Control-Bounded ADC

Fredrik Feyling

December 2020

Chapter 1

Introduction

Chapter 2

Control-Bounded ADC

2.1 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 [1]. 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 [2], 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 [3]. 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 [3], has recently defended his Ph.D. on Control-Bounded Converters. The author of this report has been given early access to a draft of the thesis that is not yet published [4].

In this section, the operating principle of a control-bounded converter is described in detail, and we follow the notation established in [3]. The theoretical presentation given in this section will be very close to that of [4], but less general and limited to what necessary for understanding the presented results. Certain derivations that would normally be cited, will be given in appendices as the reference is not available to the public.

2.2 Overview

The control-bounded ADC approaches the A/D conversion problem differently compared to conventional converters from section **??**. The control bounded ADC consists of three building blocks; an analog system (AS), digital control (DC) and digital estimator (DE). The complete system is illustrated in figure 2.1.

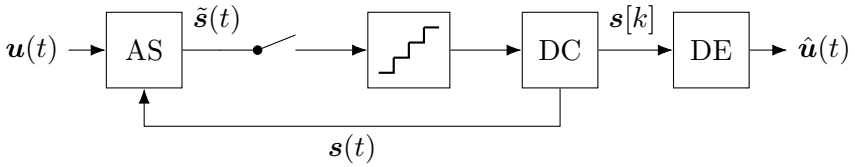


Figure 2.1: The control-bounded A/D converter

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. It is evident that the structure of the continuous time $\Sigma\Delta$ converter from figure **??** is very similar to that of figure 2.1. The fundamental difference between these converters lies in the interpretation of the intermediate quantity $\boldsymbol{s}[k]$. In the conventional view, this signal is a sampled and quantized version of the input signal $\boldsymbol{u}(t)$. In the control-bounded perspective, this signal is a control-signal that stabilizes the analog system, and is therefore only indirectly related to the input signal. These two perspectives result in two different ways of estimating $\hat{\boldsymbol{u}}(t)$.

The advantage of this approach may not be evident at first sight, as figure 2.1 looks like a $\Sigma\Delta$ converter with an alternative decimation filter. The main contribution of this approach is that defines a new interface between the analog and digital domain, that enables combinations of ASs and DCs that where previously unimaginable. This will become more clear after the more detailed analysis of the building blocks, given in the following sections.

2.3 Analog System

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

2.3.1 State Space Model

The dynamics of the AS is described using a state space model notation, illustrated in figure 2.2.

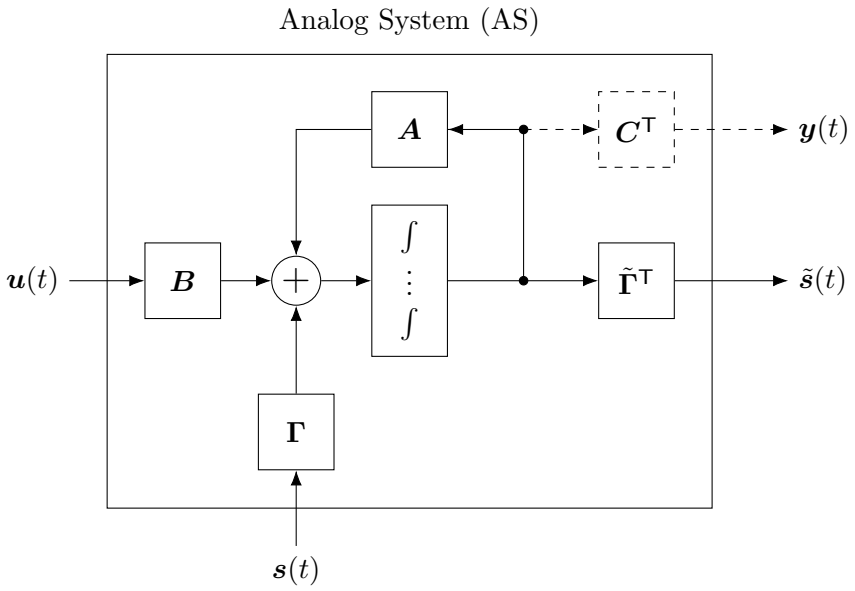


Figure 2.2: State space model of the AS.

The relation between a multi-channel input signal

$$\boldsymbol{u}(t) \triangleq (u_1(t), ..., u_L(t))^{\mathsf{T}} \in \mathbb{R}^L, \tag{2.1}$$

the state vector

$$\boldsymbol{x}(t) \triangleq (s_1(t), ..., s_N(t))^{\mathsf{T}} \in \mathbb{R}^N, \tag{2.2}$$

and the control contribution

$$\boldsymbol{s}(t) \triangleq (x_1(t), ..., x_M(t))^{\mathsf{T}} \in \mathbb{R}^M \tag{2.3}$$

is given by the differential equation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{\Gamma}\boldsymbol{s}(t). \tag{2.4}$$

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 $\boldsymbol{\Gamma} \in \mathbb{R}^{N\times M}$ are all real valued matrices.

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 is used by the DC to produce the control signal $\boldsymbol{s}(t)$. The latter is a purely 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

$$\tilde{\boldsymbol{s}}(t) \triangleq \tilde{\boldsymbol{\Gamma}}^{\mathsf{T}}\boldsymbol{x}(t) \in \mathbb{R}^{\tilde{M}}. \tag{2.5}$$

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

$$\boldsymbol{y}(t) \triangleq \boldsymbol{C}^{\mathsf{T}}\boldsymbol{x}(t) \tag{2.6}$$

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 physical design of the AS. This is indicated by dashed lines in figure 2.2. Note also that the number of controls M and the number of control observations $\tilde{M}$ are not required to be the same. Finally, note that the control contribution enter the AS in an additive way.

2.3.2 Transfer Function and Impulse Response Matrix

The set of differential equations (2.4) result in the transfer function matrix

$$\boldsymbol{G}(\omega) = \boldsymbol{C}^{\mathsf{T}}(j\omega\boldsymbol{I}_N - \boldsymbol{A})^{-1}\boldsymbol{B} \in \mathbb{C}^{\tilde{N}\times L} \tag{2.7}$$

and impulse response matrix

$$\boldsymbol{g}(t) = \boldsymbol{C}^{\mathsf{T}} \exp(\boldsymbol{A}t)\boldsymbol{B} \in \mathbb{R}^{\tilde{N}\times L}, \tag{2.8}$$

where exp(.) denotes the matrix exponential.

We will refer to (2.7) as the analog transfer function (ATF) matrix and similarly (2.8) as the analog impulse response matrix. An element $G_{k,l}$ of $\boldsymbol{G}(\omega)$ gives the transfer function from the input signal $u_l(t)$ to state $x_k(t)$, and similarly for $\boldsymbol{g}(t)$. This means that for the case of a scalar input signal, the ATF matrix reduces to a column vector.

2.4 Digital Control

The DC is a discrete time system, synchronized by a clock with period denoted T. Its only task is to keep the state vector $\boldsymbol{x}(t)$ bounded within its physical limits. A DC that can maintain a bounded AS state for a bounded input signal is called effective.

The DC operates by observing a sampled and quantized version of the control observation (2.5) and producing a control contribution (2.3) in response. The control contribution could in general take any waveform, but will here be assumed to be given by

$$s_\ell(t) = d(t - kT)s_\ell[k], \tag{2.9}$$

where the waveform $d(t)$ is given by

$$d(t) \triangleq \begin{cases} 1 & \text{if } t \in [0, T) \\ 0 & \text{otherwise.} \end{cases} \tag{2.10}$$

2.4.1 Effective digital control?

2.5 Digital Estimator

The digital estimator (DE) forms an estimate $\hat{\boldsymbol{u}}(t)$ of $\boldsymbol{u}(t)$ based on the control signals $\boldsymbol{s}[k]$, its corresponding control contribution $\boldsymbol{s}(t)$ 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.

Before going into details on the filter, we highlight the difference between the $\Sigma\Delta$- and the control-bounded interpretation of the control signals $\boldsymbol{s}[k]$ and $\boldsymbol{s}(t)$. In the $\Sigma\Delta$ perspective, $\boldsymbol{s}[k]$ represent a sampled and quantized version of some linear mapping of the state vector $\boldsymbol{x}(t)$, cf. (2.5). This interpretation results in the estimation filter being a low-pass filter, which constitutes the reconstruction part of a $\Sigma\Delta$ ADC together with a decimation filter.

In the control-bounded perspective, we completely ignore this relation between $\boldsymbol{s}[k]$ and $\boldsymbol{x}(t)$. Instead, we focus on the fact that the control contribution $\boldsymbol{s}(t)$ result in an effective control. We interpret $\boldsymbol{s}(t)$ as the control contribution needed to bound the state vector $\boldsymbol{x}(t)$, triggered by the input signal $\boldsymbol{u}(t)$. Thus, the $\boldsymbol{s}(t)$ must contain some indirect information about the input signal. How accurately the input can be reconstructed from the control signal depends on the gain of the AS, as will become evident in the following analysis.

2.5.1 Statistical Estimation Problem and Transfer Functions

In the following analysis, the system described by (2.4) 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.

Imagine the fictional signal

$$\breve{\boldsymbol{y}}(t) \triangleq (\boldsymbol{g} * \boldsymbol{u})(t) \in \mathbb{R}^{\tilde{N}}, \tag{2.11}$$

which is the signal observation that would have result in the absence of a DC. The actual signal observation can then be written as

$$\boldsymbol{y}(t) = \breve{\boldsymbol{y}}(t) - \boldsymbol{q}(t), \tag{2.12}$$

where $\boldsymbol{q}(t)$ is the control contribution seen at the signal observation, i.e. $\boldsymbol{q}(t) = (\boldsymbol{h} * \boldsymbol{s})(t)$. (2.12) holds, because the control contribution enter the AS in an additive way, as indicated in figure 2.2. Note at $\boldsymbol{q}(t)$ is fully determined by $\boldsymbol{s}[k]$ and is therefore known to the DE.

Given an estimation filter with impulse response matrix $\boldsymbol{h}(t) \in \mathbb{R}^{L \times \tilde{N}}$, the continuous time estimate $\hat{\boldsymbol{u}}(t)$ of $\boldsymbol{u}(t)$ is given by

$$\hat{\boldsymbol{u}}(t) = (\boldsymbol{h} * \boldsymbol{q})(t) \in \mathbb{R}^{L}. \tag{2.13}$$

This estimation problem is visualized in figure 2.3.

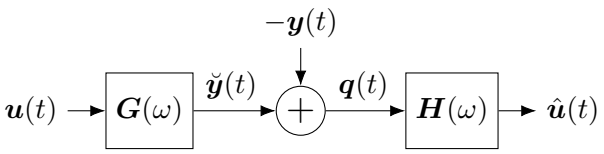


Figure 2.3: The estimation problem of a control-bounded converter

In order to determine $\boldsymbol{h}(t)$, assume $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$ are both independent, multivariate, centered, and wide-sense stationary stochastic processes. We want the filter to minimize the mean square error between $\hat{\boldsymbol{u}}(t)$ and $\boldsymbol{u}(t)$. Thus, the impulse response matrix $\boldsymbol{h}(t)$ should satisfy

$$\boldsymbol{h}(t) = \underset{\bar{\boldsymbol{h}}}{\operatorname{argmin}} \, \mathrm{E}[(\hat{\boldsymbol{u}}(t) - \boldsymbol{u}(t))^2] \tag{2.14}$$

$$= \underset{\bar{\boldsymbol{h}}}{\operatorname{argmin}} \, \mathrm{E}[((\bar{\boldsymbol{h}} * \boldsymbol{q})(t) - \boldsymbol{u}(t))^2]. \tag{2.15}$$

This is exactly the objective of the Wiener-filter [5], and the impulse response matrix is given by the solution to the well known Wiener-Hopf equations:

$$(\boldsymbol{h} * \boldsymbol{R}_{\boldsymbol{q}\boldsymbol{q}^\mathsf{T}})(\tau) = \boldsymbol{R}_{\boldsymbol{u}\boldsymbol{q}^\mathsf{T}}(-\tau) \tag{2.16}$$

where

$$\boldsymbol{R}_{\boldsymbol{q}\boldsymbol{q}^\mathsf{T}} \triangleq \mathrm{E}[\boldsymbol{q}(t)\boldsymbol{q}(t+\tau)^\mathsf{T}] \tag{2.17}$$

$$\boldsymbol{R}_{\boldsymbol{u}\boldsymbol{q}^\mathsf{T}} \triangleq \mathrm{E}[\boldsymbol{u}(t)\boldsymbol{q}(t+\tau)^\mathsf{T}] \tag{2.18}$$

are the autocovariance and cross-covariance respectively. By taking the Fourier transform of (2.16) we obtain the frequency response matrix $\boldsymbol{H}(\omega)$ as

$$\boldsymbol{H}(\omega) = \boldsymbol{G}^\mathsf{H}(\omega) \left(\boldsymbol{G}(\omega)\boldsymbol{G}^\mathsf{H}(\omega) + \eta^2 \boldsymbol{I}_N \right)^{-1} \tag{2.19}$$

The parameter η is defined as $\eta \triangleq \frac{\sigma_{\boldsymbol{y}}^2}{\sigma_{\boldsymbol{u}}^2}$, where $\sigma_{\boldsymbol{y}}^2$ and $\sigma_{\boldsymbol{u}}^2$ are the power spectral densities of $\boldsymbol{y}(t)$ and $\boldsymbol{u}(t)$ respectively.

Transfer functions? Scalar case? Bandwidth and η?

2.5.2 Estimation filter implementation

With the digital estimation filter described by (2.19), the estimation could in principle be carried out by computing $\hat{\boldsymbol{u}}(t)$ as in (2.13). This computation is however not straight forward. The signal $\boldsymbol{q}(t)$ is not bounded by $b_{\boldsymbol{y}}$ and might therefore contain unbounded quantities. In addition, the computation of $\boldsymbol{q}(t)$ from $\boldsymbol{s}[k]$ might also be computationally expensive.

In [1] 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 (2.13) as the considered time window extends towards infinity. The algorithm is also indifferent to the stability assumptions made in the previous section. In this section, a concise description of the filter algorithm is given, and the reader is referred to [3] for the derivation.

The algorithm consist of a forward recursion

$$\overrightarrow{\boldsymbol{m}}_{k+1} \triangleq \boldsymbol{A}_f \overrightarrow{\boldsymbol{m}}_k + \boldsymbol{B}_f \boldsymbol{s}[k], \tag{2.20}$$

a backward recursion

$$\overleftarrow{\boldsymbol{m}}_{k-1} \triangleq \boldsymbol{A}_b \overleftarrow{\boldsymbol{m}}_k + \boldsymbol{B}_b \boldsymbol{s}[k-1], \tag{2.21}$$

and finally the estimate

$$\hat{\boldsymbol{u}}(t_k) \triangleq \boldsymbol{W}^{\mathsf{T}} (\overleftarrow{\boldsymbol{m}}_k - \overrightarrow{\boldsymbol{m}}_k) \tag{2.22}$$

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.

$$\boldsymbol{A}_f \triangleq \exp\left(\left(\boldsymbol{A} - \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}\right) T\right) \tag{2.23}$$

$$\boldsymbol{A}_b \triangleq \exp\left(-\left(\boldsymbol{A} + \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}\right) T\right) \tag{2.24}$$

$$\boldsymbol{B}_f \triangleq \int_0^T \exp\left(\left(\boldsymbol{A} - \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}\right)(T-\tau)\right) \boldsymbol{\Gamma} d\tau \tag{2.25}$$

$$\boldsymbol{B}_b \triangleq -\int_0^T \exp\left(-\left(\boldsymbol{A} + \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}\right)(T-\tau)\right) \boldsymbol{\Gamma} d\tau \tag{2.26}$$

In equations (2.23 - 2.26), 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 (2.23 - 2.26) is obtained by solving the continuous-time algebraic Riccati (CARE) equations

$$\boldsymbol{A}\overrightarrow{\boldsymbol{V}} + \left(\boldsymbol{A}\overrightarrow{\boldsymbol{V}}\right)^{\mathsf{T}} + \boldsymbol{B}\boldsymbol{B}^{\mathsf{T}} - \frac{1}{\eta^2}\overrightarrow{\boldsymbol{V}}\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}\overrightarrow{\boldsymbol{V}} = \boldsymbol{0}_{N\times N} \tag{2.27}$$

and

$$\boldsymbol{A}\overleftarrow{\boldsymbol{V}} + \left(\boldsymbol{A}\overleftarrow{\boldsymbol{V}}\right)^{\mathsf{T}} - \boldsymbol{B}\boldsymbol{B}^{\mathsf{T}} + \frac{1}{\eta^2}\overleftarrow{\boldsymbol{V}}\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}\overleftarrow{\boldsymbol{V}} = \boldsymbol{0}_{N\times N} \tag{2.28}$$

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

$$\left(\overrightarrow{\boldsymbol{V}} + \overleftarrow{\boldsymbol{V}}\right)\boldsymbol{W} = \boldsymbol{B} \tag{2.29}$$

Chapter 3

Hadamard ADC

The main advantage of control-bounded ADCs is flexibility. The only requirement set on the system by the estimation filter is that the it obeys the differential equations (2.4). This flexibility allows the designer to taylor the system against the application to a larger degree than what is possible in conventional ADC.

The application of this work is modern high-end ultrasound probes. Such probes has up to 10,000 transducers stacked in a 2D array, and the large number of transducers is used for beamforming. With todays technology, having 10,000 ADCs inside the probe is not possible due to restrictions on area and current consumption. Todays solutions therefore employ combinations of analog and digital beamforming, dividing the transducer array into sub arrays sharing one ADC. The transducers of each sub array are combined with analog delay-and-add techniques. Having full control of each transducer would of course be favorable.

In this work, we want to take advantage of this number of channels. Instead of converting each channel individually we view the problem of converting 10,000 analog signals as one big task, resulting in one huge ADC instead of 10,000 smaller ones. The goal is that the resulting ADC will have a current consumption less than 10,000 times a single state-of-the-art ADC.

The Hadamard ADC is based on the chain-of-integrator ADC and apply the Hadamard matrix, $\boldsymbol{H}_N$, to rotate the state vector of the analog system. For N being powers of two, the Hadamard matrix is defined recursively as

$$\boldsymbol{H}_N = \boldsymbol{H}_2 \otimes \boldsymbol{H}_{N/2} \tag{3.1}$$

where

$$\boldsymbol{H}_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{3.2}$$

The Hadamard matrix is an orthogonal matrix with the useful properties

$$\boldsymbol{H}_N = \boldsymbol{H}_N^{\mathsf{T}} \tag{3.3}$$

and

$$\boldsymbol{H}_N^{\mathsf{T}}\boldsymbol{H}_N = N\boldsymbol{I}_N. \tag{3.4}$$

When transforming the state vector, the energy from all input channels will be equally distributed over all involved components. The consequence of this is that the overall ADC can be scaled towards the average, rather than the maximum signal energy. Depending on the spatial peak-to-average ration of the input channels, this will result in a gain of SNR, as will be further explored the next chapter.

3.1 Analog System

The Hadamard ADC is described by the equations

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{\Gamma}\boldsymbol{s}(t) \tag{3.5}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}^{\mathsf{T}}\boldsymbol{x}(t) \tag{3.6}$$

and

$$\tilde{\boldsymbol{s}}(t) = \tilde{\boldsymbol{\Gamma}}^{\mathsf{T}}\boldsymbol{x}(t). \tag{3.7}$$

The AS is determined solely by $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}^{\mathsf{T}}$. $\boldsymbol{\Gamma}$ and $\tilde{\boldsymbol{\Gamma}}^{\mathsf{T}}$ only affects the performance of the DC which will be described in the next section. The state-space matrices $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}^{\mathsf{T}}$ must be chosen such that the desired state-vector rotation is obtained, and at the same time provide high gain within the frequency band of interest. In addition, the parametrization should allow an energy efficient hardware implementation.

The proposed system is described by

$$\boldsymbol{A} = \boldsymbol{H}'_N\boldsymbol{A}' \in \mathbb{R}^{N\times N}, \tag{3.8}$$

$$\boldsymbol{B} = \boldsymbol{H}'_N\boldsymbol{B}' \in \mathbb{R}^{N\times L} \tag{3.9}$$

and

$$\boldsymbol{C} = \boldsymbol{C}' \in \mathbb{R}^{N\times L}, \tag{3.10}$$

where

$$\boldsymbol{H}'_N = \begin{bmatrix} \boldsymbol{H}_{N/2} & \boldsymbol{0}_{N/2} \\ \boldsymbol{0}_{N/2} & \boldsymbol{H}_{N/2} \end{bmatrix} \in \mathbb{R}^{N\times N} \tag{3.11}$$

and

$$\boldsymbol{A}' = \begin{bmatrix} \boldsymbol{0}_{N/2} & \beta\boldsymbol{L}_{N/2} \\ \beta\boldsymbol{I}_{N/2} & \boldsymbol{0}_{N/2} \end{bmatrix} \in \mathbb{R}^{N\times N}. \tag{3.12}$$

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. For the single input case,

$$\boldsymbol{L}_{N/2} = \begin{pmatrix} 0 & & & & \\ 1 & 0 & & & \\ 0 & 1 & 0 & & \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix} \in \mathbb{R}^{\frac{N}{2} \times \frac{N}{2}}, \tag{3.13}$$

$$\boldsymbol{B}' = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}^{N \times L} \tag{3.14}$$

and

$$\boldsymbol{C}' = \begin{pmatrix} 0 & 0 & \cdots & 1 \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}^{N \times L}. \tag{3.15}$$

Note that $\boldsymbol{C}$ only determines which states of the AS that is considered as an output by the DE. Thus it is a purely conceptual matrix with no physical implementation. With this choice of $\boldsymbol{C}$, only the last state of the AS is treated as an output. It is also possible to consider multiple outputs and this will indeed give increased performance. However, for simplicity, only the single output case is treated in this report.

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 parametrization matrices, we restrict both N_ℓ and L to be powers of 2. For $L > 1$, the state-space matrices generalizes as

$$\boldsymbol{L}_{N/2} = \begin{bmatrix} \boldsymbol{L}_{N_\ell/2} & & \\ & \ddots & \\ & & \boldsymbol{L}_{N_\ell/2} \end{bmatrix} \in \mathbb{R}^{\frac{N}{2} \times \frac{N}{2}}, \tag{3.16}$$

$$\boldsymbol{B}' = \begin{bmatrix} \boldsymbol{B}'_\ell & & \\ & \ddots & \\ & & \boldsymbol{B}'_\ell \end{bmatrix} \in \mathbb{R}^{N \times L} \tag{3.17}$$

and

$$\boldsymbol{C}' = \begin{bmatrix} \boldsymbol{C}'_\ell & & \\ & \ddots & \\ & & \boldsymbol{C}'_\ell \end{bmatrix} \in \mathbb{R}^{N \times L}. \tag{3.18}$$

With $N = LN_\ell$ and $\boldsymbol{L}_{N/2}$ as above, $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}^{\mathsf{T}}$ is still given by (3.8)-(3.10). In this single output case, the transfer function is a column vector given by

$$\boldsymbol{G}(\omega) = \boldsymbol{C}^{\mathsf{T}} \left(j\omega \boldsymbol{I}_N - \boldsymbol{A} \right)^{-1} \boldsymbol{B} \tag{3.19}$$

$$= \boldsymbol{C}^{\mathsf{T}} \left(j\omega \boldsymbol{I}_N - \boldsymbol{H}'_N \boldsymbol{A}' \right)^{-1} \boldsymbol{H}'_N \boldsymbol{B}', \tag{3.20}$$

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

$$G(\omega) = \left(\sqrt{\frac{N}{2}}\frac{\beta}{j\omega}\right)^{N_\ell} \tag{3.21}$$

A hardware implementation of the proposed AS is shown in figure 3.2 for $N = 8$ and $L = 2$. For this example, $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \boldsymbol{H}_8'\boldsymbol{A}' \tag{3.22}$$

$$= \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 \\ \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.23}$$

The Hadamard matrices $\boldsymbol{H}_4(Z)$ is easily implemented as shown in figure 3.1. The integrators can be implemented using either operational voltage- or transconductance amplifiers.

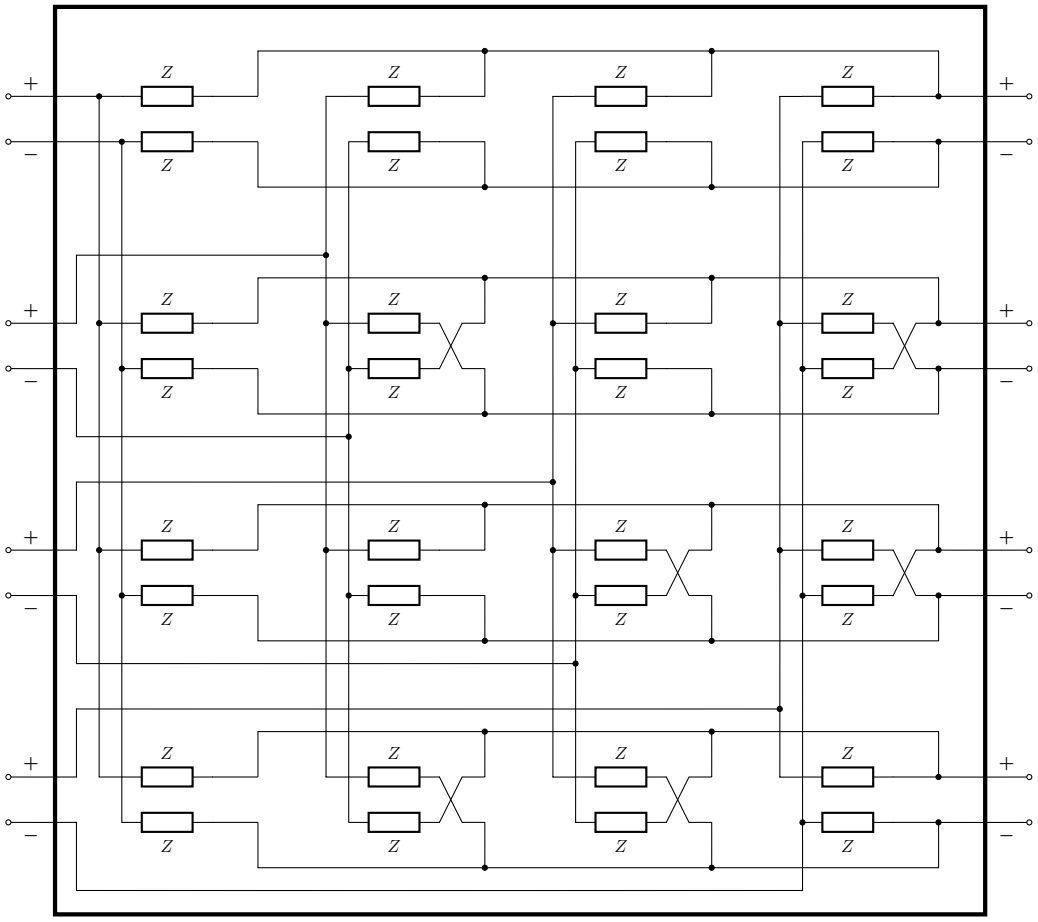


Figure 3.1: A 4th order Hadamard matrix implemented with impedance Z. Straight wires correspond to a multiplication of 1 and crossing wires to multiplication of -1

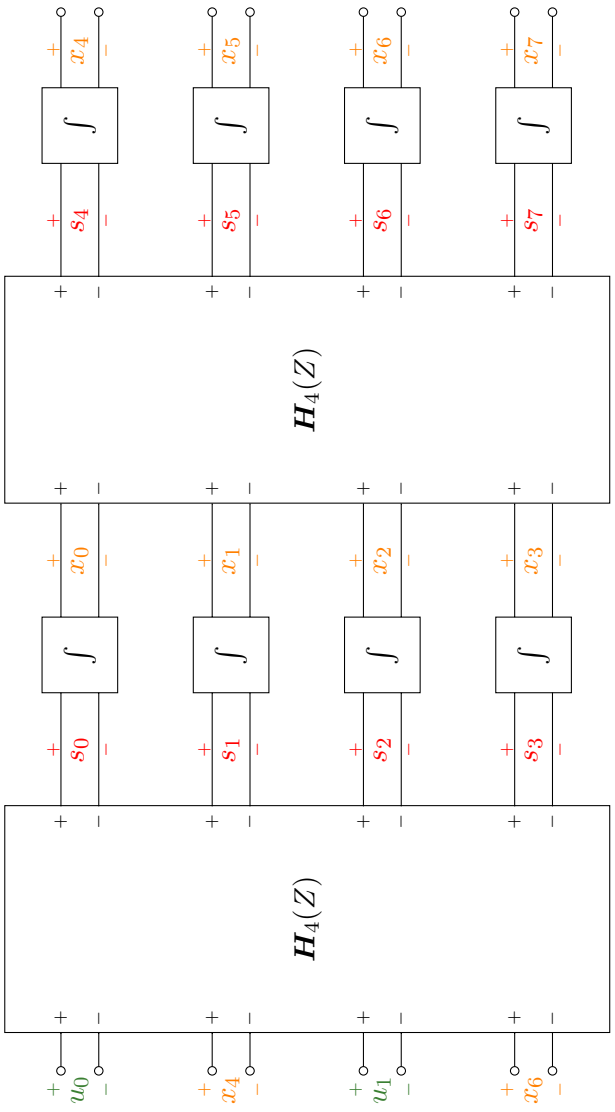
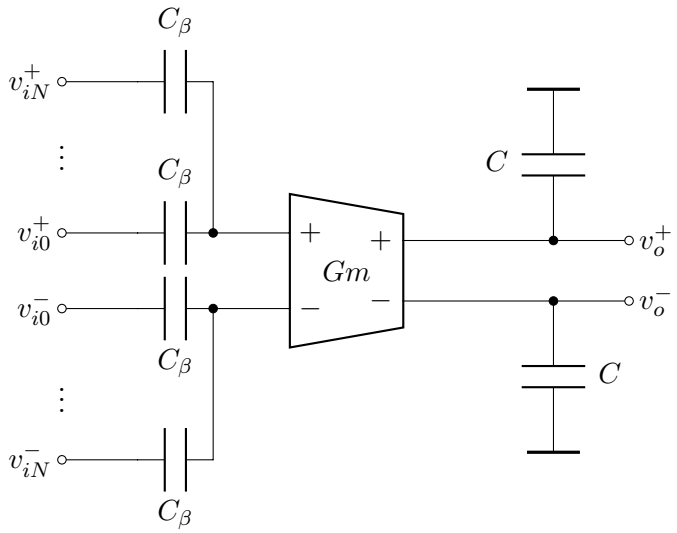
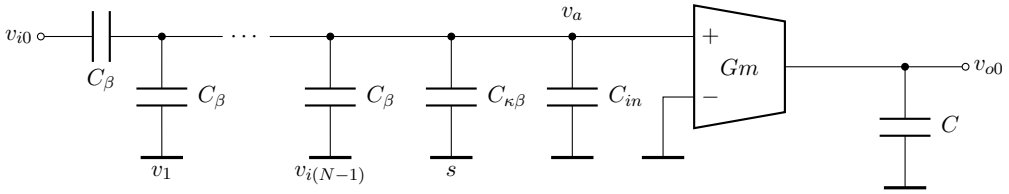
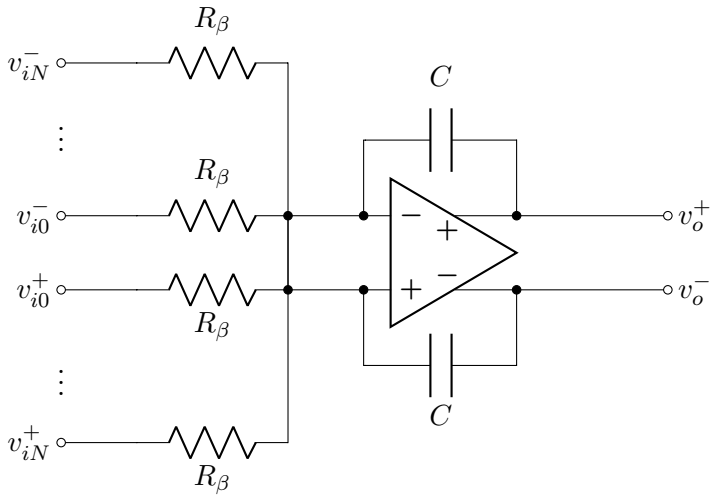


Figure 3.2: Proposed hardware implementation of the Hadamard ADC AS for $N = 8, L = 2$


C_β
v_{iN}^+
C_β
C
v_{i0}^+
Gm
v_o^+
v_{i0}^-
v_o^-
C_β
C
v_{iN}^-
C_β


v_{i0}
C_β
C_β
v_1
C_β
$v_{i(N-1)}$
$C_{\kappa\beta}$
s
v_a
C_{in}
Gm
v_{o0}
C


R_β
v_{iN}^-
C
R_β
v_{i0}^-
v_o^+
v_{i0}^+
v_o^-
R_β
C
v_{iN}^+
R_β

Bibliography

[1] 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.

[2] 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.

[3] 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.

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

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