hat{\boldsymbol{U}}(\omega) = \underbrace{\boldsymbol{H}(\omega)\boldsymbol{G}(\omega)}_{\text{STF}}\boldsymbol{U}(\omega) - \underbrace{\boldsymbol{H}(\omega)}_{\text{NTF}}\boldsymbol{Y}(\omega), \tag{3.17}$$

we recognize the noise and signal transfer functions as STF $= \boldsymbol{H}(\omega)\boldsymbol{G}(\omega)$ and NTF $= \boldsymbol{H}(\omega)$ respectively.

The Statistical Estimation Problem

Up to this point, the focus has been to describe the context and the operating principle of the digital estimator. To derive an expression for the actual estimation filter, the problem is first described as a statistical estimation problem. The derivation is carried out in detail in [5] and only the main results is presented in this section.

The error introduces by the approximation in (3.15) is treated by using statistical methods. In the following, both $\boldsymbol{y}(t)$ and $\boldsymbol{u}(t)$ is assumed to be independent, centered, multivariate and wide-sense stationary stochastic processes. The estimation filter is then determined by

$$\begin{aligned} \boldsymbol{h}(t) &= \underset{\bar{\boldsymbol{h}}}{\operatorname{argmin}}\,\mathrm{E}[(\hat{\boldsymbol{u}}(t) - \boldsymbol{u}(t))^2] & (3.18)\\ &= \underset{\bar{\boldsymbol{h}}}{\operatorname{argmin}}\,\mathrm{E}[((\bar{\boldsymbol{h}} * \boldsymbol{q})(t) - \boldsymbol{u}(t))^2]. & (3.19)\end{aligned}$$

This minimization problem is exactly the objective of the Wiener-filter [7] and the impulse response matrix is given by the solution to the well known Wiener-Hopf equations:

$$(\boldsymbol{h} * \boldsymbol{R}_{\boldsymbol{q}\boldsymbol{q}^\mathsf{T}})(\tau) = \boldsymbol{R}_{\boldsymbol{u}\boldsymbol{q}^\mathsf{T}}(-\tau) \tag{3.20}$$

where

$$\begin{aligned} \boldsymbol{R}_{\boldsymbol{q}\boldsymbol{q}^\mathsf{T}} &\triangleq \mathrm{E}[\boldsymbol{q}(t)\boldsymbol{q}(t+\tau)^\mathsf{T}] & (3.21)\\ \boldsymbol{R}_{\boldsymbol{u}\boldsymbol{q}^\mathsf{T}} &\triangleq \mathrm{E}[\boldsymbol{u}(t)\boldsymbol{q}(t+\tau)^\mathsf{T}] & (3.22)\end{aligned}$$

are the autocovariance and cross-covariance matrices respectively. By taking the Fourier transform of (3.20) we obtain the frequency response matrix $\boldsymbol{H}(\omega)$ as

$$\boldsymbol{H}(\omega) = \boldsymbol{G}^\mathsf{H}(\omega)\left(\boldsymbol{G}(\omega)\boldsymbol{G}^\mathsf{H}(\omega) + \eta^2\boldsymbol{I}_N\right)^{-1}, \tag{3.23}$$

and the reader is referred to [5] for computational details. The parameter η is defined as

$$\eta \triangleq \frac{\sigma_{\boldsymbol{y}}^2}{\sigma_{\boldsymbol{u}}^2}, \tag{3.24}$$

where $\sigma_{\boldsymbol{y}}^2$ and $\sigma_{\boldsymbol{u}}^2$ are the power spectral densities of $\boldsymbol{y}(t)$ and $\boldsymbol{u}(t)$ respectively.

3.5.2 Estimation filter implementation

With the digital estimation filter described by (3.23), the estimation could in principle be carried out by computing $\hat{\boldsymbol{u}}(t)$ as in (3.12). This computation is however not straight forward. First of all, the elements of $\boldsymbol{q}(t)$ will necessarily be very large in magnitude, as this was the condition for the approximation (3.15). Carrying out a continuous time convolution with this unbounded signal would obviously lead to numerical problems. In addition the computation of $\boldsymbol{q}(t)$ from $\boldsymbol{s}[k]$, as illustrated in figure 3.4, might be computationally expensive.

In [2] it was shown that the estimate $\hat{\boldsymbol{u}}(t)$ can be computed in an alternative way, using a non-standard version of the Kalman smoothing algorithm. This algorithm converges to the the estimate (3.12) as the considered time window extends towards infinity. The algorithm is also indifferent to the stability assumptions made in the previous section.

As the algorithm is nothing more than an efficient way of computing (3.12), a description of the implementation is not needed for understanding the behaviour of the digital estimator. It is however important for simulations and a concise description of the filter algorithm is provided in appendix B.

3.5.3 Practical Remarks

We conclude this section with some practical considerations.

Controlling the Filter Bandwidth

In (3.24) the parameter η was defined in terms of the power spectral densities of $\boldsymbol{y}(t)$ and $\boldsymbol{u}(t)$, when these signals are modeled as independent stochastic processes. In practice however, η is a free variable and is used by the designer to control the bandwidth of the estimation filter. To see this, consider the scalar input case where both $\boldsymbol{G}(\omega)$ and $\boldsymbol{H}(\omega)$ are column vectors. In this case, the noise transfer function (3.23) reduces to

$$\boldsymbol{H}(\omega) = \text{NTF} = \frac{\boldsymbol{G}^{\mathsf{H}}(\omega)}{||\boldsymbol{G}(\omega)||_2^2 + \eta^2} \in \mathbb{C}^{1\times\tilde{N}}, \tag{3.25}$$

and the signal transfer function becomes

$$\text{STF} = \frac{||\boldsymbol{G}(\omega)||_2^2}{||\boldsymbol{G}(\omega)||_2^2 + \eta^2} \in \mathbb{R}. \tag{3.26}$$

Assuming $||\boldsymbol{G}(\omega)||_\infty$ is monotonically decreasing in ω, the bandwidth of the digital estimator may be defined in terms of the critical frequency, ω_c, as

$$||\boldsymbol{G}(\omega_c)||_2^2 = \eta^2. \tag{3.27}$$

As the parameter η appeared with a precise definition from the optimum filter derivation, it might sound strange that this parameter is now treated as a free variable. This may be understood as follows. In the derivation that lead to (3.24), no assumptions where made on the bandwidth of the input signal $\boldsymbol{u}(t)$. If $||\boldsymbol{G}(\omega)||_\infty$ is monotonically decreasing in ω, then above a certain frequency, the error signal $\boldsymbol{y}(t)$ will become comparable to $\boldsymbol{u}(t)$ in magnitude. Beyond this frequency, $\boldsymbol{q}(t)$ contains more error than information, and the quality of the estimate is improved by reducing the influence of these higher frequency components. Therefore, with no prior knowledge of $\boldsymbol{u}(t)$, the optimum "cut-off" frequency of the estimation filter is given by $||\boldsymbol{G}(\omega_c)||_2 = \sigma_{\boldsymbol{y}}^2/\sigma_{\boldsymbol{u}}^2$.

In a practical application however, we usually know which frequency components of $\boldsymbol{u}(t)$ that contains the sought information. In this case the quality of the estimate would obviously be improved by choosing the cut-off frequency based on this prior knowledge.

Signal-to-Noise Ratio

An analytic derivation of the SNR of the control-bounded ADC is given in [4]. The analysis models the output of the analog system, $\boldsymbol{y}(t)$, as white noise, i.e. assuming the power spectral density is given by $\boldsymbol{S}_{\boldsymbol{y}\boldsymbol{y}^\mathsf{T}}(\omega) \approx \sigma_{\boldsymbol{y}|\mathcal{B}}^2 \boldsymbol{I}_{\tilde{N}}$. In this expression, $\mathcal{B}$ denotes the frequency band of interest and $\sigma_{\boldsymbol{y}|\mathcal{B}}^2$ is the variance of $\boldsymbol{y}(t)$ within this frequency band. From this assumption an approximated expression for the SNR is obtained as

$$\mathrm{SNR} \approx \frac{\sigma_{\boldsymbol{y}|\mathcal{B}}^2}{2\pi} \int_{\omega\in\mathcal{B}} \frac{1}{||\boldsymbol{G}(\omega)||_2^2} d\omega. \tag{3.28}$$

Even though this is an approximation it reveals a useful intuition of how the quantities affect the performance of the ADC. $\sigma_{\boldsymbol{y}|\mathcal{B}}^2$ relates to the magnitude of $\boldsymbol{y}(t)$, and is minimized by tightening the control bound $b_{\boldsymbol{x}}$. $||\boldsymbol{G}(\omega)||_2^2$ is maximized by increasing the gain of the analog system. Therefore, a tight control bound together with a high analog system gain result in large SNR.

The SNR is also related to the bandwidth parameter η, as seen by considering the ratio between the STF and NTF

$$\begin{aligned} \frac{\mathrm{STF}(\omega_c)}{||\boldsymbol{H}(\omega_c)||_2} &= \frac{||\boldsymbol{G}(\omega_c)||_2^2}{||\boldsymbol{G}(\omega_c)||_2^2 + \eta^2} \left(\frac{||\boldsymbol{G}(\omega_c)||_2}{||\boldsymbol{G}(\omega_c)||_2^2 + \eta^2} \right)^{-1} && (3.29) \\ &= ||\boldsymbol{G}(\omega_c)||_2 && (3.30) \\ &= \eta. && (3.31) \end{aligned}$$

Therefore a trade-off has to be made between the bandwidth of the ADC and the suppression of the conversion error. This is similar to the trade-off in a $\Sigma\Delta$ ADC when considering the cut-off frequency of the decimation filter.

This trade-off will be exemplified when particular ADC implementations is considered in the following chapters.

Simulations

When simulating control a bounded ADC one

Chapter 4

The Chain-of-Integrators ADC

The simplest control-bounded ADC is the Chain-of-integrators ADC, as presented in [4]. This configuration consists of an integrator chain, where each integrator is stabilized by a local, independent digital control loop. The structure of this analog system resembles the MASH $\Sigma\Delta$ modulator, and it is shown in [5] that the performance is also very similar. A block diagram of the full chain-of-integrators ADC is shown in figure 4.1.

As highlighted in the previous section, the main contribution of the control-bounded ADC is the design flexibility guaranteed by the digital estimator. Thus, this straight forward approach does not utilize this flexibility and does indeed show some shortcomings compared to more advanced structures, like the Hadamard ADC presented in the next chapter.

However, the chain-of-integrators serves as an important starting point and a thorough understand of its operating principle is important for understanding the structure presented next. The theoretical analysis is also rather straight forward and several important results from the analysis of the chain-of-integrators are directly applicable to other architectures. In addition, the simulations on the chain-of-integrators demonstrates the developed simulation framework which is an essential tool for further work on the topic.

This chapter is organized as follows. The first section presents a description of the analog system together with a transfer function analysis, based on the results from section 3.3.2. The digital control is then briefly described and the conditions for an effective control is discussed. Finally we present and discuss the simulation results.

4.1 Analog System

The analog system of the chain-of-integrators ADC is shown in figure 4.1. The (ideal) integrators has a transfer function β/s, and the parameter β is referred to as the *integrator gain*. The input signal $u(t)$ is passed through N such integrators, each being controlled by a local, independent control loop. Note that the input signal and output estimates of this ADC are both scalars, as a multi-input chain-of-integrators would be nothing more than L equal systems in parallel.

The system dynamics is described by the equation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{CI}\boldsymbol{x}(t) + \boldsymbol{B}_{CI}u(t) + \boldsymbol{\Gamma}_{CI}\boldsymbol{s}(t). \tag{4.1}$$

where the system matrix and the input matrix are given by

$$\boldsymbol{A}_{CI} = \begin{pmatrix} 0 & & & \\ \beta & 0 & & \\ & \ddots & \ddots & \\ & & \beta & 0 \end{pmatrix} \in \mathbb{R}^{N\times N} \tag{4.2}$$

and

$$\boldsymbol{B}_{CI} = \begin{pmatrix} \beta & 0 & \cdots & 0 \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}^{N\times 1} \tag{4.3}$$

respectively.

The N dimensional state-vector $\boldsymbol{x}(t)$ is observed directly by the local digital control, and the control observation matrix is given by

$$\tilde{\boldsymbol{\Gamma}}_{CI}^{\mathsf{T}} = \boldsymbol{I}_N. \tag{4.4}$$

The output of the 1 bit D/A converter is given by

$$s_i(t) = \begin{cases} \kappa, & \text{if } s_i[k] = 1 \\ -\kappa, & \text{if } s_i[k] = 0, \end{cases} \tag{4.5}$$

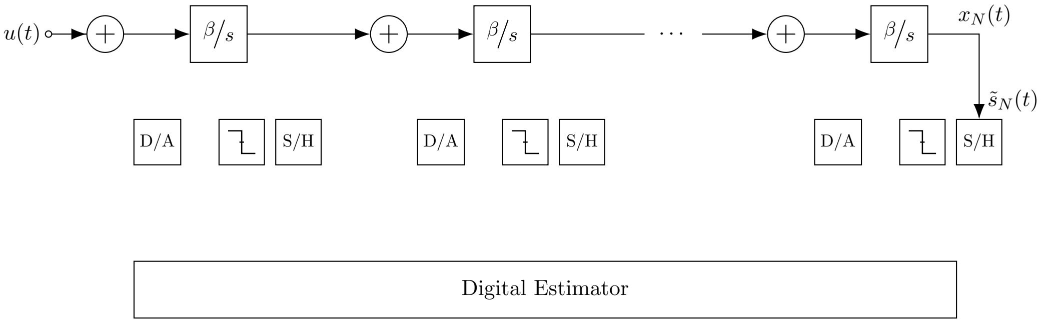

Figure 4.1: A block diagram of an Nth order chain-of-integrators ADC

and κ is referred to as the *control gain.* The control matrix is given by

$$\boldsymbol{\Gamma}_{CI} = \begin{pmatrix} \kappa\beta & & \\ & \ddots & \\ & & \kappa\beta \end{pmatrix}. \tag{4.6}$$

As mentioned in chapter 3, the signal observation matrix $\boldsymbol{C}^\mathsf{T}$ maps the state vector $\boldsymbol{x}(t)$ to the output vector $\boldsymbol{y}(t)$. The functionality of this matrix is basically to tell the estimation algorithm which states that could be treated as bounded. As this matrix is purely conceptual, it has no part in the physical implementation and may be chosen independent of the analog system. Typically one would choose to map either all or only the last state to the output, by choosing either

$$\boldsymbol{C}_{CI_s}^\mathsf{T} = \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{1\times N} \tag{4.7}$$

or

$$\boldsymbol{C}_{CI_m}^\mathsf{T} = \boldsymbol{I}_N. \tag{4.8}$$

These different choices of $\boldsymbol{C}^\mathsf{T}$ is referred to as single and multiple output respectively. Intuitively one would think that considering all internal states of the analog system in the estimation filter would give increased performance, and this is indeed the case. The computational complexity of the filter is also indifferent to the choice of $\boldsymbol{C}^\mathsf{T}$, so $\boldsymbol{C}_{CI_m}^\mathsf{T}$ is the natural choice for this matrix. The single output matrix is still considered in this paper for the sake of a tractable analysis.

4.1.1 Transfer Function Analysis

For this single input ADC, the analog transfer function of 3.7 reduces to a column vector. Each element of the transfer function vector is given by

$$G_k(\omega) = \prod_{\ell=0}^{N-1} \frac{\beta}{j\omega}. \tag{4.9}$$

Hence for the single output case,

$$G_s(\omega) = \left(\frac{\beta}{j\omega}\right)^N \tag{4.10}$$

and

$$||G_s(\omega)||_2^2 = |G_{N-1}(\omega)|^2 = \left(\frac{\beta}{\omega}\right)^{2N}. \tag{4.11}$$

For multiple output,

$$||G_m(\omega)||_2^2 = \frac{1 - \left(\frac{\omega}{\beta}\right)^{2N}}{\left(\frac{\omega}{\beta}\right)^{2N}\left(1 - \frac{\omega^2}{\beta^2}\right)}. \tag{4.12}$$

A comparison of the analog transfer function obtained from single and multiple output is shown in figure 4.2, for $\beta = 2\pi \cdot 20\,\text{MHz}$ and $N = 5$. As the figure shows, the difference is mainly visible for frequencies above the unity gain of the integrators.

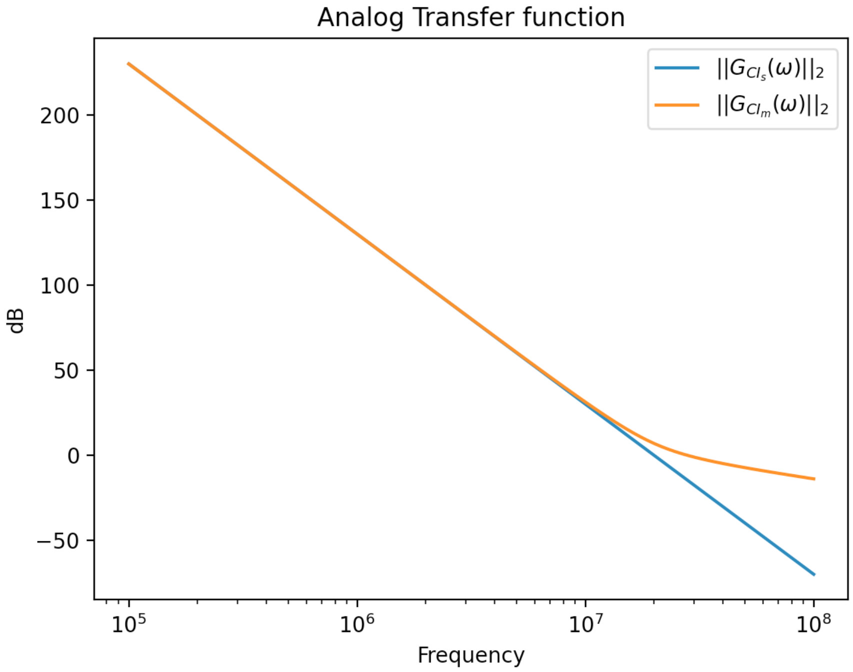

Figure 4.2: Comparison of analog transfer function obtained from single and multiple output, for $\beta = 2\pi \cdot 20\,\text{MHz}$ and $N = 5$.

4.2 Effective digital control

In the chain-of-integrators ADC, each integrator is under local digital control. This simple control structure is easy to implement and allows for an analytic derivation of the effective control criteria. This derivation is carried out in detail in [5] and the main results is presented here.

In the analysis, the input is assumed to be bounded, i.e. $|u(t)| \leq b_u \forall t$, and the conditions for effective control ensure $||\boldsymbol{x}(t)||_\infty \leq b_{\boldsymbol{x}} \forall t$, see section 3.4. For the chain-of-integrators this is guaranteed If

$$|\kappa| \geq b_{\boldsymbol{x}} \tag{4.13}$$

and

$$T|\beta|(|\kappa| + b_{\boldsymbol{x}}) \leq b_{\boldsymbol{x}}. \tag{4.14}$$

4.2.1 Implications on Sampling Rate

A natural choice of κ and $b_{\boldsymbol{x}}$ is to let both equal the positive supply voltage. With $\kappa = b_{\boldsymbol{x}}$, (4.14) reduces to

$$T|\beta| \leq \frac{1}{2}. \tag{4.15}$$

To see how this condition influences the sampling rate, let $f_s = \frac{1}{T}$ be the sampling frequency of the digital control and let $f_u = \frac{\beta}{2\pi}$ be the unity gain frequency of the integrators. Equation (4.15) may then be written as

$$f_s \geq 4\pi f_u, \tag{4.16}$$

i.e. given $\kappa = b_{\boldsymbol{x}}$, the sampling rate must be approximately 12.6 times the unity gain frequency of the integrators in order to guarantee an effective control.

To place this requirement in the context of oversampling, the unity gain frequency must be related to the frequency band of interest. For simplicity, only the single output transfer function is now considered. Assume that the frequency content of the input signal is upper bounded by a frequency f_0 and the critical frequency of the estimation filter is set as $f_c \geq f_0$. Then from (3.27) and (4.11), the parameter η may be expressed as

$$\eta = \left(\frac{\beta}{\omega_c}\right)^N = \left(\frac{f_u}{f_c}\right)^N. \tag{4.17}$$

From (3.29)-(3.31), η is also the relation between the magnitude of the signal and noise transfer function at the critical frequency. From (4.17) it is seen that η grows with system order as long as $\frac{f_u}{f_c} \geq 1$. We therefore define a practical limit

$$f_u \geq 2 f_c \tag{4.18}$$

for this relation, in order to take advantage from the system order.

Based on this discussion the relation between the signal bandwidth f_0 and the sampling rate may ultimately be expressed as

$$f_s \geq 8\pi f_0, \tag{4.19}$$

or equivalently

$$\text{OSR} \geq 4\pi. \tag{4.20}$$

Thus, for a chain-of-integrators ADC with guaranteed stability and $\kappa = b_{\boldsymbol{x}}$, the minimum oversampling rate is approximately 12.6.

As mentioned in section 3.4, designing for stability guarantee is not attractive from a performance point of view. In section 3.5.3 it was shown that T, β and $b_{\boldsymbol{x}}$ are the only parameters affecting the SNR of a control-bounded ADC and it is therefore an inevitable trade-off between stability and performance. The preferred way of determining this trade-off is by simulations, and the results obtained in this section serves as a useful starting point.

4.3 Simulations

The process of simulating a control-bounded ADC is divided into two separate steps. First, the interaction between the analog system and the digital control is simulated in time domain to generate the control signal $\boldsymbol{s}[k]$. In the following, this part of the simulation process is referred to as the *system simulation.* These control signals are then applied as input to the digital estimation filter, which generates the output estimate $\hat{\boldsymbol{u}}(t)$. For the chain-of-integrators ADC, the system simulations are done using the Spectre simulation platform [8] and the estimation filter is implemented in python.

4.3.1 System Simulation

The circuit used for the system simulation is a 5th order system, derived from the diagram of figure 4.1. The integrators are implemented using a first order opamp-RC filter as shown in figure 4.3, and the opamp is modelled by an ideal voltage controlled voltage source. The Sample-and-hold, 1 bit quantizer and 1 bit dac, are all incorporated in an ideal, clocked comparator written in VerilogA. The digital control signal $\boldsymbol{s}[k]$ does therefore not exist in the system simulation and is extracted from the control contribution, $\boldsymbol{s}(t)$, in a post-processing step.

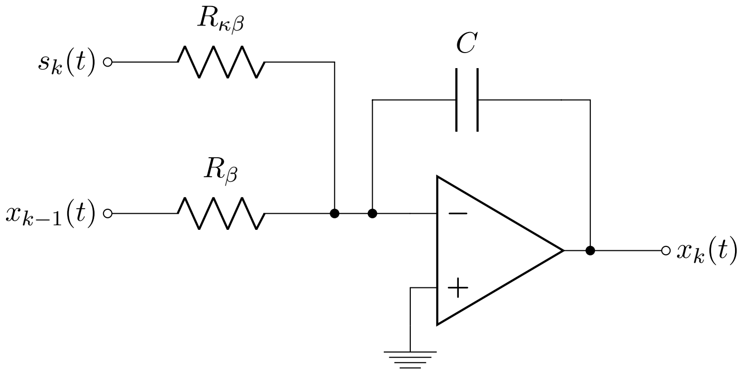

Figure 4.3: An Opamp-RC integrator used for the chain-of-integrators simulation.

4.3.2 Estimation Filter Implementation

The digital estimation filter is implemented in python and the source code is available at GitHub [9]. The filter is first initialized by computing the offline matrices, defined by equations (B.4)-(B.10). The actual estimate is than carried out recursively as described by (B.1)-(B.3). As the recursive estimate takes place after the system simulation is complete, both the forward and the backward recursion uses all available control signals. In a production environment, this is obviously not possible and only a limited amount of "future" samples are available. The filter algorithm is quite sensitive to the boundary conditions, but usually a few hundred samples is enough for the filter to settle properly.

4.3.3 Simulation Results

The parameters used for the simulation is summarized in table 4.1. The integrator gain is achieved by choosing $R_\beta = 320\,\mathrm{k\Omega}$ and $C = 10\,\mathrm{fF}$. The comparator output is $\pm 1\,\mathrm{V}$, and the control gain is set to $\kappa = 1$ by having $R_{\kappa\beta} = R_\beta$.

Table 4.1: Simulation parameters, Chain-of-Integrators

Parameter	Symbol	Value
Critical frequency	f_c	10 MHz
Unity gain frequency	f_u	50 MHz
Integrator gain	β	315 MHz
Sampling frequency	f_s	650 MHz
Oversampling ratio	OSR $\left(\frac{f_s}{2f_c}\right)$	32.5
Control gain	κ	1

A plot of the estimated power spectral density is provided in figure 4.4, for both the single and multiple output case. Note that the difference is mainly notable for frequencies above the critical frequency of the estimation filter.

A snapshot of the time-domain simulation is shown in figure 4.5, which compares the state signals $x_1(t)$ and $x_5(t)$ for a full scale sinusoidal input. Note that the input signal is clearly present in $x_1(t)$, while it is almost invisible in $x_5(t)$.

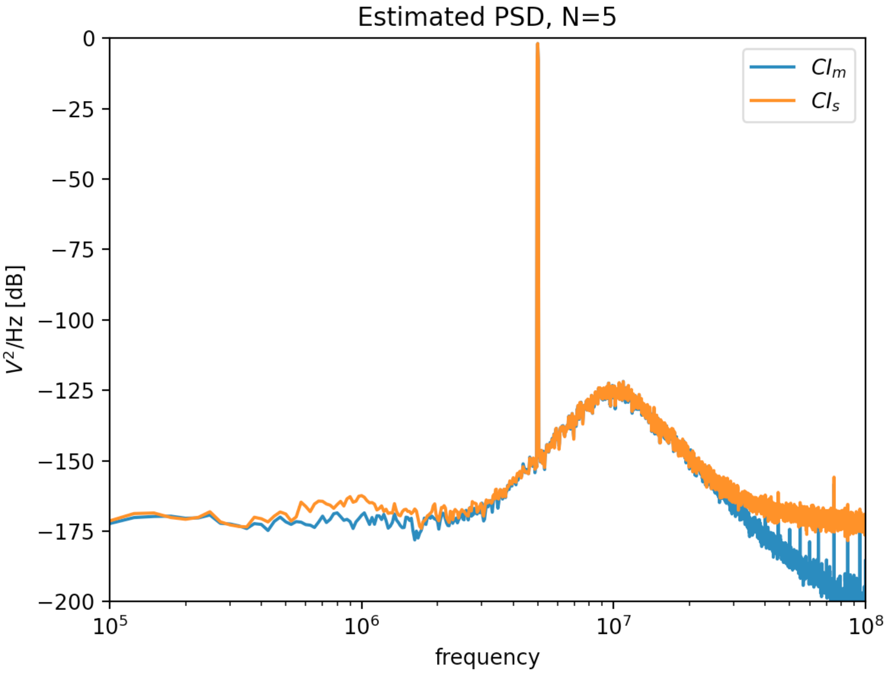

Figure 4.4: Estimated PSD for a 5th order chain-of-integrator ADC.

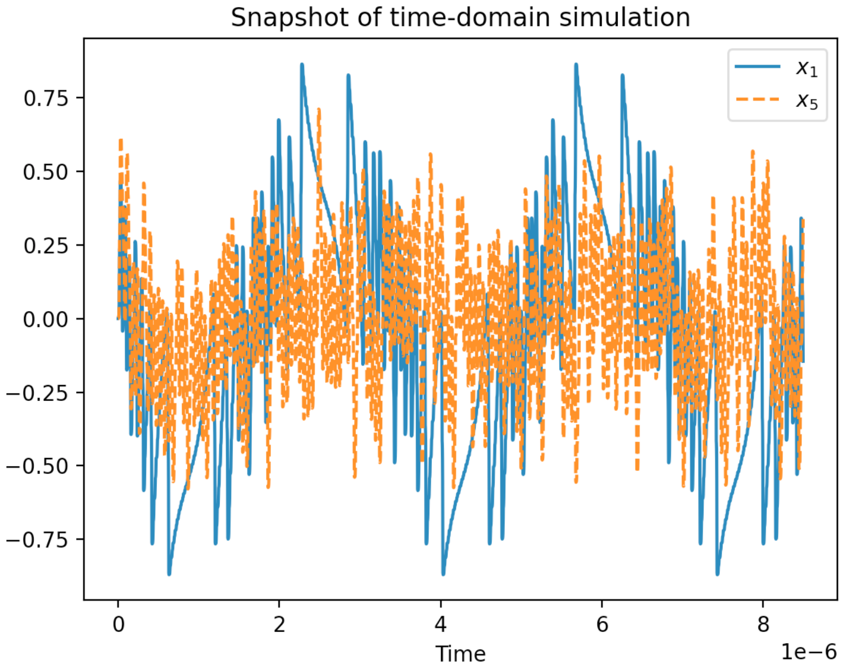

Figure 4.5: Time-domain snapshot of $x_1(t)$ and $x_5(t)$, for a full scale sinusoidal input.

Chapter 5

Proposed ADC Architecture

The chain-of-integrator ADC presented in the previous chapter provides an intuitive introduction to the control-bounded ADC, with an architecture not very different from the well known MASH $\Sigma\Delta$ modulator. As highlighted in chapter 3, the main advantage of the control-bounded ADC is the design freedom guaranteed by the digital estimator, which imposes no restrictions on the analog system and the digital control.

As an attempt to take advantage of this design freedom, the Hadamard ADC was proposed by Malmberg in [5] and [6]. The main objective of the Hadamard ADC is to distribute the mismatch sensitivity equally among all involved circuit components. In addition, the Hadamard ADC enables a beneficial way of combining multiple input channels. The properties of the Hadamard ADC will be further explored later in this chapter.

The key idea of the Hadamard ADC is to separate the logical states from the physical ones. This is done by applying a Hadamard transform to the input signal vector before the integration. This way, the *physical* states entering each integrator contains an orthogonal mixture of each of the input channels, i.e. the state space is rotated.

One way of realizing this Hadamard system is presented in [5] and [6]. This architecture has good mismatch properties and performs very well in combination with overcomplete control and multiple input channels. A disadvantage of this architecture is however the use of buffers in the analog system. These are active components that will consume a considerable amount of power, without contributing with any gain to the overall system.

In this chapter, an alternative hardware implementation is presented. The proposed architecture has no buffers, thereby providing more gain for the same number of active components. The architecture also show the same tolerance to component mismatch as the original architecture. This novel architecture is presented together with a theoretical transfer function analysis and simulation results.

5.1 Analog System

5.1.1 Parametrization

The Hadamard transform is obtained by performing an analog multiplication of the input vector with a Hadamard matrix. For N being powers of two, the Hadamard matrix is defined recursively as

$$\boldsymbol{H}_N = \boldsymbol{H}_2 \otimes \boldsymbol{H}_{N/2} \tag{5.1}$$

where

$$\boldsymbol{H}_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{5.2}$$

The Hadamard matrix is an orthogonal matrix with the useful properties

$$\boldsymbol{H}_N = \boldsymbol{H}_N^{\mathsf{T}} \tag{5.3}$$

and

$$\boldsymbol{H}_N^{\mathsf{T}} \boldsymbol{H}_N = N \boldsymbol{I}_N. \tag{5.4}$$

The Hadamard ADC is described by the equations

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_H \boldsymbol{x}(t) + \boldsymbol{B}_H \boldsymbol{u}(t) + \boldsymbol{\Gamma}_H \boldsymbol{s}(t) \tag{5.5}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}_H^{\mathsf{T}} \boldsymbol{x}(t) \tag{5.6}$$

and

$$\tilde{\boldsymbol{s}}(t) = \tilde{\boldsymbol{\Gamma}}_H^{\mathsf{T}} \boldsymbol{x}(t). \tag{5.7}$$

The $\boldsymbol{\Gamma}$-matrices are covered together with the digital control in the next section. The proposed architecture is derived from the parametrization introduced in [5], where

$$\boldsymbol{A}_H = \frac{1}{N} \boldsymbol{H}_N \boldsymbol{A}_{CI} \boldsymbol{H}_N^{\mathsf{T}}, \tag{5.8}$$

$$\boldsymbol{B}_H = \frac{1}{N} \boldsymbol{H}_N \boldsymbol{B}_{CI} \tag{5.9}$$

and

$$\boldsymbol{C}_H^{\mathsf{T}} = \boldsymbol{C}_{CI}^{\mathsf{T}} \boldsymbol{H}_N^{\mathsf{T}} \tag{5.10}$$

where $\boldsymbol{A}_{CI}, \boldsymbol{B}_{CI}$ and $\boldsymbol{C}_{CI}^{\mathsf{T}}$ refers to the chain-of-integrator matrices defined in equation (4.2)-(4.8). As mentioned in the introduction, the motivation for the proposed architecture is to use all active components for integration. By simply replacing the buffers with integrators, a different parametrization is obtained, with a very similar transfer function and performance characteristics.

For the single input case, the proposed system is described by

$$\boldsymbol{A}_H = \boldsymbol{H}'_N \boldsymbol{A}' \in \mathbb{R}^{N \times N}, \tag{5.11}$$

$$\boldsymbol{B}_H = \boldsymbol{H}'_N \boldsymbol{B}_{CI} \in \mathbb{R}^{N \times 1} \tag{5.12}$$

and

$$\boldsymbol{C}_H = \boldsymbol{C}_{CI_s} \in \mathbb{R}^{N \times 1} \tag{5.13}$$

or

$$\boldsymbol{C}_H = \boldsymbol{C}_{CI_m} \in \mathbb{R}^{N \times N}, \tag{5.14}$$

where $\boldsymbol{C}_{CI_s}$ and $\boldsymbol{C}_{CI_m}$ yields the single and multiple output case respectively. Furthermore,

$$\boldsymbol{H}'_N = \begin{bmatrix} \boldsymbol{H}_{N/2} & \boldsymbol{0}_{N/2} \\ \boldsymbol{0}_{N/2} & \boldsymbol{H}_{N/2} \end{bmatrix} \in \mathbb{R}^{N \times N} \tag{5.15}$$

and

$$\boldsymbol{A}' = \begin{bmatrix} \boldsymbol{0}_{N/2} & \beta \boldsymbol{L}_{N/2} \\ \beta \boldsymbol{I}_{N/2} & \boldsymbol{0}_{N/2} \end{bmatrix} \in \mathbb{R}^{N \times N}. \tag{5.16}$$

The matrix $\boldsymbol{A}'$ is described as a block matrix and the sub-matrix $\boldsymbol{A}'_{21} = \beta \boldsymbol{L}_{N/2}$ is a strictly lower triangular matrix, given by

$$\boldsymbol{L}_{N/2} = \begin{pmatrix} 0 & & & & \\ 1 & 0 & & & \\ 0 & 1 & 0 & & \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix} \in \mathbb{R}^{\frac{N}{2} \times \frac{N}{2}}, \tag{5.17}$$

As for the chain-of-integrators, both the single and the multiple output case are possible when choosing the signal observation matrix, $\boldsymbol{C}_H^\mathsf{T}$. However, for the sake of a tractable analysis, only the single output ($\boldsymbol{C}_H^\mathsf{T} = \boldsymbol{C}_{CI_s}^\mathsf{T}$) is considered in this chapter. Note that the term single output refers to *output per channel*, i.e. for an ADC with L inputs, a single output configuration will give L outputs to the estimation filter.

For the multiple input case, i.e. $L > 1$, we define $N = LN_\ell$, where N_ℓ is the order of the single input system. Due to the shape of the state-space matrices, we restrict both N_ℓ and L to be powers of 2. For $L > 1$, the state-space matrices generalizes as

$$\boldsymbol{L}_{N/2} = \begin{bmatrix} \boldsymbol{L}_{N_\ell/2} & & \\ & \ddots & \\ & & \boldsymbol{L}_{N_\ell/2} \end{bmatrix} \in \mathbb{R}^{\frac{N}{2} \times \frac{N}{2}}, \tag{5.18}$$

$$\boldsymbol{B}_H = \begin{bmatrix} \boldsymbol{B}_{H_\ell} & & \\ & \ddots & \\ & & \boldsymbol{B}_{H_\ell} \end{bmatrix} \in \mathbb{R}^{N \times L} \tag{5.19}$$

and

$$\boldsymbol{C}_H = \begin{bmatrix} \boldsymbol{C}_{H_\ell} & & \\ & \ddots & \\ & & \boldsymbol{C}_{H_\ell} \end{bmatrix} \in \mathbb{R}^{N \times L}, \tag{5.20}$$

where subscript ℓ refers to the single input equivalent, defined by (5.12) and (5.14). With $N = LN_\ell$ and $\boldsymbol{L}_{N/2}$ as above, $\boldsymbol{A}_H$ is still given by (5.11).

In this single output case, the transfer function is a column vector given by

$$\boldsymbol{G}(\omega) = \boldsymbol{C}_H^\top \left(j\omega \boldsymbol{I}_N - \boldsymbol{A}_H \right)^{-1} \boldsymbol{B}_H \tag{5.21}$$

where each element gives the transfer function of the corresponding input. It is shown in appendix C that all inputs will experience the same transfer function, given by

$$G(\omega) = \left(\sqrt{\frac{N}{2}} \frac{\beta}{j\omega} \right)^{N_\ell}. \tag{5.22}$$

Note in particular that the gain of the analog system is scaled by a factor $\sqrt{\frac{N}{2}} = \sqrt{\frac{LN_\ell}{2}}$. This implies that the gain β provided by each integrator should be reduced by a factor $\sqrt{\frac{N}{2}}$ in order to achieve the same analog system gain. This will ease the requirements on the involved amplifiers and enable a lower current consumption.

5.1.2 Proposed Hardware Implementation

The proposed hardware implementation of this system is shown in figure 5.1 with $N = 8$ and $L = 2$. In this figure, the integrators are abstracted out, and the choice of integrator topology is discussed subsequently. For this example, $\boldsymbol{A}_H$ is given by

$$\boldsymbol{A}_H = \boldsymbol{H}_8' \boldsymbol{A}' \tag{5.23}$$

$$= \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 \\ \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{5.24}$$

In figure 5.1 color coding is used to distinguish between the inputs (green), the analog states (orange) and the control contributions (red).

The boxes labeled $\boldsymbol{H}_4(Z)$ contains the analog implementation of a 4-by-4 Hadamard matrix. The differential implementation enables an efficient realization of the ±1 operation of the Hadamard matrix, by crossing/not crossing the wires. An implementation of the $H_4(Z)$ matrix is shown in figure 5.2, where the impedance Z will be resistive or capacitive depending on the integrator topology.

The two inputs u_0 and u_1 is connected to the first Hadamard matrix together with the states x_4 and x_6. The inputs to the first column of integrators will therefore be an orthogonal mixture of each of these signals. The outputs of these integrators are denoted x_0-x_3. Because of the state-space rotation achieved by $\boldsymbol{H}_4(Z)$, these physical states are separated from the logical ones. The control contributions s_0-s_3 is applied at the input to these integrators to bound their outputs.

These four states are then applied as an input to the second Hadamard matrix. As the Hadamard matrix is its own inverse up tp a constant, cf. (5.3) and (5.4), this second Hadamard matrix rotates the state-space back to the origin. In consequence, the states x_4-x_7 lies in the "normal" state-space, and the logical states coincides with the physical ones at this point. This means that e.g. x_4 is the result of u_0 passing through two integrators, each being controlled by a digital feedback loop. To increase the system order, x_4 is fed back to the first Hadamard matrix, and the relation between x_5 and x_4 is the same as the relation between x_4 and u_0. The same holds for u_1, x_6 and x_7.

The system may therefore be understood as 4th order, multiple-input chain-of-integrators, where all inputs are distributed over the first column of integrators. As the Hadamard matrix is only defined for N being powers of two, both the number of inputs, L, and the system order per input, N_ℓ, is also restricted to being powers of two. Increasing N_ℓ from 4 to 6 would require a $\boldsymbol{H}_6$ matrix which is undefined. Increasing the number of inputs to e.g. $L = 3$ could however be achieved by leaving the 4th input shortened.

5.1.3 Choice of integrators

For the implementation of the fully differential integrators, both Gm-C and opamp-RC integrators are considered. A schematic of both topologies are shown in figure 5.4 and 5.3 respectively. The Hadamard ADC is only simulated by solving the state-space equations (5.5) directly using an ODE solver library in python, and the system is not simulated on circuit level. A discussion of the different integrator topologies is still included to provide a starting point for future work.

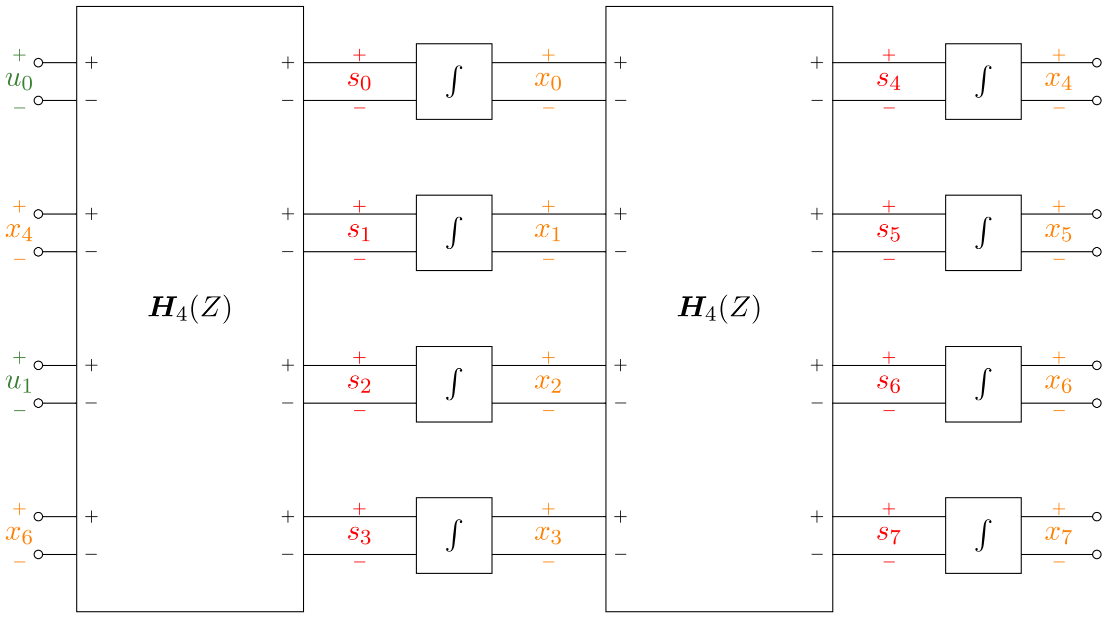

Figure 5.1: Proposed hardware implementation of the Hadamard ADC AS for $N = 8, L = 2$

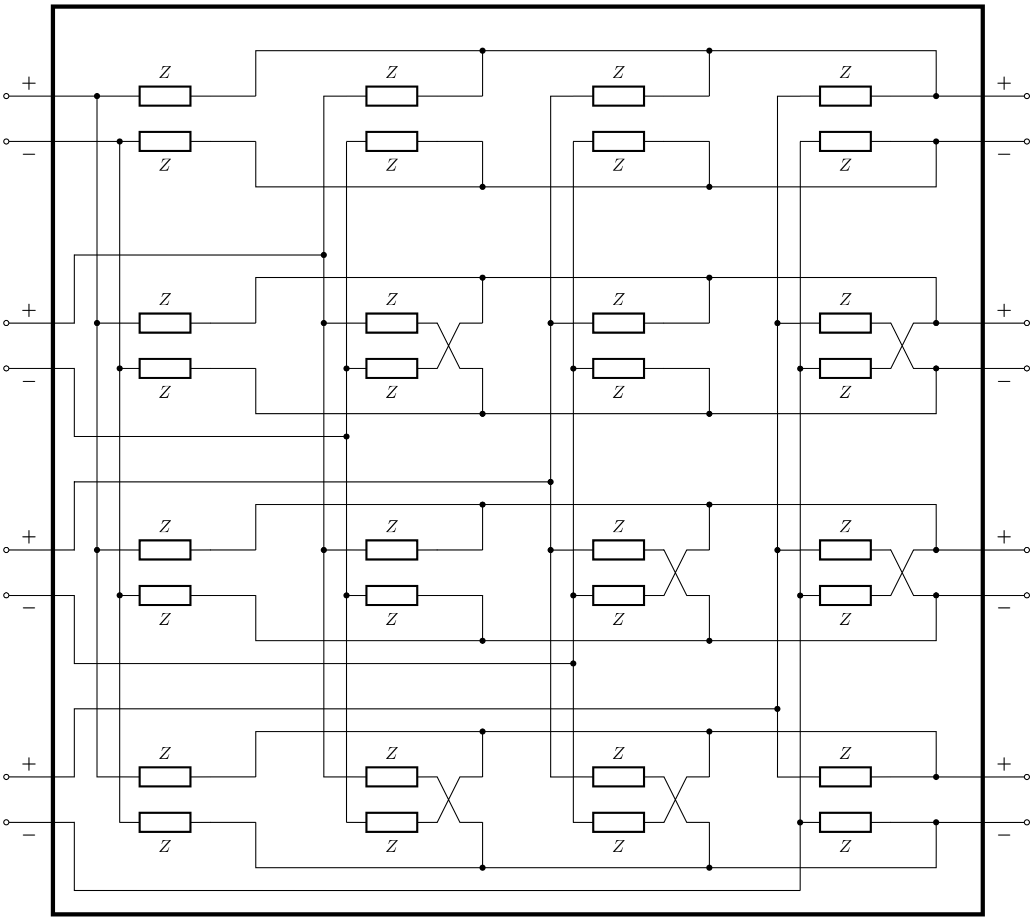

Figure 5.2: A 4th order Hadamard matrix implemented with impedance Z. Straight wires correspond to a multiplication of 1 and crossing wires to multiplication of -1

The Opamp-RC Integrator

The fully differential opamp-RC integrator shown in figure 5.3 integrates the inputs through an operational amplifier with capacitive feedback. This topology requires the Hadamard matrices to be implemented with resistors. The Hadamard matrices takes a voltage signal as input and deliver a current signal at the output. Referring to figure 5.2, each element of the matrix will perform a voltage to current conversion, with a potential sign change. The output currents from each element of the matrix is then summed at the integrators virtual ground.

Referring to figure 5.3, the integrator gain is recognized as

$$\beta = \frac{1}{R_\beta C}. \tag{5.25}$$

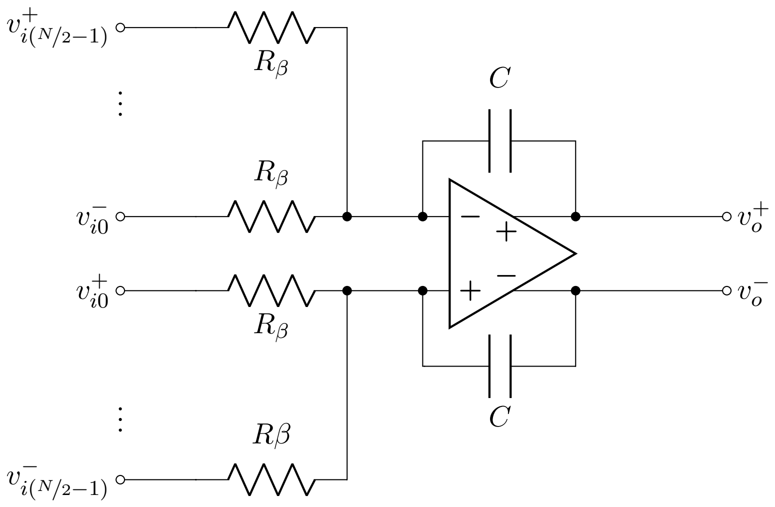

Figure 5.3: A fully differential, opamp-RC integrator with current summation at the input.

The Gm-C Integrator

The fully differential Gm-C integrator is shown in figure 5.4. This integrator is based on an open loop, operational transconductance amplifier (OTA). There is no virtual ground at the input, and hence no current summation is possible at this node. The proposed solution is to do voltage summation through a floating gate configuration. Hence the Gm-C integrator requires a capacitive implementation of the Hadamard matrices. In this case, the Hadamard matrices operates with voltage signals at both the input and the output.

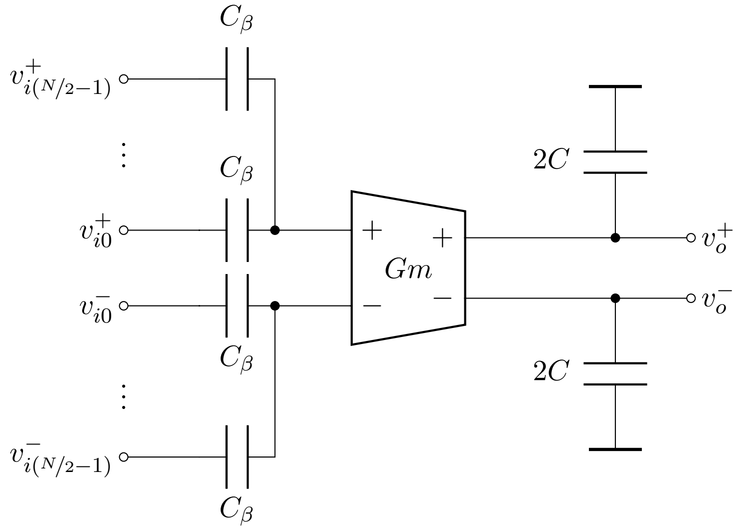

Figure 5.4: A fully differential, floating gate, Gm-C integrator with voltage summation on the input.

The transfer function analysis for this integrator is slightly more involved, as the additive operation is performed by a capacitive voltage division at the input of the OTA. For simplicity, the transfer function is analysed for the single ended equivalent of figure 5.4 and the result is extended to the fully differential case subsequently.

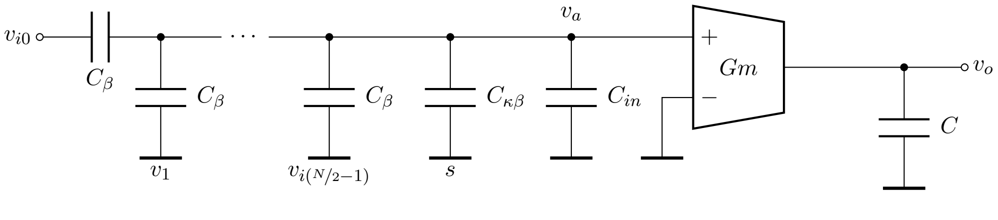

Figure 5.5: A single ended equivalent of the fully differential Gm-C integrator of figure 5.4 used for transfer function analysis.

The single ended equivalent of figure 5.4 is shown in figure 5.5. The capacitors C_β is part of the Hadamard matrix and $C_{\kappa\beta}$ is connected to the digital control. The capacitor C_{in} is modelling the input capacitance of the OTA. In the following analysis, the transfer function from v_{i0} to v_o if found using the superposition principle, and both the control signal s and the other input signals are grounded. Furthermore, the input v_{i0} is assumed to behave as an ideal voltage source. From these assumptions, the OTA input, v_a, is given by

$$v_a = v_{i0}\left(\frac{C_\beta}{\frac{N}{2}C_\beta + C_{\kappa\beta} + C_{in}}\right). \tag{5.26}$$

Referring to figure 5.1, most of the integrators will be connected to another Hadamard matrix. Hence, the output voltage v_o will depend on the total capacitance on this node. To make the integrator gain independent of the Hadamard matrix, we choose $C \gg C_\beta$ and assume that the total capacitance on the output node is approximately equal to C. From this assumption, the integrator gain is recognized as

$$\beta = \frac{Gm}{C}\frac{C_\beta}{\frac{N}{2}C_\beta + C_{\kappa\beta} + C_{in}}. \tag{5.27}$$

Discussion of the Different Integrator Realizations

The two mentioned integrator realizations has very different properties and challenges. As neither of them are evaluated in a circuit simulation, the following discussion is based general knowledge about these circuits, and is intended to provide a background for future work.

The major disadvantage of the opamp-RC integrator is the need for resistors in the Hadamard matrices. The need to drive resistive loads, means that the amplifiers most likely will need two stages, and therefore consume more bias current. In addition, when determining the resistor values, there will be an unpleasant trade-off between current consumption and area.

The accuracy of the integrator gain is depending directly on the values of the involved resistors and capacitors, which is difficult to control accurately in integrated CMOS processes. The linearity will however be good, due to to the negative feedback.

For the Gm-C integrator the situation is the opposite. The OTAs are only driving capacitive loads, and it should be possible to use a power efficient, one-stage architecture for the realization. The necessary transconductance, Gm, depends on the capacitive loading seen by the OTA. The load capacitance, is only limited by the assumption $C \gg C_\beta$ and could be made quite small. Hence the OTAs used in the Gm-C integrators could presumably be designed for a much lower current consumption than the opamps of the opamp-RC realization.

The accuracy of the Gm-C integrator depends on the matching of the involved capacitance, as well as the Gm of the OTA. The former could be determined to about 10 bit accuracy in modern CMOS processes without trimming, which is a lot better than what is achievable for the RC-time constant in the opamp-RC integrators.

A weak point of the Gm-C integrator is the Gm which need to be sufficiently accurate and linear, not to degrade the overall system performance. As the OTA operates in an open-loop configuration, there is no feedback available to “hide” the internal characteristics of the OTA. Different techniques for tuning and linearizing open-loop OTAs are cover in literature (e.g. [1]), and the achievable performance will probably depend on the technology.

Another challenge of the Gm-C integrator comes from the floating gate operation. With the considerable gate leakage in modern CMOS processes, there will be necessary to periodically reset the input node of the OTAs. The associated design challenges as well as the impact on the overall system performance needs further investigation.

In summary, from a system architecture point of view, the Gm-C realization seems the most promising in terms of power consumption and area, which are the main limitations for this application. Achieving the required performance for the OTAs will be a design challenge, but the potential power consumption is lower than for the opamp-RC alternative.

5.1.4 Digital Control

The previous section was concerned with the matrices $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}^{\mathsf{T}}$, but $\boldsymbol{\Gamma}$ and $\tilde{\boldsymbol{\Gamma}}$ still need to be determined. There are an infinite number of possible choices for these matrices.

Appendix A

General Transfer Function of the Analog System

In this appendix presents the derivation of the general transfer function of the analog system. From (3.1) the frequency domain relation between the input and the state vector is obtained as

$$\begin{aligned} j\omega \boldsymbol{X}(\omega) &= \boldsymbol{A}\boldsymbol{X}(\omega) + \boldsymbol{B}\boldsymbol{U}(\omega) && \text{(A.1)} \\ \left(j\omega \boldsymbol{I}_N - \boldsymbol{A}\right)\boldsymbol{X}(\omega) &= \boldsymbol{B}\boldsymbol{U}(\omega) && \text{(A.2)} \\ \boldsymbol{X}(\omega) &= \left(j\omega \boldsymbol{I}_N - \boldsymbol{A}\right)^{-1}\boldsymbol{B}\boldsymbol{U}(\omega). && \text{(A.3)} \end{aligned}$$

The output vector is obtained by multiplying the state vector with the signal signal observation matrix, $\boldsymbol{C}^\mathsf{T}$. Hence

$$\boldsymbol{Y}(j\omega) = \boldsymbol{C}^\mathsf{T}\left(j\omega \boldsymbol{I}_N - \boldsymbol{A}\right)^{-1}\boldsymbol{B}\boldsymbol{U}(j\omega) \tag{A.4}$$

and we recognize the ATF as

$$\boldsymbol{G}(\omega) = \boldsymbol{C}^\mathsf{T}\left(j\omega \boldsymbol{I}_N - \boldsymbol{A}\right)^{-1}\boldsymbol{B}. \tag{A.5}$$

Appendix B

Description of the Estimation Filter Algorithm

This appendix provides a short and concise description of the estimation filter algorithm.

The algorithm consist of a forward recursion

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

a backward recursion

$$\overleftarrow{\boldsymbol{m}}_{k-1} \triangleq \boldsymbol{A}_b \overleftarrow{\boldsymbol{m}}_k + \boldsymbol{B}_b \boldsymbol{s}[k-1], \tag{B.2}$$

and finally the estimate

$$\hat{\boldsymbol{u}}(t_k) \triangleq \boldsymbol{W}^{\mathsf{T}}(\overleftarrow{\boldsymbol{m}}_k - \overrightarrow{\boldsymbol{m}}_k). \tag{B.3}$$

The matrices $\boldsymbol{A}_f, \boldsymbol{A}_b, \boldsymbol{B}_f, \boldsymbol{B}_b$ and $\boldsymbol{W}$ is computed offline, and is given by the following equations.

$$\boldsymbol{A}_f \triangleq \exp\left(\left(\boldsymbol{A} - \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}\right) T\right) \tag{B.4}$$

$$\boldsymbol{A}_b \triangleq \exp\left(-\left(\boldsymbol{A} + \frac{1}{\eta^2}\overleftarrow{\boldsymbol{V}}\right) T\right) \tag{B.5}$$

$$\boldsymbol{B}_f \triangleq \int_0^T \exp\left(\left(\boldsymbol{A} - \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}\right)(T-\tau)\right) \boldsymbol{\Gamma} d\tau \tag{B.6}$$

$$\boldsymbol{B}_b \triangleq -\int_0^T \exp\left(-\left(\boldsymbol{A} + \frac{1}{\eta^2}\overleftarrow{\boldsymbol{V}}\right)(T-\tau)\right) \boldsymbol{\Gamma} d\tau \tag{B.7}$$

In equations (B.4 - B.7), exp(.) denotes the matrix exponential, which is not to be confused with the element-wise exponential operation.

The matrices $\overrightarrow{\boldsymbol{V}}$ and $\overleftarrow{\boldsymbol{V}}$ used in (B.4 - B.7) is obtained by solving the continuous-time algebraic Riccati (CARE) equations

$$\boldsymbol{A}\overrightarrow{\boldsymbol{V}} + \left(\boldsymbol{A}\overrightarrow{\boldsymbol{V}}\right)^{\mathsf{T}} + \boldsymbol{B}\boldsymbol{B}^{\mathsf{T}} - \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}\overrightarrow{\boldsymbol{V}} = \mathbf{0}_{N\times N} \tag{B.8}$$

and

$$\boldsymbol{A}\overleftarrow{\boldsymbol{V}} + \left(\boldsymbol{A}\overleftarrow{\boldsymbol{V}}\right)^{\mathsf{T}} - \boldsymbol{B}\boldsymbol{B}^{\mathsf{T}} + \frac{1}{\eta^2}\overleftarrow{\boldsymbol{V}}\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}\overleftarrow{\boldsymbol{V}} = \mathbf{0}_{N\times N}. \tag{B.9}$$

The matrix $\boldsymbol{W}$ is finally obtained by solving the linear equation system

$$\left(\overrightarrow{\boldsymbol{V}} + \overleftarrow{\boldsymbol{V}}\right)\boldsymbol{W} = \boldsymbol{B}. \tag{B.10}$$

Appendix C

Transfer Function Analysis

This appendix contains the derivation of the transfer function of the AS described by equations 5.11, 5.12 and **??**. It will become apparent that for multiple inputs, i.e. $L > 1$, the transfer function decouples into a set of identical expressions. For this analysis, we first consider the case of single input and then show how the obtained results generalizes for the multiple input case. For a single input, the transfer function is a scalar given by

$$G(\omega) = \boldsymbol{C}^{\mathsf{T}} \left(j\omega \boldsymbol{I}_N - \boldsymbol{A}\right)^{-1} \boldsymbol{B} \tag{C.1}$$

$$= \boldsymbol{C}^{\mathsf{T}} \left(j\omega \boldsymbol{I}_N - \boldsymbol{H}_N' \boldsymbol{A}'\right)^{-1} \boldsymbol{H}_N' \boldsymbol{B}' \tag{C.2}$$

$$= \boldsymbol{C}^{\mathsf{T}} \boldsymbol{M}^{-1} \boldsymbol{H}_N' \boldsymbol{B}', \tag{C.3}$$

where we have defined $\boldsymbol{M} \triangleq (j\omega \boldsymbol{I}_N - \boldsymbol{H}_N' \boldsymbol{A}') \in \mathbb{R}^{N \times N}$. To obtain an analytic expression for the transfer function, we first need a closed form expression for the inverse of this matrix. Referring to equations 5.15 and 5.16, the matrix $\boldsymbol{M}$ can be expressed in block form as

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{21} \\ \boldsymbol{M}_{12} & \boldsymbol{M}_{22} \end{bmatrix} = \begin{bmatrix} j\omega \boldsymbol{I}_{N/2} & -\beta \boldsymbol{H}_{N/2} \boldsymbol{L}_{N/2} \\ -\beta \boldsymbol{H}_{N/2} & j\omega \boldsymbol{I}_{N/2} \end{bmatrix}. \tag{C.4}$$

The general inverse of a block matrix can be expressed using

$$\boldsymbol{D}_1 = \boldsymbol{M}_{11} - \boldsymbol{M}_{12} \boldsymbol{M}_{22}^{-1} \boldsymbol{M}_{21} \tag{C.5}$$

and

$$\boldsymbol{D}_2 = \boldsymbol{M}_{22} - \boldsymbol{M}_{21} \boldsymbol{M}_{11}^{-1} \boldsymbol{M}_{12} \tag{C.6}$$

as

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{D}_1^{-1} & -\boldsymbol{M}_{11}^{-1} \boldsymbol{M}_{12} \boldsymbol{D}_2^{-1} \\ -\boldsymbol{D}_2^{-1} \boldsymbol{M}_{21} \boldsymbol{M}_{11}^{-1} & \boldsymbol{D}_2^{-1} \end{bmatrix}. \tag{C.7}$$

For this particular matrix, $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ coincide as

$$\boldsymbol{D} = j\omega \boldsymbol{I}_{N/2} - \frac{\beta^2}{j\omega} \frac{N}{2} \boldsymbol{L}_{N/2} \tag{C.8}$$

and the inverse of $\boldsymbol{M}$ can then be written as

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{D}^{-1} & \frac{\beta}{j\omega}\boldsymbol{H}_{N/2}\boldsymbol{L}_{N/2}\boldsymbol{D}^{-1} \\ \frac{\beta}{j\omega}\boldsymbol{D}^{-1}\boldsymbol{H}_{N/2} & \boldsymbol{D}^{-1} \end{bmatrix}. \tag{C.9}$$

It remains to find an expression for $\boldsymbol{D}^{-1}$. We first define the parameter $\psi \triangleq \frac{\beta^2}{(j\omega)^2}\frac{N}{2}$ and write

$$\boldsymbol{D} = j\omega\left(\boldsymbol{I}_{N/2} - \psi\boldsymbol{L}_{N/2}\right). \tag{C.10}$$

By (5.17), the matrix $\boldsymbol{L}_{N/2}$ is strictly lower triangular which implies that $\boldsymbol{L}_{N/2}^k = \boldsymbol{0}$ for $k \geq \frac{N}{2}$. We can therefore express $\boldsymbol{D}^{-1}$ by the Neumann series (generalized geometric series) of $\boldsymbol{L}_{N/2}$ as

$$\begin{aligned} \boldsymbol{D}^{-1} &= \frac{1}{j\omega}\left(\boldsymbol{I}_{N/2} - \psi\boldsymbol{L}_{N/2}\right)^{-1} && \text{(C.11)} \\ &= \frac{1}{j\omega}\sum_{k=0}^{\infty}\psi^k\boldsymbol{L}_{N/2}^k && \text{(C.12)} \\ &= \frac{1}{j\omega}\left(\boldsymbol{I}_{N/2} + \sum_{k=1}^{\frac{N}{2}-1}\psi^k\boldsymbol{L}_{N/2}^k\right). && \text{(C.13)} \end{aligned}$$

Because of the shape of $\boldsymbol{L}_{N/2}$, the matrix $\left(\boldsymbol{I}_{N/2} + \sum_{k=1}^{\frac{N}{2}-1}\psi^k\boldsymbol{L}_{N/2}^k\right)$ will have the form

$$\left(\boldsymbol{I}_{N/2} + \sum_{k=1}^{\frac{N}{2}-1}\psi^k\boldsymbol{L}_{N/2}^k\right) = \begin{pmatrix} 1 & 0 & & & \\ \psi & 1 & 0 & & \\ \psi^2 & \psi & 1 & 0 & \\ \vdots & & \ddots & \ddots & 0 \\ \psi^{\frac{N}{2}-1} & \cdots & & \psi & 1 \end{pmatrix} \tag{C.14}$$

To obtain a compact expression, we introduce the vector

$$\boldsymbol{\psi}_i^M \triangleq \begin{pmatrix} \boldsymbol{0}_i \\ \psi^0 \\ \psi \\ \vdots \\ \psi^{M-1-i} \end{pmatrix}, \tag{C.15}$$

where $\boldsymbol{0}_i \in \mathbb{R}^i$ is an all-zero column vector of length i. Using this vector, we write

$$\left(\boldsymbol{I}_{N/2} + \sum_{k=1}^{\frac{N}{2}-1}\psi^k\boldsymbol{L}_{N/2}^k\right) = \boldsymbol{\Psi}_{N/2} \tag{C.16}$$

where

$$\boldsymbol{\Psi}_{N/2} \triangleq \begin{bmatrix} \boldsymbol{\psi}_0^{N/2} & \cdots & \boldsymbol{\psi}_{N/2-1}^{N/2} \end{bmatrix}. \tag{C.17}$$

After the introduction of this helper matrix, we can write the expression for the transfer function as

$$G(\omega) = \boldsymbol{C}^{\mathsf{T}} \begin{bmatrix} \boldsymbol{D}^{-1} & \frac{\beta}{j\omega}\boldsymbol{H}_{N/2}\boldsymbol{L}_{N/2}\boldsymbol{D}^{-1} \\ \frac{\beta}{j\omega}\boldsymbol{D}^{-1}\boldsymbol{H}_{N/2} & \boldsymbol{D}^{-1} \end{bmatrix} \boldsymbol{H}'_N\boldsymbol{B} \tag{C.18}$$

$$= \boldsymbol{C}^{\mathsf{T}} \begin{bmatrix} \frac{1}{j\omega}\boldsymbol{\Psi}_{N/2} & \frac{\beta}{(j\omega)^2}\boldsymbol{H}_{N/2}\boldsymbol{L}_{N/2}\boldsymbol{\Psi}_{N/2} \\ \frac{\beta}{(j\omega)^2}\boldsymbol{\Psi}_{N/2}\boldsymbol{H}_{N/2} & \frac{1}{j\omega}\boldsymbol{\Psi}_{N/2} \end{bmatrix} \boldsymbol{H}'_N\boldsymbol{B} \tag{C.19}$$

Before proceeding, we recognize the following. As

$$\boldsymbol{B}' = (\beta \quad 0 \quad \cdots \quad 0)^{\mathsf{T}} \in \mathbb{R}^{N\times 1} \tag{C.20}$$

we get

$$\boldsymbol{H}'_N\boldsymbol{B}' = \beta \begin{bmatrix} \boldsymbol{1}_{N/2} \\ \boldsymbol{0}_{N/2} \end{bmatrix}. \tag{C.21}$$

Together with $\boldsymbol{C}^{\mathsf{T}} = (0 \quad \cdots \quad 1)$ we see that only $(\boldsymbol{M}^{-1})_{11} = \frac{\beta}{(j\omega)^2}\boldsymbol{\Psi}_{N/2}\boldsymbol{H}_{N/2}$ will influence the transfer function, and we can write

$$G(\omega) = (0 \ \cdots \ 1)\,\frac{\beta}{(j\omega)^2}\boldsymbol{\Psi}_{N/2}\boldsymbol{H}_{N/2}\beta\boldsymbol{1}_{N/2} \tag{C.22}$$

$$= \frac{\beta^2}{(j\omega)^2}(0 \ \cdots \ 1)\,\boldsymbol{\Psi}_{N/2}\boldsymbol{H}_{N/2}\boldsymbol{1}_{N/2} \tag{C.23}$$

The matrix product $\boldsymbol{\Psi}_{N/2}\boldsymbol{H}_{N/2}\boldsymbol{1}_{N/2}$ can be analyzed recursively as

$$\boldsymbol{\Psi}_{N/2}\boldsymbol{H}_{N/2}\boldsymbol{1}_{N/2} = \begin{bmatrix} \boldsymbol{\Psi}_{N/4} & \boldsymbol{0}_{N/4\times N/4} \\ \hat{\boldsymbol{\Psi}}_{N/4} & \boldsymbol{\Psi}_{N/4} \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_{N/4} & \boldsymbol{H}_{N/4} \\ \boldsymbol{H}_{N/4} & -\boldsymbol{H}_{N/4} \end{bmatrix} \boldsymbol{1}_{N/2} \tag{C.24}$$

$$= 2\begin{bmatrix} \boldsymbol{\Psi}_{N/4}\boldsymbol{H}_{N/4}\boldsymbol{1}_{N/4} \\ \hat{\boldsymbol{\Psi}}_{N/4}\boldsymbol{H}_{N/4}\boldsymbol{1}_{N/4} \end{bmatrix}, \tag{C.25}$$

where

$$\hat{\boldsymbol{\Psi}}_{N/4} \triangleq \begin{bmatrix} \psi^{N/4}\boldsymbol{\psi}_0 & \cdots & \psi\boldsymbol{\psi}_0 \end{bmatrix}. \tag{C.26}$$

Furthermore,

$$\hat{\boldsymbol{\Psi}}_{N/4}\boldsymbol{H}_{N/4}\boldsymbol{1}_{N/4} = \begin{bmatrix} \psi^{N/8}\hat{\boldsymbol{\Psi}}_{N/8} & \hat{\boldsymbol{\Psi}}_{N/8} \\ \psi^{N/4}\hat{\boldsymbol{\Psi}}_{N/8} & \psi^{N/8}\hat{\boldsymbol{\Psi}}_{N/8} \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_{N/8} & \boldsymbol{H}_{N/8} \\ \boldsymbol{H}_{N/8} & -\boldsymbol{H}_{N/8} \end{bmatrix} \boldsymbol{1}_{N/2} \tag{C.27}$$

$$= 2\begin{bmatrix} \psi^{N/8}\hat{\boldsymbol{\Psi}}_{N/8}\boldsymbol{H}_{N/8}\boldsymbol{1}_{N/8} \\ \psi^{N/4}\hat{\boldsymbol{\Psi}}_{N/8}\boldsymbol{H}_{N/8}\boldsymbol{1}_{N/8} \end{bmatrix}. \tag{C.28}$$

Starting at

$$\hat{\boldsymbol{\Psi}}_1\boldsymbol{H}_1\boldsymbol{1}_1 = \psi, \tag{C.29}$$

these recursive expressions may be combined to give

$$\boldsymbol{\Psi}_{N/2}\boldsymbol{H}_{N/2}\mathbf{1}_{N/2} = \frac{N}{2}\boldsymbol{\psi}_0^{N/2}. \tag{C.30}$$

Finally, the transfer function is given by

$$\begin{aligned}
G(\omega) &= (0\ \cdots\ 1)\,\frac{\beta}{(j\omega)^2}\boldsymbol{\Psi}_{N/2}\boldsymbol{H}_{N/2}\beta\mathbf{1}_{N/2} && \text{(C.31)}\\
&= \frac{\beta^2}{(j\omega)^2}\frac{N}{2}\,(0\ \cdots\ 1)\,\boldsymbol{\psi}_0^{N/2} && \text{(C.32)}\\
&= \frac{\beta^2}{(j\omega)^2}\frac{N}{2}\psi^{N/2-1} && \text{(C.33)}\\
&= \left(\frac{\beta^2}{(j\omega)^2}\frac{N}{2}\right)^{N/2} && \text{(C.34)}\\
&= \left(\sqrt{\frac{N}{2}}\frac{\beta}{j\omega}\right)^{N} && \text{(C.35)}
\end{aligned}$$

Multiple inputs

The transfer function expression (C.35) was derived assuming a single input only. We now show how this results generalizes for multiple inputs. We consider the case of $L = 2$ inputs and the extension to arbitrary L is straightforward.

For $L > 1$, we let $N \triangleq N_\ell L$, where N_ℓ is the system order for a single channel. For $L = 2$ we have

$$\boldsymbol{C}^{\mathsf{T}} = \begin{pmatrix} \mathbf{0}_{N/2}^{\mathsf{T}} & 0 & \cdots & 1 & 0 & \cdots & 0 \\ \mathbf{0}_{N/2}^{\mathsf{T}} & 0 & \cdots & 0 & 0 & \cdots & 1 \end{pmatrix} \in \mathbb{R}^{2\times N}, \tag{C.36}$$

$$\boldsymbol{B}' = \begin{pmatrix} \beta & \cdots & 0 & 0 & \cdots & 0 & \mathbf{0}_{N/2}^{\mathsf{T}} \\ 0 & \cdots & 0 & \beta & \cdots & 0 & \mathbf{0}_{N/2}^{\mathsf{T}} \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}^{N\times 2}, \tag{C.37}$$

and

$$\boldsymbol{H}_N'\boldsymbol{B}' = \beta\begin{bmatrix} \mathbf{1}_{N_\ell/2} & \mathbf{1}_{N_\ell/2} \\ \mathbf{1}_{N_\ell/2} & -\mathbf{1}_{N_\ell/2} \\ \mathbf{0}_{N/2} & \mathbf{0}_{N/2} \end{bmatrix} \in \mathbb{R}^{N\times 2}, \tag{C.38}$$

$\boldsymbol{A}'$ is as given by (5.16), but with

$$\boldsymbol{L}_{N/2} \triangleq \begin{bmatrix} \boldsymbol{L}_{N_\ell/2} & \mathbf{0}_{N_\ell/2\times N_\ell/2} \\ \mathbf{0}_{N_\ell/2\times N_\ell/2} & \boldsymbol{L}_{N_\ell/2} \end{bmatrix}. \tag{C.39}$$

In (C.12) we used the power series of $\boldsymbol{L}_{N/2}$. For $L = 2$ we have

$$\boldsymbol{L}_{N/2}^{k} = \begin{bmatrix} \boldsymbol{L}_{N_\ell/2} & \boldsymbol{0}_{N_\ell/2 \times N_\ell/2} \\ \boldsymbol{0}_{N_\ell/2 \times N_\ell/2} & \boldsymbol{L}_{N_\ell/2} \end{bmatrix}^{k} \tag{C.40}$$

$$= \begin{bmatrix} \boldsymbol{L}_{N_\ell/2}^{k} & \boldsymbol{0}_{N_\ell/2 \times N_\ell/2} \\ \boldsymbol{0}_{N_\ell/2 \times N_\ell/2} & \boldsymbol{L}_{N_\ell/2}^{k} \end{bmatrix}, \tag{C.41}$$

and in consequence

$$\boldsymbol{\Psi}_{N/2} = \begin{bmatrix} \boldsymbol{\Psi}_{N_\ell/2} & \boldsymbol{0}_{N_\ell/2 \times N_\ell/2} \\ \boldsymbol{0}_{N_\ell/2 \times N_\ell/2} & \boldsymbol{\Psi}_{N_\ell/2} \end{bmatrix}. \tag{C.42}$$

Furthermore

$$\boldsymbol{\Psi}_{N/2}\boldsymbol{H}_{N/2} = \begin{bmatrix} \boldsymbol{\Psi}_{N_\ell/2} & \boldsymbol{0}_{N_\ell/2 \times N_\ell/2} \\ \boldsymbol{0}_{N_\ell/2 \times N_\ell/2} & \boldsymbol{\Psi}_{N_\ell/2} \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_{N_\ell/2} & \boldsymbol{H}_{N_\ell/2} \\ \boldsymbol{H}_{N_\ell/2} & -\boldsymbol{H}_{N_\ell/2} \end{bmatrix} \tag{C.43}$$

$$= \begin{bmatrix} \boldsymbol{\Psi}_{N_\ell/2}\boldsymbol{H}_{N_\ell/2} & \boldsymbol{\Psi}_{N_\ell/2}\boldsymbol{H}_{N_\ell/2} \\ \boldsymbol{\Psi}_{N_\ell/2}\boldsymbol{H}_{N_\ell/2} & -\boldsymbol{\Psi}_{N_\ell/2}\boldsymbol{H}_{N_\ell/2} \end{bmatrix}, \tag{C.44}$$

and

$$\boldsymbol{\Psi}_{N/2}\boldsymbol{H}_{N/2} \begin{bmatrix} \boldsymbol{1}_{N_\ell/2} & \boldsymbol{1}_{N_\ell/2} \\ \boldsymbol{1}_{N_\ell/2} & -\boldsymbol{1}_{N_\ell/2} \end{bmatrix} = \begin{bmatrix} 2\boldsymbol{\Psi}_{N_\ell/2}\boldsymbol{H}_{N_\ell/2} & \boldsymbol{0}_{N_\ell/2} \\ \boldsymbol{0}_{N_\ell/2} & 2\boldsymbol{\Psi}_{N_\ell/2}\boldsymbol{H}_{N_\ell/2} \end{bmatrix}, \tag{C.45}$$

The expression for the transfer function then becomes

$$\boldsymbol{G}(\omega) = \frac{\beta^2}{(j\omega)^2} \begin{pmatrix} 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{pmatrix} \boldsymbol{\Psi}_{N/2}\boldsymbol{H}_{N/2} \begin{bmatrix} \boldsymbol{1}_{N_\ell/2} & \boldsymbol{1}_{N_\ell/2} \\ \boldsymbol{1}_{N_\ell/2} & -\boldsymbol{1}_{N_\ell/2} \end{bmatrix} \tag{C.46}$$

$$= \frac{2\beta^2}{(j\omega)^2} \begin{pmatrix} 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{pmatrix} \begin{bmatrix} \boldsymbol{\Psi}_{N_\ell/2}\boldsymbol{H}_{N_\ell/2} & \boldsymbol{0}_{N_\ell/2} \\ \boldsymbol{0}_{N_\ell/2} & \boldsymbol{\Psi}_{N_\ell/2}\boldsymbol{H}_{N_\ell/2} \end{bmatrix} \tag{C.47}$$

$$= \begin{pmatrix} \left(\sqrt{\frac{2N_\ell}{2}\frac{\beta}{j\omega}}\right)^{N_\ell} \\ \left(\sqrt{\frac{2N_\ell}{2}\frac{\beta}{j\omega}}\right)^{N_\ell} \end{pmatrix} = \begin{pmatrix} \left(\sqrt{\frac{N}{2}\frac{\beta}{j\omega}}\right)^{N_\ell} \\ \left(\sqrt{\frac{N}{2}\frac{\beta}{j\omega}}\right)^{N_\ell} \end{pmatrix}. \tag{C.48}$$

From this derivation we see that for an ADC with an arbitrary number of input channels, the transfer function is

$$G(\omega) = \left(\sqrt{\frac{N}{2}\frac{\beta}{j\omega}}\right)^{N_\ell} \tag{C.49}$$

for all channels.

Bibliography

[1] T. C. Carusone, D. Johns, and K. Martin, *Analog Integrated Circuit Design, second edition.* John Wiley and Sons Inc., 2013.

[2] H. Loeliger, L. Bolliger, G. Wilckens, and J. Biveroni, "Analog-to-digital conversion using unstable filters," in *2011 Information Theory and Applications Workshop*, pp. 1–4, 2011.

[3] H. Loeliger and G. Wilckens, "Control-based analog-to-digital conversion without sampling and quantization," in *2015 Information Theory and Applications Workshop (ITA)*, pp. 119–122, 2015.

[4] 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.

[5] H. Malmberg, "Control-bounded converters." 2020.

[6] H. Malmberg and H. A. Loeliger, "Analog-to-digital conversion using self-averaging analog hadamard networks," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–1, 2020.

[7] B. D. O. Anderson and J. B. Moore, *Optimal Filtering.* Prentice Hall, 1979.

[8] I. Cadence Design Systems, "Spectre simulation platform." `https://www.cadence.com/ko_KR/home/tools/custom-ic-analog-rf-design/circuit-simulation/spectre-simulation-platform.html`, 2020.

[9] F. Feyling, "cbadcsim2." `https://github.com/fredrief/cbadcsim2`, 2020.