Doctoral Thesis

Digital estimation of continuous-time signals using factor graphs

Author[s]:
Bolliger, Lukas

Publication Date:
2012

Permanent Link:
https://doi.org/10.3929/ethz-a-007342807

Rights / License:
In Copyright - Non-Commercial Use Permitted
Digital Estimation of Continuous-Time Signals Using Factor Graphs

A dissertation submitted to
ETH Zurich
for the degree of
Doctor of Sciences

presented by
Lukas Bolliger

Master of Science ETH in Electrical Engineering and Information Technology, ETH Zurich, Switzerland
born on May 03, 1981
citizen of Luzern LU and Uerkheim AG

accepted on the recommendation of
Prof. Dr. H.-A. Loeliger, examiner
Prof. Dr. A. Singer, co-examiner
Dr. Ch. Vogel, co-examiner

2012
It might seem daunting to think that effort and chance, as much as innate talent, are what counts. But I find it encouraging because, while our genetic makeup is out of our control, our degree of effort is up to us. And the effects of chance, too, can be controlled to the extent that by committing ourselves to repeated attempts, we can increase our odds of success.

(“The Drunkard’s Walk”, Leonard Mladinov)
I would like to express my gratitude to Prof. H.-A. Loeliger for giving me the opportunity to be one of his PhD students. It was great to be advised by someone like Andi whose enthusiasm for science is catching. He was always eager to help when I was stuck and his ability to grasp my problem was always impressive. After meetings with him my brain was almost overloaded with new ideas and approaches to continue my work.

Next I would like to thank Dr. Christian Vogel, it was in collaboration with him during his time at the ISI when my research topic took shape. I was especially pleased when he agreed to be one of the co-referees at my PhD examination. Thanks also go to Prof. Andrew Singer for the interesting discussions when he was in Zürich to also be a co-referee at my PhD defence.

I would like to thank everybody at the ISI. All the lunch and coffee breaks, where also non-scientific topics were discussed, were a great balance to work. I especially want to mention Georg Wilckens, who is a great friend since the beginning of our time at ETH. It was great to extend our time together at ETH by working at the same institute. Christoph Reller was always helpful when mathematical problems arose and a great companion at the EUSIPCO in Barcelona. Mehdi Molkaraie, who was my office mate, helped me to motivate myself when it came to writing my thesis. Rita Hildebrand, with her cheerful spirit, was always helpful with administrative issues and Patrick Strebel was always helpful when it came to technical problems.

Thanks also go to the everyone who joined the “Fiiroebier” each Friday evening, which was always a great start to the weekend. Espe-
cially Clemens Wacha, who was always happy to meet for a beer and was a great support when work or private life happened to be stressful. Special thanks also go Thomas Kaufmann, who is my closest friend since highschool. It was a great coincidence that we chose the same subject at ETH, and since he was always about one year ahead in his PhD he could warn me on what is about to come.

Special thanks go to my girlfriend Veronique Huth who endured me during my stressful finish. It took a lot of patience of her at times when it was hard for me to get my mind off work. With her I could recharge my batteries and I am very thankful that she managed to persuade me to come along on trips even when I felt that the deadline was way to close.

Last but not least I want to thank my parents who stood behind me in all my major life decisions. They taught me to be persistent when I did not succeed on the first try and they gave me the courage to set goals for myself even if I knew they will be hard to achieve. Without their guidance and support I would never have made it to where I am now.
Abstract

A digital signal may represent noisy, quantized, discrete-time observations of a continuous-time signal. If the continuous-time signal is bandlimited to half the sampling rate, and the samples are taken without noise and with infinite precision, the continuous-time signal can be fully reconstructed from its digital samples.

In this thesis we introduce a model of a continuous-time signal that is not bandlimited in the traditional sense, but its spectrum is shaped by the transfer function of a known, finite-order linear system/filter. Among other applications the continuous-time system/filter may represent the analog parts of a receiver, or it may represent a sensor that delivers a filtered version of the actual force applied.

We use Forney factor graphs to represent the continuous-time signal and its noisy, quantized, discrete-time observations. The factor graph approach, which is well developed for discrete-time systems, is extended in this thesis for continuous-time systems. We use message passing algorithms in the factor graph to estimate both, the input and output signal of the continuous-time system/filter at arbitrary time instances. The estimation of the output signal is essentially a Kalman smoother. The estimate of the input signal (which appears to be new) can be viewed as a generalized bandlimited reconstruction of a continuous-time signal.

Two applications are described in this thesis. We show how our model can be extended to increase the quality of a digital signal that was sampled with clock jitter. We also introduce a general view on the widely used sigma-delta converter: an A/D converter may consist of any unstable linear system with feedback. The feedback ensures that the internal
variables of the system stay in a predefined range. We use our proposed model to represent the components of such an “Unstable Linear Filter ADC” and we can then estimate its input signal given quantized, discrete-time observations of some internal variables of the system.
Kurzfassung

Ein digitales Signal kann betrachtet werden als eine Liste von zeitdiskreten, verrauschten und quantisierten Beobachtungen eines zeitkontinuierlichen Signals. Falls das zeitkontinuierliche Signal bandbegrenzt ist, und die Beobachtungen mit einer höheren Rate als die doppelte Bandbreite abgetastet werden, kann das zeitkontinuierliche Signal wieder fehlerfrei rekonstruiert werden.

In dieser Arbeit beschreiben wir ein Modell eines zeitkontinuierlichen Signals welches nicht bandbegrenzt ist, sondern ein Spektrum hat welches geformt ist durch die Übertragungsfunktion eines bekannten, linearen und zeitinvarianten Systems / Filters endlicher Ordnung. Unter anderem könnte das System / Filter den analogen Teil eines Empfängers oder einen Sensor welcher eine gefilterte Version der zu messenden Größe liefert modellieren.


Zum Modell und den Algorithmen beschreiben wir zwei Anwendungen. Wir zeigen, wie das Modell erweitert werden kann für zeitdiskre-
Contents

Abstract vii
Kurzfassung ix

1 Introduction 1
  1.1 Motivation .......................... 1
  1.2 Contributions ........................ 3
  1.3 Outline ............................. 3
  1.4 Notation ............................. 4

2 Factor Graph Model and Estimation Algorithms 5
  2.1 Overview ............................. 5
  2.2 On Factor Graphs ........................ 8
  2.3 On $f(x(t_1)|x(t_0))$ and its Factor Graphs .......... 9
  2.4 Message Passing Through $f(x(t_1)|x(t_0))$ ............ 10
  2.5 Applications .......................... 14
    2.5.1 Simulation of Discrete-Time Samples of $X(t)$ and $Y(t)$ 14
    2.5.2 The SNR of the Noisy Discrete-Time Samples ........ 15
    2.5.3 State Estimation from Discrete-Time Observations 17
    2.5.4 Output Signal Estimation .................. 17
    2.5.5 Input Signal Estimation ........................ 18
  2.6 Vector Input Signals and Vector Observations ........... 19

3 Output Signal Estimation 23
  3.1 Estimation Given Noisy Observations .................. 23
  3.2 Interpolation of a Discrete-Time Signal ................. 25

4 Input Signal Estimation 31
4.1 A Real-World Input Signal .......................... 31
4.2 Direct Estimate of the Input Signal ................. 33
4.3 Additional Spectral Shaping ......................... 38

5 Correction of Clock Jitter 39
   5.1 Overview ............................................ 39
   5.2 The Jitter Error .................................... 40
   5.3 Factor Graph Representation ....................... 41
   5.4 Definitions of SNR .................................. 43
   5.5 The Algorithm ....................................... 44
   5.6 Numerical Results .................................. 45

6 Analog-to-Digital Conversion Using Unstable Linear Filters 49
   6.1 Overview ............................................ 49
   6.2 A Single Integrator .................................. 51
   6.3 A Linear System of Order Two ..................... 53
   6.4 Factor Graph and Estimation Algorithm .......... 56
   6.5 Implementation of an Example System .......... 57
      6.5.1 Generation of the System ..................... 57
      6.5.2 Simulation of the System ..................... 61
      6.5.3 Training the Estimation Algorithm .......... 62
      6.5.4 Performance of the Example System ........ 63

7 Conclusion and Future Work 67

A Connections to Control Theory 69
   A.1 Offline Control ..................................... 69
   A.2 Online Control ..................................... 70

B Additional Proofs 73
   B.1 Proof of the Discrete-Time Decomposition in Figure 2.3 73
   B.2 Derivation of $\Theta(t)$ and $\overline{\Theta}(t)$ in (2.14), (2.16), (2.43) and
      (2.44) .................................................. 76
   B.3 Proof of $\hat{u}(t)$ in Table 2.1 and Table 2.2 .... 77
   B.4 Proof of (2.17) and (2.48) ......................... 78

C The Butterworth Filter 81

D Additions to Chapter 6 83
   D.1 The Sigma-Delta Converter with a Pre-Filter .... 83
Bibliography
Chapter 1

Introduction

1.1 Motivation

A digital signal can be viewed as a list of observations of a continuous-time signal at specific time instances (the sampling instances). In the special case of uniform sampling, if the continuous-time signal is band-limited to half the sampling rate (the Nyquist rate), and the samples are taken without noise and with infinite precision, the continuous-time signal can be reconstructed perfectly [1]. In real-world applications quantization and other noise sources disturb the signal, and thus perfect reconstruction is not possible even if the signal was sampled above the Nyquist rate. Since it is impossible to implement an ideal lowpass filter, and thus to generate a strictly bandlimited signal, even with infinite precision perfect reconstruction would be impossible.

In this thesis a new model of a continuous-time signal is introduced, where a digital signal is not viewed as the discrete-time observations of a strictly bandlimited continuous-time signal. We model a digital signal as the discrete-time observations of a continuous-time signal whose spectrum is shaped by a known, linear system (e.g., a continuous-time linear filter).

Consider some continuous-time linear time-invariant system / filter, which is fed by some input signal $U(t)$. The system output $Y(t)$ is
sampled, and the samples are corrupted by discrete-time additive white Gaussian noise. From the noisy samples $\tilde{Y}_k$ we wish to estimate the noise-free samples $Y_k$, the continuous-time signal $Y(t)$, the state trajectory $X(t)$ of the system/filter, and the input signal $U(t)$ at arbitrary instants $t$.

In this thesis we model such a setup as depicted in Figure 1.1. Since no assumption is made about the spectral distribution of $U(t)$, it is modeled as continuous-time white Gaussian noise. Given this model, the signal $Y(t)$ is not bandlimited in the strict sense required by the sampling theorem [1]; instead, its spectrum is shaped by the transfer function of the continuous-time linear system/filter. We assume that the continuous-time system is known and has a finite-dimensional state space representation.

Applications of this model are ubiquitous. For example, Figure 1.1 might model an analog-to-digital (A/D) converter with a non-ideal anti-aliasing filter and with quantization noise $Z_k$; or, Figure 1.1 might model a sensor that delivers a filtered version of the desired quantity $U(t)$. In both examples, modeling the input signal $U(t)$ as white Gaussian noise amounts to a mere power constraint without any assumptions on the spectral distribution (as will be discussed in Chapter 2).
1.2 Contributions

The main contributions of this work are as follows. We show how a continuous-time signal can be modeled as the output of a finite-order, continuous-time linear time-invariant system/filter that is driven by white Gaussian noise. We also show how a digital signal can be viewed as the noisy, discrete-time observations of such a continuous-time signal.

We represent the model in a Forney factor graph and use message passing algorithms to estimate the continuous-time signal at arbitrary time instances (which is equivalent to Kalman smoothing). The factor graph approach is well developed for discrete-time systems. We show how this approach can be extended for continuous-time systems. In our factor graph representation, in addition to the output signal and the internal variables of the system/filter, also an estimate of the input signal, at arbitrary time instances, is accessible. The estimation of the input signal appears to be new.

Two applications where our proposed model and algorithms can be used are introduced in this thesis. We show how the noise induced by clock jitter can be added to the model by a simple extension, and thus the estimate of the continuous-time signal can be improved. We also show how unstable linear filters can be used in A/D converters, where we will give a more general view on the widely used sigma-delta converter.

1.3 Outline

This thesis is organized as follows. In Chapter 2 we show how continuous-time signals can be viewed as the output signal of a continuous-time system/filter that is driven by white Gaussian noise. We show how this model can be represented in a factor graph and how the input signal, output signal and the state variables of the system/filter can be estimated at arbitrary time instances using message passing algorithms.

In Chapter 3 the estimation of the output signal of the continuous-time system/filter is described in detail and Chapter 4 addresses the estimation of the input signal.

In Chapter 5 we show how not only an estimate of the output signal
itself, but also of its slope can be calculated and how this information can be used to improve the estimate of a continuous-time signal if the discrete-time samples are subject to clock jitter.

In Chapter 6 we show how unstable linear filters can be used in A/D conversion. Feedback is added to the unstable filter to ensure that its internal variables stay in a predefined range. We show how the algorithms introduced in Chapter 2 can be used to estimate the continuous-time input signal.

1.4 Notation

The following notation for matrices and vectors will be used:

- Bold symbols denote matrices and vectors (where we only use column vectors).
- The conjugate complex of some complex value $x$ is denoted as $\overline{x}$.
- The transpose of a matrix $X$ is denoted as $X^T$, and $X^H \triangleq \overline{X}^T$ denotes its Hermitian transpose.
- $\mathcal{N}(m, \sigma^2)$ denotes the Gaussian density function with mean $m$ and variance $\sigma^2$ and $\mathcal{N}(m, V)$ denotes the multivariate Gaussian density function with mean vector $m$ and covariance matrix $V$. 
Chapter 2

Factor Graph Model and Estimation Algorithms

2.1 Overview

In this chapter we describe how the model introduced in Chapter 1 can be represented by factor graphs. Factor graphs [2–4] and similar graphical models [5, 6] allow a unified description of system models and algorithms in different fields. In particular, Gaussian message passing in factor graphs subsumes discrete-time Kalman filtering [7–9] and many variations of it. So far, however, factor graphs have hardly been used for continuous-time signals/systems. In this chapter, we extend the factor graph approach to models as in Figure 1.1. The associated continuous-time signals then become computational objects that can be handled with arbitrary temporal resolution by Gaussian message passing. In consequence, the factor graph approach can be used, e.g., for signal interpolation (Chapter 3), or for the digital enhancement of analog circuits (Chapter 5 and Chapter 6) [10, 11]. Parts of this chapter are published in [12].

More specifically, let $X \in \mathbb{R}^n$ be the state of a linear system (e.g., as
in Figure 1.1) that evolves in time according to

$$\dot{X}(t) = AX(t) + bU(t)$$  \hfill (2.1)$$

where $\dot{X}$ denotes the derivative with respect to time. The real square matrix $A \in \mathbb{R}^{n \times n}$ and the real vector $b \in \mathbb{R}^n$ are assumed to be fixed and known.

The output of the system is the discrete-time signal $Y_1, Y_2, \ldots \in \mathbb{R}$ with

$$Y_k = c^T X(t_k)$$  \hfill (2.2)$$

where $t_1, t_2, \ldots$ (with $t_k < t_{k+1}$) are known sampling instances. The vector $c \in \mathbb{R}^n$ is assumed to be known as well. Usually, a noisy version of $Y_k$ is observed, denoted as $\tilde{Y}_k$:

$$\tilde{Y}_k = Y_k + Z_k,$$  \hfill (2.3)$$

where $Z_k \in \mathbb{R}$ are independent Gaussian random variables with mean zero and variance $\sigma^2_Z$.

We model the real and scalar input signal $U(t)$ as continuous-time white Gaussian noise, i.e., for $t < t'$, the integral

$$\int_t^{t'} U(\tau) \, d\tau$$  \hfill (2.4)$$

is a zero-mean Gaussian random variable with variance $\sigma^2_U(t' - t)$, and any number of such integrals are independent random variables provided that the corresponding integration intervals are disjoint. In consequence, it is appropriate to replace (2.1) by

$$dX(t) = AX(t) \, dt + bU(t) \, dt$$  \hfill (2.5)$$

where the undefined instantaneous value of $U(t)$ appears only in the infinitesimal value $U(t) \, dt$.

Modeling $U(t)$ as continuous-time white Gaussian noise induces, for any fixed initial state $X(t_0) = x(t_0)$, a probability density $f(x(t_1)|x(t_0))$ over the possible values of $X(t_1)$, which (unsurprisingly) turns out to be Gaussian.
The function $f(x(t_1)|x(t_0))$ (considered as a function defined on $\mathbb{R}^n \times \mathbb{R}^n$, parametrized by $t_1 - t_0$) may be used as a building block in factor graphs (Section 2.3). Discrete-time linear Gaussian factor graphs as in [4] are thus extended to include exact models of continuous-time systems of the form (2.1) (between arbitrary discrete points in time). In consequence, we immediately obtain efficient algorithms for a variety of problems. The underlying continuous-time system does not have to be time invariant: time invariance is required only between known discrete points in time.

The following applications will be addressed:

- System simulation, i.e., the generation of discrete-time samples of $X(t_k), Y_k$ and $\tilde{Y}_k$ for arbitrary $t_1, t_2, t_3, \ldots$ (Section 2.5.1 and Section 2.5.2)

- MAP / MMSE / LMMSE estimation\(^1\) of the state vector $X(t_k)$ for arbitrary instants $t_1, t_2, \ldots$ from noisy discrete-time observations of $Y(t) = c^T X(t)$ (for some known vector $c$) at arbitrary sampling times (Section 2.5.3).

- Interpolation of sampled filter output $Y(t) = c^T X(t)$ (for known $c$) from and to arbitrary discrete-time sampling points (Section 2.5.4 and Chapter 3).

- MAP / MMSE / LMMSE estimation of the input signal $U(t)$ for arbitrary instants $t$ from noisy discrete-time observations. It will be suggested that this estimate of the input signal may be viewed as a generalization of a bandlimited estimate of $U(t)$ that is suited for real filters (Chapter 4).

The algorithm for the estimation of $X(t)$ and $Y(t)$ is essentially a Kalman smoother represented in factor graph language. Factor graphs and message passing algorithms allow efficient implementation of Kalman smoothers [4, 13].

Now Kalman filtering, which originated in control theory [7, 8], has long been a standard tool in signal processing [14,15] and statistics [16].

\(^1\)Maximum a posteriori (MAP) estimation, minimum mean squared error (MMSE) estimation, and linear (or affine) minimum mean squared error (LMMSE) estimation coincide for linear Gaussian models, cf., e.g., [4, Section V].
However, estimation of the input signal $U(t)$ does not seem to have been addressed in the Kalman filtering literature. Moreover, while continuous-time models with discrete-time observations (as in Figure 1.1) are standard\footnote{But note that the Kalman-Bucy filter \cite{9} addresses the different situation where the observations are also continuous-time signals.} in control theory, they seem to have been somewhat neglected in digital signal processing. It turns out, indeed, that the algorithms introduced in this chapter can easily be put in the form of simple control problems as in \cite{17} (cf. Appendix A).

### 2.2 On Factor Graphs

Throughout this thesis, Forney factor graphs (normal factor graphs \cite{18} as in \cite{3} and \cite{4}) will be used. The edges in the factor graph represent variables and the nodes/boxes represent factors.

In this notation, the system model of Figure 1.1 may be represented by the factor graph shown in Figure 2.1. More precisely, Figure 2.1 represents the joint probability density of the variables in the system model given by the function $f(\mathbf{x}(t_1)|\mathbf{x}(t_0))$ of Section 2.1 together with the observations $\tilde{Y}_k \in \mathbb{R}$ as defined in (2.3). For details of this factor graph notation cf. \cite{4}.

Note that:

a) Figure 2.1 shows only a section (from $t_k$ to $t_{k+1}$) of the factor graph; the complete factor graph starts at time $t_0$ and ends at some time $t_K$, and it may contain additional nodes to represent any pertinent initial or final conditions.

b) Apart from the Gaussian nodes/factors (i.e., $f(\mathbf{x}(t_1)|\mathbf{x}(t_0))$ and $\mathcal{N}(0,\sigma^2_Z)$), the nodes/boxes in Figure 2.1 represent linear constraints.

c) All the edges in this factor graph are directed, which allows us to refer to the forward message $\mu_X$ and to the backward message $\overleftarrow{\mu}_X$ along the edge representing some variable $X$. 


2.3. On \( f(x(t_1)|x(t_0)) \) and its Factor Graphs

Assume some smooth input signal \( u(t) \), integrating (2.1) from \( t = t_0 \) to \( t_1 \) (with \( T \triangleq t_1 - t_0 \geq 0 \)) yields

\[
x(t_1) = e^{AT} x(t_0) + \int_0^T e^{A(T - \tau)} bu(t_0 + \tau) d\tau \\
= e^{AT} x(t_0) + \sum_{k=0}^{N-1} \int_{kT/N}^{(k+1)T/N} e^{A(T - \tau)} bu(t_0 + \tau) d\tau
\]

Figure 2.1: A Forney factor graph of the system in Figure 1.1.

Moreover, refer to the discussion in [4] of message passing in linear Gaussian models that immediately applies to Figure 2.1. Gaussian messages will be parametrized by a mean vector \( m \) and a covariance matrix \( V \) (or the inverse covariance matrix \( W \)), or mean \( m \) and variance \( V \) in the scalar case. We will use the symbols \( \vec{m}_X \) and \( \vec{V}_X \) (or \( \vec{m}_X \) and \( \vec{V}_X \)) to denote the parameters of the forward message (along some edge/variable \( X \) or \( X \)) and \( \hat{m}_X \) and \( \hat{V}_X \) (or \( \hat{m}_X \) and \( \hat{V}_X \)) for the parameters of the backward message.
\[ e^{AT} \mathbf{x}(t_0) + \sum_{k=0}^{N-1} e^{A(T-kT/N)} \mathbf{b} \int_{kT/N}^{(k+1)T/N} u(t_0 + \tau) d\tau, \quad (2.8) \]

where (2.7)–(2.8) become exact for \( N \to \infty \).

If \( u(t) \) is white Gaussian noise as described in Section 2.1, and thus a random signal \( U(t) \), the integrals in (2.8) are independent Gaussian random variables with mean zero and variance \( \sigma^2_U T/N \) and, for \( N \to \infty \), the sum in (2.8) results in a zero-mean Gaussian random vector with covariance matrix \( \mathbf{V}_S \) [19, 20]

\[ \mathbf{V}_S = \sigma^2_U \int_0^T e^{A(T-\tau)} \mathbf{b} \mathbf{b}^T (e^{A(T-\tau)})^T d\tau \quad (2.9) \]

\[ = \sigma^2_U \int_0^T e^{A\tau} \mathbf{b} \mathbf{b}^T (e^{A\tau})^T d\tau. \quad (2.10) \]

It is thus clear that, for fixed \( \mathbf{x}(t_0) \), \( \mathbf{x}(t_1) \) is a Gaussian random vector \( \mathbf{X}(t_1) \) with mean \( e^{AT} \mathbf{x}(t_0) \) and covariance matrix \( \mathbf{V}_S \), i.e.,

\[ f(\mathbf{x}(t_1)|\mathbf{x}(t_0)) \propto e^{(\mathbf{x}(t_1)-e^{AT}\mathbf{x}(t_0))^T \mathbf{V}_S^{-1} (\mathbf{x}(t_1)-e^{AT}\mathbf{x}(t_0)) \quad (2.11) \]

where “\( \propto \)” denotes equality up to a constant scale factor.

The function (2.11) may immediately be used as a node in the factor graph in Figure 2.1. Nevertheless, it will also be useful to represent the function (2.11) by nontrivial factor graphs. A first such factor graph is shown in Figure 2.2.

A second factor graph for \( f(\mathbf{x}(t_1)|\mathbf{x}(t_0)) \) is shown in Figure 2.3, where a similar decomposition into \( N \) discrete steps as in (2.6) is illustrated. This factor graph is only an approximate representation of \( f(\mathbf{x}(t_1)|\mathbf{x}(t_0)) \), but the representation becomes exact in the limit \( N \to \infty \); the proof is given in Appendix B.1. Note that, in Figure 2.3, snapshots of the continuous-time input signal \( U(t) \) are represented explicitly, which we will use for the estimation of the input signal.

### 2.4 Message Passing Through \( f(\mathbf{x}(t_1)|\mathbf{x}(t_0)) \)

Now consider sum-product message passing through a node representing the function \( f(\mathbf{x}(t_1)|\mathbf{x}(t_0)) \) as discussed in Section 2.3. Assuming
2.4. Message Passing Through $f(x(t_1)|x(t_0))$

Figure 2.2: A factor graph of $f(x(t_{k+1})|x(t_k))$ according to (2.6)–(2.11).

Figure 2.3: Decomposition of the node / factor $f(x(t_1)|x(t_0))$ into $N$ discrete time steps. This representation is exact only in the limit $N \to \infty$. 
Table 2.1: Messages through node / factor $f(x(t_1)|x(t_0))$ with $t_1 > t_0$. 

![Diagram of factor graph model](image)

\[
\begin{align*}
\overrightarrow{m}_X(t_1) &= e^{A(t_1-t_0)} \overrightarrow{m}_X(t_0) \\
\overleftarrow{V}_X(t_1) &= e^{A(t_1-t_0)} \overleftarrow{V}_X(t_0) e^{A^T(t_1-t_0)} \\
&+ \sigma_u^2 \int_0^{t_1-t_0} e^{A^T\Sigma} b^T e^{A^T} d\tau \\
&\quad \{Q \Theta(t_1-t_0)Q^U \text{ see (2.13)}\} \\
\hat{u}(t) &= \sigma_u^2 b^T \left( \overleftarrow{V}_X(t) + \overrightarrow{V}_X(t) \right)^{-1} \left( \overrightarrow{m}_X(t) - \overleftarrow{m}_X(t) \right)
\end{align*}
\]
that the incoming messages to such a node are Gaussian, the outgoing
messages are Gaussian as well [3] with parameters as given in Table 2.1.

The computation of the forward message (with mean vector (ii.i.1)
and covariance matrix (ii.i.2)) is obvious from Section 2.3. Both the
forward message and the backward message (with parameters (ii.i.3)
and (ii.i.4)) are easily obtained from Figure 2.2, (2.10) and (2.11), and
Tables 2 and 3 of [4].

If the matrix \( A \) is diagonalizable, then the integrals in (ii.i.2) and
(ii.i.4) can easily be expressed in closed form. Specifically, if

\[
A = Q \begin{pmatrix}
\lambda_1 & 0 \\
. & . \\
0 & \lambda_n 
\end{pmatrix} Q^{-1}
\]  

(2.12)

for some complex square matrix \( Q \), then

\[
\sigma^2_U \int_0^t e^{A \tau} b b^T e^{A^T \tau} d\tau = Q \Theta(t) Q^H
\]

(2.13)

where the elements of the square matrix \( \Theta(t) \) are given by

\[
\Theta(t)_{k,\ell} \triangleq \sigma^2_U \frac{(Q^{-1} b)_k \overline{(Q^{-1} b)_\ell}}{\lambda_k + \lambda_\ell} \left( e^{(\lambda_k + \lambda_\ell)t} - 1 \right),
\]

(2.14)

and

\[
\sigma^2_U \int_0^t e^{-A \tau} b b^T e^{-A^T \tau} d\tau = Q \Theta(t) Q^H
\]

(2.15)

with

\[
\Theta(t)_{k,\ell} \triangleq \sigma^2_U \frac{(Q^{-1} b)_k \overline{(Q^{-1} b)_\ell}}{\lambda_k + \lambda_\ell} \left( 1 - e^{-(\lambda_k + \lambda_\ell)t} \right),
\]

(2.16)

where \( (Q^{-1} b)_k \) denotes the \( k \)-th element of the vector \( Q^{-1} b \).

The proof of (2.13) and (2.15) is given in Appendix B.2.

Of particular interest in Table 2.1 is (ii.i.5): if the overall factor graph
is linear, Gaussian, and cycle free (as, e.g., in Figure 2.1), then \( \hat{u}(t) \triangleq \)
$m_{U(t)}$ and $\hat{x}(t) \triangleq m_{X(t)}$ are the MAP / MMSE / LMMSE estimate of $U(t)$ and $Y(t)$ \cite{4}. If the factor graph in Figure 2.1 consists of $K$ sections between $t_0$ and $t_K$ (with observations starting at $t_1$), then the estimate pair $\hat{u}(t)$ and $\hat{x}(t_k)$ is the solution that minimizes the cost function

\[
\frac{1}{\sigma_U^2} \int_{t_0}^{t_K} \hat{u}(t)^2 \, dt + \frac{1}{\sigma_Z^2} \sum_{k=1}^{K} (\hat{y}_k - c^T \hat{x}(t_k))^2. \tag{2.17}
\]

The proof of (ii.i.5) is given in Appendix B.3, where we also show that the variance of the estimate $\hat{u}(t)$ is infinite (which reminds us that continuous-time white Gaussian noise cannot be estimated). However, the estimate $\hat{u}(t)$ as in (ii.i.5) can be a highly useful estimate as will be discussed in Chapter 4. The cost function (2.17) shows that $\sigma_U^2$ is merely a constraint on the energy of the estimate $\hat{u}(t)$ (the proof of (2.17) is given in Appendix B.4). If $\sigma_U^2/\sigma_Z^2$ is set properly, the estimates $\hat{u}(t)$ and $\hat{y}(t) \triangleq c^T \hat{x}(t)$ will have a spectrum shaped by the continuous-time system / filter, which will also be discussed in Chapter 3.

## 2.5 Applications

### 2.5.1 Simulation of Discrete-Time Samples of $X(t)$ and $Y(t)$

Given a model as in Figure 1.1 with given state space matrices of the system / filter, a sample trajectory of $x(t_k)$ can be generated. From the state-trajectory the corresponding output signal $y(t_k)$ can be generated as well. If $t_1 < t_2 < t_3 < \ldots \ldots$, samples $x(t_1), x(t_2), x(t_3), \ldots$ of $X(t)$ can be obtained as follows. Assume some known $X(t_k) = x(t_k)$. Conditioned on $X(t_k) = x(t_k)$, $X(t_{k+1})$ is Gaussian with mean $\overrightarrow{m}_{X(t_{k+1})}$ and covariance matrix $\overrightarrow{V}_{X(t_{k+1})}$ computed according to (ii.i.1) and (ii.i.2) with $\overrightarrow{m}_{X(t_k)} = x(t_k)$ and $\overrightarrow{V}_{X(t_k)} = 0$. From this distribution, we can draw a sample $x(t_{k+1})$ and then use $X(t_{k+1}) = x(t_{k+1})$ to generate a sample of $X(t_{k+2})$, etc.

Samples of $Y(t) = c^T X(t)$ are easily obtained as $y(t_k) = c^T x(t_k)$. Thus, we can simulate discrete-time samples of filtered continuous-time white Gaussian noise.
Figure 2.4: Example output signal of a 4-th order continuous-time Butterworth filter with different $-3$ dB-frequencies $f_c$.

See Figure 2.4 for an example plot of samples of the output signals of 4-th order Butterworth filters (cf. Appendix C for a brief introduction to Butterworth filters). The output signals of filters with different $-3$ dB-frequency $f_c$ were generated. As seen in the figure, the signal energy is concentrated around frequencies smaller than $f_c$ of the Butterworth filter, i.e., the spectrum of the simulated signals are shaped by the transfer function of the continuous-time filter.

2.5.2 The SNR of the Noisy Discrete-Time Samples

We define the SNR of the noisy samples $\tilde{y}_k$ as

$$\text{SNR}_{\tilde{y}} \triangleq \frac{\mathbb{E}[Y_k^2]}{\sigma_Z^2},$$

which is a ratio of the expected value of the signal squared to the variance of the noise. We now describe how this quantity can be calculated from the state space parameters and a given ratio $\sigma_U^2 / \sigma_Z^2$.

Let the system be stable, i.e., all eigenvalues of $A$ (the poles of the filter) have a negative real part. If a state trajectory $X(t_k)$ is generated as described in Section 2.5.1, and if any prior conditions can be neglected, then the state space vector $X(t)$ is a Gaussian random variable with mean vector

$$\mathbb{E}[X(t)] = 0$$

and covariance matrix

$$\text{Cov}(X(t)) = \Sigma(t)$$
and covariance matrix
\[ E\left[ X(t) (X(t))^\top \right] = \vec{V}_{X(\infty)} \] (2.20)
with
\[ \vec{V}_{X(\infty)} \triangleq Q \lim_{t \to \infty} \vec{\Theta}(t) Q^H, \] (2.21)
where \( \vec{\Theta}(t) \) is defined as in (2.14). The message \( \vec{\mu}_{X(\infty)} \) with mean \( \vec{m}_{X(\infty)} = 0 \) and covariance matrix \( \vec{V}_{X(\infty)} \) is called the “steady message”.

The expected power of \( Y(t) \) can now be calculated:
\[ E[Y(t)^2] = c^\top E\left[ X(t) (X(t))^\top \right] c \] (2.22)
\[ = c^\top \vec{V}_{X(\infty)} c, \] (2.23)
which, with (2.18), directly implies that
\[ \text{SNR}_{\tilde{y}} = \frac{E[Y_k^2]}{\sigma_Z^2} \] (2.24)
\[ = \frac{c^\top \vec{V}_{X(\infty)} c}{\sigma_Z^2}. \] (2.25)

Note that \( \vec{V}_{X(\infty)} \) is proportional to \( \sigma_U^2 \) (cf. (2.14) and (2.21)). Thus, we can now generate noisy samples \( \tilde{y}_k \) from samples \( y(t_k) \) (generated as described in Section 2.5.1) with a specific SNR by calculating the matching \( \sigma_Z^2 \), given a filter and \( \sigma_U^2 \).

For a Butterworth filter the SNR of the noisy samples turns out to be
\[ \text{SNR} \approx \frac{\sigma_U^2}{\sigma_Z^2} f_c \cdot 2.052, \] (2.26)
for the 4th-order filter, and
\[ \text{SNR} \approx \frac{\sigma_U^2}{\sigma_Z^2} f_c \cdot 2.023. \] (2.27)
for the 6th-order filter.

Throughout this thesis, SNR will be given in dB, i.e., \( 10 \cdot \log_{10} (\text{SNR}) \).
2.5. Applications

\[ X(t_k) \rightarrow f(x'(t)|x(t_k)) \rightarrow f(x(t_{k+1})|x(t')) \rightarrow X(t_{k+1}) \]

**Figure 2.5:** Splitting the node / factor \( f(x(t_{k+1})|x(t_k)) \) to access the state at an intermediate point in time \( t' \).

### 2.5.3 State Estimation from Discrete-Time Observations

The MAP / MMSE / LMMSE estimates of \( X(t_k) \) based on discrete-time observations \( \tilde{y}_1, \tilde{y}_2, \tilde{y}_3, \ldots \) may be obtained by forward-backward Gaussian message passing in the factor graph of Figure 2.1 (of course, this is just Kalman smoothing [7] expressed in factor graph language [4, 13]).

If all messages \( \overrightarrow{\mu}_{X(t_k)} \) and \( \overleftarrow{\mu}_{X(t_k)} \) in the factor graph in Figure 2.1 are calculated, the MAP / MMSE / LMMSE estimate \( \hat{x}(t_k) \triangleq m_{X(t_k)} \) (cf. Section 2.4) can be calculated.

Splitting the node / factor \( f(x(t_{k+1})|x(t_k)) \) as shown in Figure 2.5, allows such estimates also to be obtained for times \( t' \) between observations. This way, arbitrary temporal resolution may be achieved.

### 2.5.4 Output Signal Estimation

It is straightforward to extend state estimation from the previous section to estimating the noise-free output \( Y(t) = c^T X(t) \) at any fixed time \( t \). The estimate \( \hat{y}(t) \) of \( Y(t) \) is

\[
\hat{y}(t) = c^T \hat{x}(t).
\]

(2.28)

For any stochastic process \( Y(t) \) representable as filtered white Gaussian noise, we may thus obtain estimates of \( Y(t) \) at arbitrary points in time based on observations \( \tilde{y}_k \) at discrete times \( t_k \). The energy of the estimate, especially the energy of frequency components where the damping
of the linear system/filter is large, can be controlled by the input power constraint \( \sigma_U^2 \). This was mentioned in Section 2.4, and will be discussed in more detail in Chapter 3.

To ensure that the estimate of \( Y(t) \) is a smooth signal, i.e., \( \frac{d}{dt} \hat{y}(t) \) does not contain any jumps, the linear system/filter needs to fulfill some precondition. The derivative of

\[
Y(t) = c^T X(t)
\]

(2.29)
can be expressed using the state space equation (2.1):

\[
\dot{Y}(t) = c^T \dot{X}(t)
\]

(2.30)

\[
= c^T (AX(t) + bU(t))
\]

(2.31)

\[
= c^T AX(t) + c^T bU(t).
\]

(2.32)

Since \( U(t) \) is modeled as continuous-time white Gaussian noise, and thus \( U(t) \) will contain jumps, \( c^T b = 0 \) implies that \( Y(t) \) also has jumps.

### 2.5.5 Input Signal Estimation

Using (ii.i.5), an estimate of the input signal \( U(t) \) at arbitrary points in time can be calculated as well. This does not seem to have been proposed before. As described in Section 2.4 the estimate \( \hat{u}(t) \) is a regularized MMSE estimate where \( \sigma_U^2 / \sigma_Z^2 \) acts as a power constraint on the input signal \( U(t) \). Similarly to the estimate of the output signal in Section 2.5.4, some preconditions on the state space parameters need to be fulfilled to ensure that the estimate \( \hat{u}(t) \) is smooth.

We did not find an analytical solution to this problem, but from our simulations, it seems that the second derivative of \( Y(t) \)

\[
\ddot{Y}(t) = c^T \ddot{X}(t)
\]

(2.33)

\[
= c^T (A \dot{X}(t) + \dot{bU}(t))
\]

(2.34)

\[
= c^T A^2 X(t) + c^T AbU(t) + c^T \dot{bU}(t)
\]

(2.35)

needs to be independent of \( U(t) \) (and thus \( \dot{U}(t) \)). Our simulations yielded a smooth estimate of the input signal if both

\[
c^T Ab = 0
\]

(2.36)

\[
c^T b = 0.
\]

(2.37)
2.6 Vector Input Signals and Vector Observations

So far, only a scalar input signal $U(t) \in \mathbb{R}$ and scalar observations $\tilde{Y}_k \in \mathbb{R}$ were addressed. In this section, the model and message passing rules are extended for a system with a non-scalar input signal and non-scalar observations.

A special case where it can be beneficial to model the input signal $U(t)$ as a vector is, if thermal noise that directly influences the states of the system, shall be included in the model. The first element of the vector $U(t)$ may then represent the actual input signal of the filter and separate inputs that model the thermal noise sources can be added to the vector $U(t)$.

The state space equation (2.1) for a system with an input signal $U(t) \in \mathbb{R}^\eta$ changes to

$$\dot{X}(t) = AX(t) + BU(t) \quad (2.38)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times \eta}$ and $X(t) \in \mathbb{R}^n$.

Lets now model the input signal $U(t)$ as multivariate continuous-time white Gaussian noise. As in (2.4), this means that for $t < t'$, the integral

$$\int_t^{t'} U(\tau) d\tau \quad (2.39)$$

is a zero-mean multivariate Gaussian random variable with covariance matrix $V_U(t' - t)$, where we assume $V_U$ to be known. Any number of integrals as in (2.39) are independent random variables provided that the corresponding integration intervals are disjoint.

As in (2.6)–(2.8), assuming that every component of the input signal $U(t)$ is smooth, integrating (2.38) from $t = t_0$ to $t_1$ (with $T \triangleq t_1 - t_0 > 0$) yields

$$X(t_1) \approx e^{AT} X(t_0)$$

$$+ \sum_{k=1}^{N-1} e^{A(T-kT/N)} B \int_{kT/N}^{(k+1)T/N} U(t_0 + \tau) d\tau, \quad (2.40)$$
where (2.40) is exact for $N \to \infty$.

If $U(t)$ is white Gaussian noise, the integrals in (2.40) are independent Gaussian random vectors with zero-mean and covariance matrix $V_UT/N$. Thus, for $N \to \infty$, the sum in (2.40) results in a zero-mean Gaussian random vector with covariance matrix

$$V_S = \int_0^T e^{A\tau} BVU \, B^T e^{-A\tau} \, d\tau, \quad (2.41)$$

where $T = t_1 - t_0$.

Similarly to the scalar case, $X(t_1)$, for fixed $X(t_0) = x(t_0)$, $X(t_1)$, is a Gaussian random vector with mean $e^{AT}x(t_0)$ and covariance matrix $V_S$, i.e.,

$$f(x(t_1)|x(t_0)) \propto e^{(x(t_1)-e^{AT}x(t_0))^T V_S^{-1} (x(t_1)-e^{AT}x(t_0))} \quad (2.42)$$

where “$\propto$” denotes equality up to a constant scale factor.

The message passing rules through node / factor $f(x(t_0)|x(t_1))$ as in Table 2.1 for a vector input directly follow and are shown in Table 2.2.

The matrix $Q$ and the eigenvalues $\lambda_i$ of $A$ are defined as in (2.12) and the square matrices $\Theta(t)$ and $\bar{\Theta}(t)$ are now defined as

$$\Theta(t)_{k,\ell} \triangleq \frac{\psi_{k,\ell}}{\lambda_k + \lambda_\ell} \left(e^{(\lambda_k + \lambda_\ell)t} - 1\right), \quad (2.43)$$

$$\bar{\Theta}(t)_{k,\ell} \triangleq \frac{\bar{\psi}_{k,\ell}}{\lambda_k + \lambda_\ell} \left(1 - e^{-(\lambda_k + \lambda_\ell)t}\right), \quad (2.44)$$

where $\psi_{k,\ell}$ denotes the elements of the matrix

$$\Psi \triangleq Q^{-1} BVU \left(Q^{-1} B\right)^H. \quad (2.45)$$

Not only the input signal, but also the observations can be vectors. If $Y(t) \in \mathbb{R}^\nu$ is a vector, (2.2) changes to

$$Y_k = C^T X(t_k) \quad (2.46)$$

where $Y_k \in \mathbb{R}^\nu$, $C \in \mathbb{R}^{n \times \nu}$, and the noisy observations $\tilde{Y}_k$ as in (2.3) are

$$\tilde{Y}_k = Y_k + Z_k. \quad (2.47)$$
Table 2.2: Messages through node / factor $f(x(t_1)|x(t_0))$ with $t_1 > t_0$ and a non-scalar input signal $U(t)$.

\[
\begin{align*}
\dot{m}_X(t_1) &= e^{A(t_1-t_0)}\dot{m}_X(t_0) \\
\dot{V}_X(t_1) &= e^{A(t_1-t_0)}\dot{V}_X(t_0)e^{AT(t_1-t_0)} \\
&\quad + \int_0^{t_1-t_0} e^{AT\tau}BV_UB^Te^{AT\tau}d\tau \quad \text{see (2.43)} \\
\dot{m}_X(t_0) &= e^{-A(t_1-t_0)}\dot{m}_X(t_1) \\
\dot{V}_X(t_0) &= e^{-A(t_1-t_0)}\dot{V}_X(t_1)e^{-AT(t_1-t_0)} \\
&\quad + \int_0^{t_1-t_0} e^{-AT\tau}BV_UB^Te^{-AT\tau}d\tau \quad \text{see (2.44)} \\
\hat{u}(t) &= V_UB^T\left(V_X(t) + \dot{V}_X(t)\right)^{-1}(\dot{m}_X(t) - \dot{m}_X(t)) \\
&\quad \text{(ii.ii.5)} \\
&\quad \text{(ii.ii.6)}
\end{align*}
\]
Figure 2.6: A Forney factor graph of the system in Figure 1.1 with non-scalar observations.

The random variables \( Z_k \in \mathbb{R}^\nu \) are independent Gaussian vectors with mean zero and covariance matrix \( V_Z \).

The factor graph in Figure 2.1, is easily adapted for the estimation of \( Y(t) \), \( X(t) \) and \( U(t) \) and depicted in Figure 2.6. If this factor graph consists of \( K \) sections between \( t_0 \) and \( t_K \) (with observations starting at \( t_1 \)), the estimates \( \hat{u}(t_k) \triangleq m_U(t) \) and \( \hat{x}_k \triangleq m_X(t_k) \), both calculated by sum-product message passing, minimize the cost function

\[
\int_{t_0}^{t_k} \hat{u}(t)V_U^{-1}((\hat{u}(t))^T dt \\
+ \sum_{k=1}^{K} (\tilde{y}_k - C^T \hat{x}(t_k))V_Z^{-1}(\tilde{y}_k - C^T \hat{x}(t_k))^T.
\] (2.48)

The proof of (2.48) is given in Appendix B.3.
Chapter 3

Output Signal Estimation

3.1 Estimation Given Noisy Observations

In this chapter the estimation of the output signal $Y(t)$ of a system as depicted in Figure 1.1 is discussed. If an estimate $\hat{y}(t)$ of the output signal $Y(t)$ is calculated using the algorithm described in Chapter 2, the cost function

$$\frac{1}{\sigma_U^2} \int_{t_0}^{t_K} \hat{u}(t)^2 \, dt + \frac{1}{\sigma_Z^2} \sum_{k=1}^{K} (\tilde{y}_k - c^T \hat{x}(t_k))^2 \tag{3.1}$$

(see also (2.17)) is minimized. This cost function consists of an energy constraint on the estimate $\hat{u}(t)$ of the input signal $U(t)$ (first part of the cost function), and a constraint on the difference between the estimate $\hat{y}(t_k) = c^T \hat{x}(t_k)$ and the observed sample $\tilde{y}_k$ (second part of the cost function). The regularization parameter $\sigma_U^2 / \sigma_Z^2$ weights the two constraints. If the value of this parameter is very large, there is only little penalty on the energy of the estimate of $U(t)$ (and also on the estimate of $Y(t)$), and $\hat{y}(t_k)$ will be very close to the observed samples, i.e., $\hat{y}(t_k) \approx \tilde{y}_k$. If the value of $\sigma_U^2 / \sigma_Z^2$ is very small, the algorithm will find a signal that contains only very little energy but allows a large difference between the estimate $\hat{y}(t_k)$ and the observation $\tilde{y}_k$ (where in the extreme case of $\sigma_U^2 = 0$, $\hat{u}(t) = \hat{y}(t) = 0$).
Chapter 3. Output Signal Estimation

![Figure 3.1](image-url)

Figure 3.1: Estimation of the output signal $Y(t)$ from noisy samples $\tilde{y}_k$ (fat dots), which were generated with SNR $\tilde{\text{SNR}}_\tilde{y} = 5$ dB. Solid line: estimate of $Y(t)$ at correct SNR $\tilde{\text{SNR}}_\tilde{y} = 5$ dB. Dotted line: estimation with assumed SNR $\tilde{\text{SNR}}_\tilde{y} = 35$ dB; dashed line: estimation with assumed SNR $\tilde{\text{SNR}}_\tilde{y} = -2$ dB.

As described in Section 2.5.2, the SNR of the noisy samples $\tilde{y}_k$ is proportional to the regularization parameter $\sigma_U^2/\sigma_Z^2$. Hence, changing the value of $\sigma_U^2/\sigma_Z^2$ is equivalent to changing SNR $\tilde{\text{SNR}}_\tilde{y}$ that is assumed by the estimation algorithm.

Figure 3.1 shows a numerical example on the effect of changing SNR $\tilde{\text{SNR}}_\tilde{y}$ that is assumed by the algorithm. Noisy samples $\tilde{y}_k$ (fat dots) were generated as described in Section 2.5.2, using a 4-th order Butterworth filter (cf. Appendix C for a brief introduction to Butterworth filters) with $-3$ dB-frequency $f_c = 0.1$ Hz and SNR $\tilde{\text{SNR}}_\tilde{y} = 5$ dB. The signal $Y(t)$ was estimated using the Kalman smoother as described in Section 2.5.4, where SNR $\tilde{\text{SNR}}_\tilde{y}$ that was assumed by the algorithm was set to $-2$ dB (dashed line), 5 dB (solid line) and 35 dB (dotted line).

As expected, choosing a large value for SNR $\tilde{\text{SNR}}_\tilde{y}$ results in overfitting and choosing a small value results in an estimate that contains only little energy. Choosing the correct value for SNR $\tilde{\text{SNR}}_\tilde{y}$ results in an estimate where high frequency components (which are damped by the Butterworth filter) are eliminated, but components with frequencies smaller than $f_c$ are still present.

If the sampling rate $f_s$ is increased, the signal $Y(t)$ can be estimated
3.2. Interpolation of a Discrete-Time Signal

with a smaller estimation error [21]. The effect on the estimation error of increasing the sampling rate is depicted in Figure 3.2. Samples were generated using two different Butterworth filters where the filter order was \( N = 4 \) (solid lines) and \( N = 8 \) (dashed lines). Samples \( \tilde{y}_k \) where generated with three different SNR\( \tilde{y} \): \(-10 \) dB, 0 dB and 10 dB.

We define the SNR of the estimate as

\[
\text{SNR}_{\tilde{y}} \triangleq \frac{\mathbb{E}[Y^2_k]}{\mathbb{E}[(\tilde{Y}_k - Y_k)^2]}.
\] (3.2)

In Figure 3.2 it can clearly be seen how a larger sampling rate \( f_s \) increases SNR\( \tilde{y} \) and thus, decreases the estimation error.

The signal that was generated with the Butterworth filter or order \( N = 8 \) can be reconstructed with a higher SNR\( \tilde{y} \) than with \( N = 4 \). A higher order Butterworth filter has higher damping for frequencies larger than \( f_c \), and thus the energy of the signal \( Y(t) \) is more concentrated in frequencies smaller than \( f_c \). This allows better reconstruction of the signal \( Y(t) \) if the noise \( Z_k \) is white Gaussian noise, i.e., its energy is evenly spread over the whole spectrum. If \( Y(t) \) was strictly bandlimited, optimal linear filtering results in the SNR\( \tilde{y} \), which is plotted as the dotted line in Figure 3.2 (cf. [22, Section V.D.1 Noncausal Wiener-Kolmogorov Filtering]). Recall that the Kalman smoother is a linear filter. For the solid and the dashed lines in Figure 3.2 the same model that was used for signal generation was also used for the signal estimation. Since all noise sources are Gaussian this implies that the Kalman smoother is the optimal estimator [7].

3.2 Interpolation of a Discrete-Time Signal

Using the Kalman smoother as described in Section 2.5.3, an estimate \( \hat{x}(t) \) of the state \( X(t) \) can be calculated at arbitrary time instances \( t \), and thus it is possible to interpolate a discrete-time signal at arbitrary time instances by calculating \( \hat{y}(t) = c^T \hat{x}(t) \).

Let \( \tilde{y}_k \) be a discrete-time signal that was sampled with rate \( f_s \) from a continuous-time signal \( y(t) \). Under the assumptions that \( y(t) \) is strictly
Figure 3.2: Empirical estimation error (3.2) vs. normalized sampling frequency $f_s/f_c$, parametrized by $\text{SNR}_y$ (2.24), for a Butterworth filter of order 4 and order 8 compared to the optimal SNR for linear filtering of strictly bandlimited signals. The values are averaged over 10 simulations of signals with a length of 10 000 samples.
Figure 3.3: Example of a signal interpolation.
bandlimited to $f_c/2$ and no noise was added in the sampling process, i.e., $\tilde{y} = y(t_k)$, it is possible to perfectly reconstruct $y(t)$ at arbitrary time instances in theory [1]. Since it is not possible to generate a finite-dimensional state space model that represents an ideal lowpass filter, perfect reconstruction cannot be expected. This is illustrated in Figure 3.3, where the signal

$$y(t) = \sin(2\pi ft)$$

(3.3)

with $f = 0.95 \cdot f_s/2$, was sampled with rate $f_s = 1$ Hz:

$$\tilde{y}_k = y(k/f_s)$$

$$= \sin(2\pi fk/f_s).$$

(3.4)

(3.5)

The samples $\tilde{y}_k$ are plotted as fat dots. Note that the frequency $f$ is very close to the Nyquist rate $f_s/2$.

We now interpolate the discrete-time signal $\tilde{y}_k$ using our Kalman smoother. For the linear filter used by the Kalman smoother we used a Butterworth filter of order $N = 8$ and $-3$ dB-frequency at the Nyquist rate $f_c = f_s/2$. The Butterworth filter was chosen because its Fourier spectrum is very flat for frequencies smaller than $f_c$, and frequencies above $f_c$ are damped, where the damping depends on the filter order. Order $N = 8$ was chosen since the filter order should be as large as possible (to best approximate an ideal lowpass filter), but numerical problems arose with simple implementations of the Kalman smoother with Butterworth filters of order larger than 8. These numerical problems where not addressed in the scope of this thesis and should be subject to future research.

Since the samples $\tilde{y}_k$ are observed without additive noise, $\sigma_Z^2$ is set to zero. Note that in this case, any positive value $\sigma_U^2 \neq 0$ will have the same effect on the estimate $\hat{y}(t)$.

In Figure 3.3 the signal $y(t)$ (solid line) and the samples $\tilde{y}_k$ (fat dots) are plotted together with the estimate $\hat{y}(t)$ (dashed line) that was calculated by the Kalman smoother.

In some regions in the figure the estimate $\hat{y}(t)$ has a much smaller amplitude than the target signal $y(t)$. Since the sampling rate $f_s$ is larger than twice the frequency of the signal $y(t)$, this implies that components with frequencies larger than half the sampling rate $f_s$ are present in $\hat{y}(t)$. 
The difference between $\hat{y}(t)$ and $y(t)$ is due to the mismatch between the actual signal generation and the model used for signal estimation. The algorithm tries to find the input signal $\hat{u}(t)$ of the Butterworth filter that minimizes the cost function (2.17) given the model and the observations. Let $u(t)$ be the input signal of the filter that causes $y(t)$ in Figure 3.3, and $\hat{u}(t)$ the input signal that caused $\hat{y}(t)$, then

$$\int_{t_0}^{t_K} \hat{u}(t)^2 < \int_{t_0}^{t_K} u(t)^2,$$

and thus, given the model, $\hat{u}(t)$ is a better solution than $u(t)$. This is visualized in Figure 3.3, where the signals $u(t)$ and $\hat{u}(t)$ are plotted as well. Since $\sigma_Z^2 = 0$, the algorithm finds the signal $\hat{y}(t)$ with $\hat{y}(t_k) = \tilde{y}_k$, where the energy of $\hat{u}(t)$ is minimized.

In this example the $-3$ dB-frequency $f_c$ of the Butterworth filter was set to the Nyquist rate $f_s/2$. The slope of the Fourier spectrum of a Butterworth filter around $f_c$ is not very steep (cf. Appendix C). Thus, frequencies just above the Nyquist rate are not punished very hard in comparison to frequencies just below the Nyquist rate. Better results can be expected if $f_c$ is smaller.

Numerical results, which are visualized in Figure 3.4 support this claim. Three Butterworth filters of the same order $N = 8$, but different $-3$ dB-frequency $f_c$ where used to interpolate the samples of the signal

$$y(t) = \sin(2\pi ft).$$

The figure shows the SNR of the estimate $\hat{y}(t)$ (3.2) for different frequencies $f$. The best result was delivered with the Butterworth filter that has the smallest $f_c$. This suggests that the smaller $f_c$, the better the interpolation (ideally $f_c$ could be set to zero). This could not be verified due to numerical problems in the implementation, caused by very small estimation errors (if SNR$_{\hat{y}}$ is very large). We expect that the same numerical problems are responsible that the lines in Figure 3.4 are not smooth.

To interpolate a digital signal that was sampled with a very high SNR and close to the Nyquist rate, filters of very high order are needed. The Kalman smoother introduced in Chapter 2 might still be a useful interpolation algorithm for such a signals if the numerical issues of high order filters are solved. The strength of the interpolation algorithm is
Figure 3.4: Empirical SNR \( \hat{y} \) for signal interpolation for input signals \( y(t) = \sin(2\pi ft) \). The length of the signal \( y(t) \) was the length of 30 periods \( (T = 1/f) \) of the signal frequency, where the first and the last five periods were cut off for the calculation of the SNR.

that any hardware can be represented in the model, and thus especially digital signals that were sampled with very cheap hardware can be estimated efficiently.
Chapter 4

Input Signal Estimation

4.1 A Real-World Input Signal

As described in Section 2.5.5, an estimate of the input signal $U(t)$ in a model as depicted in Figure 1.1 can be calculated. In Chapter 2, the signal $U(t)$ is modeled as continuous-time white Gaussian noise, which is only a mathematical construct that can be used in a mathematical model, but does not exist in the real world. The assumption of white Gaussian noise as described in Chapter 2 results to an energy constraint on the estimate of the input signal, with no prior information about its spectrum.

In real-world applications the signal $U(t)$ will more likely be generated by a model as depicted in Figure 4.1, where $U(t)$ is filtered white Gaussian noise, and thus has a Fourier spectrum shaped by a continuous-time filter (Filter 1). Observed are the samples $\tilde{y}_k$, which are noisy observations of the output of a continuous-time filter (Filter 2) that is fed with the input signal $U(t)$.

The signal $W(t)$ in the figure is continuous-time white Gaussian noise where

$$\int_t^{t'} W(\tau)d\tau$$

(4.1)
Figure 4.1: Model of a real-world input/output signal pair.

is a zero-mean Gaussian random variable with variance $\sigma^2_{W}(t' - t)$. As in Chapter 2, $Z_k$ is discrete-time white Gaussian noise with $\mathbb{E}[Z_k^2] = \sigma^2_Z$.

Let the state variables of the filters evolve in time according to

\[
\begin{align*}
\dot{X}_1(t) &= A_1 X_1(t) + b_1 W(t) \quad (4.2) \\
\dot{X}_2(t) &= A_2 X_2(t) + b_2 U(t), \quad (4.3)
\end{align*}
\]

where

\[
\begin{align*}
U(t) &= c_1^T X_1(t) \quad (4.4) \\
Y(t) &= c_2^T X_2(t) \quad (4.5)
\end{align*}
\]

The total system can then be written as

\[
\begin{align*}
\begin{pmatrix}
\dot{X}_1(t) \\
\dot{X}_2(t)
\end{pmatrix} &= 
\begin{pmatrix}
A_1 & 0 \\
b_2 c_1^T & A_2
\end{pmatrix}
\begin{pmatrix}
X_1(t) \\
X_2(t)
\end{pmatrix} + 
\begin{pmatrix}
b_1 \\
b_2
\end{pmatrix} W(t) \quad (4.6)
\end{align*}
\]

\[
\begin{align*}
\begin{pmatrix}
U(t) \\
Y(t)
\end{pmatrix} &= 
\begin{pmatrix}
c_1^T & 0 \\
0 & c_2^T
\end{pmatrix}
\begin{pmatrix}
X_1(t) \\
X_2(t)
\end{pmatrix} + 
\begin{pmatrix}
b_1 \\
b_2
\end{pmatrix} \quad (4.7)
\end{align*}
\]

Given these state space equations, samples $x_{\text{tot}}(t_k)$ can be generated as described in Section 2.5.1, and thus samples $u(t_k)$ and $y(t_k)$ can be calculated.
4.2 Direct Estimate of the Input Signal

In Section 2.5.5 we described how the input signal of a continuous-time linear filter can be estimated. The signal \( U(t) \) is modeled as continuous-time white Gaussian noise, which is, of course, impossible to perfectly reconstruct; nevertheless a MAP / MMSE / LMMSE estimate that minimizes the cost function

\[
\frac{1}{\sigma_U^2} \int_{t_0}^{t_K} \hat{u}(t)^2 \, dt + \frac{1}{\sigma_Z^2} \sum_{k=1}^{K} (\tilde{y}_k - \mathbf{c}^T \hat{x}(t_k))^2,
\]

(4.8)

(see also (2.17)) can be calculated. The regularization parameter \( \sigma_U^2 / \sigma_Z^2 \) weights the cost of the energy of the estimate \( \hat{u}(t) \) of the input signal (first part), and the cost of the difference between the output estimate \( \tilde{y}(t_k) \) and the observations \( \tilde{y}_k \) (second part). As explained in Section 2.5.2 the parameter \( \sigma_U^2 / \sigma_Z^2 \) defines the SNR of the observed samples \( \tilde{y}_k \).

We now would like to estimate the signal \( U(t) \) that was generated by a model as depicted in Figure 4.1, from noisy discrete-time observations \( \tilde{y}_k \), without knowledge about the existence of Filter 1, and thus without prior information about the Fourier spectrum of \( U(t) \).

Lets illustrate this with numerical examples where samples \( u(t_k) \), \( y(t_k) \) and \( \tilde{y}_k \) are generated using two Butterworth filters. The \(-3 \,\text{dB}\)-frequency of Filter 1 is denoted as \( f_{c,1} \), and the \(-3 \,\text{dB}\)-frequency of Filter 2 as \( f_{c,2} \) accordingly.

For a first example, let \( f_{c,1} = 0.2 \,\text{Hz} \) and \( f_{c,2} = 1 \,\text{Hz} \). Note that in this case \( f_{c,1} < f_{c,2} \). Given noisy samples \( \tilde{y}_k \), an estimate \( \hat{u}(t) \) of \( U(t) \) is calculated as described in Section 2.5.5. If the value for \( \sigma_U^2 / \sigma_Z^2 \) is chosen that it matches SNR\( \tilde{y} \), which is known from signal generation, the estimate of \( U(t) \) is denoted as \( \hat{u}_{\text{SNR}}(t) \). An example of such an estimate is depicted in Figure 4.2.

It is clearly visible in Figure 4.2 that high frequency components are present in the estimate \( \hat{u}_{\text{SNR}}(t) \) that are not present in the reference signal \( u(t) \). This is not surprising: the algorithm assumes that \( U(t) \) is continuous-time white Gaussian noise, and thus assumes that all frequency components are present with equal probability. Thus, the optimal solution that minimizes the cost function (4.8) includes frequencies \( f \) with \( f_{c,1} < f < f_{c,2} \).
Chapter 4. Input Signal Estimation

Figure 4.2: Example of an input reconstruction, where the input signal $U(t)$ has a smaller bandwidth than Filter 2.

One approach to address this problem is to set $\sigma^2_U/\sigma^2_Z$ smaller in order to increase the penalty for high frequencies. Using a numerical algorithm\(^1\), the optimal $\sigma^2_U/\sigma^2_Z$ that minimizes the MSE

$$\frac{1}{N} \sum_{k=1}^{N} (u(t_k) - \hat{u}(t_k))^2$$

was calculated. The estimate that minimizes (4.9) is denoted as $\hat{u}_{opt}(t)$ and depicted in Figure 4.3. As seen in Figure 4.3, the signal energy of $\hat{u}_{opt}(t)$ is smaller than of $\hat{u}_{SNR}(t)$ but high frequency components are still present. Thus, $\hat{u}_{opt}(t)$ minimizes the MSE (4.9), but it does not seem to be a better estimate than $\hat{u}_{SNR}(t)$.

To illustrate the spectrum of the signals $u(t)$, $\hat{u}_{SNR}(t)$ and $\hat{u}_{opt}(t)$, the absolute values of their corresponding discrete-time Fourier spectrum are plotted in Figure 4.4. It is clearly visible how $\hat{u}_{SNR}(t)$ and $\hat{u}_{opt}(t)$ contain frequency components in the range $f_{c,1} < f < f_{c,2}$, where $\hat{u}_{opt}(t)$ seems to be very similar to $\hat{u}_{SNR}(t)$, but contains less energy in all frequency components.

In a second numerical example the input signal $U(t)$ has a larger “bandwidth” (recall that $U(t)$ is not strictly bandlimited) than the second filter, i.e., $f_{c,1} > f_{c,2}$. In that case, the second filter damps frequency

\(^1\)In the simulations of this chapter the “Golden Section Search” as described in [23] was used.
4.2. Direct Estimate of the Input Signal

Figure 4.3: Example of an input reconstruction with optimized $\sigma_U^2/\sigma_Z^2$, where the input signal $U(t)$ has a lower bandwidth than Filter 2.

Figure 4.4: Spectrum of the input signal $u(t)$ and its estimates, where $f_{c,1} = 0.2$ Hz, and $f_{c,2} = 1$ Hz. The curves are averages over 1000 simulations with signal length of 10 000 samples.
components that are present in $U(t)$, and thus it is impossible to reconstruct all frequency components of $U(t)$.

In Figure 4.5 an example of an input reconstruction, where $f_{c,1} = 1$ Hz and $f_{c,2} = 0.2$ Hz, is shown. The estimates, $\hat{u}_{\text{SNR}}(t)$ and $\hat{u}_{\text{opt}}(t)$ (as defined above) are both plotted in the figure and seem to be very similar. The assumption that $U(t)$ is continuous-time white Gaussian noise (that the energy is spread over all frequencies), only differs from the actual Fourier spectrum of $U(t)$ in a region where the damping of Filter 2 is very large. The frequencies where the model differs from the generated signal would never reach the samples $\tilde{y}_k$ anyway.

Figure 4.5 supports the claim that the lowpass components of the signal $U(t)$ are reconstructed and high frequency components are removed. The spectrum of $u(t)$ and its estimates is shown in Figure 4.6. It can clearly be seen that the energy in the passband of the second filter is similar to the energy of $u(t)$.

The example where $f_{c,2} > f_{c,1}$ shows, that the direct estimate of $U(t)$ can be highly useful if a signal is received and passed through an anti-aliasing filter before sampling. If the anti-aliasing filter does not have a flat Fourier spectrum in the passband, or disturbs the phase of the signal, the direct input estimate delivers the lowpass version of the signal that was actually received.
Figure 4.6: Spectrum of the input signal $u(t)$ and its estimates, where $f_{c,1} = 1$ Hz, and $f_{c,2} = 0.2$ Hz. The curves are averages over 1000 simulations with signal length of 10 000 samples.
4.3 Additional Spectral Shaping

If prior information about the spectrum of $U(t)$ is present it can be included in the model. If the spectrum of $U(t)$ can be described by a filter represented by finite order state space equations it can simply be added as a pre filter as in (4.6). The estimate of the input signal $U(t)$ is then found by calculating an estimate of the state vector $X_{\text{tot}}(t)$ that delivers an estimate of $U(t)$ using (4.7).

Using the same signal as in Figure 4.2 and Figure 4.3, the estimate with additional spectral shaping is depicted in Figure 4.7. The algorithm with additional spectral shaping works unsurprisingly well since the model that was used for signal generation exactly matches the model that was used for signal estimation.

In the case where the spectrum of the input signal is unknown, the direct estimate of the input signal as in Section 4.2 could be used to get a first estimate $\hat{u}(t)$ of $U(t)$. A guess about the spectrum of $U(t)$ could then be made using the spectrum of $\hat{u}(t)$, and the estimate of $U(t)$ could be improved iteratively.
Chapter 5

Correction of Clock Jitter

5.1 Overview

Clock jitter in A/D conversion occurs when, due to hardware imperfections, the exact sampling instants are unknown. For bandlimited signals approaches for clock jitter correction based on LMMSE estimation are introduced in [24] and [25]. In [24], a filter bank that converts a non-uniformly sampled signal in a uniformly sampled signal is extended for unknown delays in the sampling process. In [25], a Fourier series of a discrete-time signal is used to construct an LMMSE filter for clock jitter correction. In this chapter we show how the model and algorithms introduced in Chapter 2 can be used to improve the estimate of a continuous-time signal from discrete-time samples that are subject to clock jitter. The main difference between the model used in this thesis and the models used in [24] and [25] is, that we do not assume that the continuous-time signal in strictly bandlimited. The content of this chapter is also published in [11].

Clearly, if the slope of the continuous-time signal is large at the sampling instant, even a small deviation of the sampling instant has a large impact on the observation. Thus, based on the value of the derivative
of the continuous-time signal, a different noise power is added by clock jitter at each observation. An iterative algorithm is presented where the estimates of the derivatives of the continuous-time signal are used to improve the quality of the digital signal. Each iteration by itself calculates a LMMSE estimate.

## 5.2 The Jitter Error

For $k = 1, \ldots, K$ let a discrete-time (digital) signal $\tilde{Y}_k$ be noisy, discrete-time observations of continuous-time filtered noise as described in Chapter 2 with

$$\tilde{Y}_k = Y(T_k) + Z_k.$$  \hfill (5.1)

The sampling instances $T_k$ are random variables with

$$T_k = t_k + D_k,$$  \hfill (5.2)

where $t_1 < t_2 < t_3, \ldots$ (the desired sampling instances) are known, and $D_k$ is a zero-mean random delay with known variance $\mathbb{E}[D_k^2] = \sigma_D^2$.

Some delay $D_k$ of the sampling instant causes a higher error for $\tilde{Y}_k$ if the slope of $Y(t)$ at $t = t_k$ is large. This is expressed by a first order Taylor approximation of $Y(t)$:

$$\tilde{Y}_k = Y(t_k + D_k) + Z_k$$

$$\approx Y(t_k) + D_k \dot{Y}(t_k) + Z_k.$$  \hfill (5.4)

Thus, we define the jitter error as

$$J_k \triangleq D_k \dot{Y}(t_k),$$  \hfill (5.5)

with

$$\mathbb{E}[J_k] = 0$$  \hfill (5.6)

$$\mathbb{E}[J_k^2] = \sigma_D^2 \mathbb{E}[\dot{Y}(t_k)^2].$$  \hfill (5.7)
5.3. Factor Graph Representation

For the observations $\tilde{Y}_k$ of $Y(t)$ this means

$$\tilde{Y}_k = Y(T_k) + Z_k$$

$$\approx Y(t_k) + D_k \dot{Y}(t_k) + Z_k.$$  (5.8)

Given the state space model as described in Chapter 2, $\dot{Y}(t)$ can be expressed as

$$\dot{Y}(t) = c^T \dot{X}(t)$$

$$= c^T A X(t) + c^T bU(t).$$  (5.10)

As shown in Section 2.5.4, the estimate of $Y(t)$ is only smooth if $c^T b = 0$. Since only filters are considered which fulfill this condition, the slope of $Y(t)$ can directly be calculated from the state space variables:

$$\dot{Y}(t) = c^T A X(t),$$  (5.12)

and therefore, an estimate of $\dot{Y}(t)$ can be calculated directly, if an estimate of $X(t)$ is present.

5.3 Factor Graph Representation

The factor graph in Figure 2.1, which is used for our Kalman smoother, is easily extended for clock jitter correction as shown in Figure 5.1. As all noise sources in this thesis, $J_k$ is modeled as a Gaussian random variable. According to (5.5), $\sigma^2_J(t_k) \triangleq \mathbb{E}[J_k^2]$ depends on $\dot{Y}(t_k)$, and thus $\sigma^2_J(t_k)$ is not constant for all $k$.

Calculating all the messages and estimates in Figure 5.1 as described in Section 2.4 results in minimizing the cost function

$$\frac{1}{\sigma_U^2} \int_{t_0}^{t_K} \dot{u}(t)^2 dt + \sum_{k=0}^{K-1} \frac{(\tilde{y}_k - c^T \dot{x}(t_k))^2}{\sigma^2_Z + \sigma^2_J(t_k)}. $$  (5.13)

Note that, as opposed to the cost function (2.17), not just the fraction $\sigma^2_U/\sigma^2_Z$ can be used for regularization. The penalty for $\tilde{y}_k - \dot{y}(t_k)$ (with $\dot{y}(t_k) \triangleq c^T \dot{x}(t_k)$) also depends on the value of $\sigma^2_J(t_k)$. 
Figure 5.1: A Forney factor graph of the system in Figure 1.1, extended for a signal that is observed with clock jitter.
5.4 Definitions of SNR

The following definitions of SNR are used in Section 5.5 and Section 5.6. For the discrete-time (digital) signal $\tilde{Y}_k$ three different SNR are defined:

\[
\text{SNR}_{\tilde{y}} \triangleq \frac{\mathbb{E}[Y(t)^2]}{\sigma_Z^2 + \mathbb{E}[J_k^2]} \quad (5.14)
\]
denotes the SNR of the noisy, discrete-time signal $\tilde{y}_k$,

\[
\text{SNR}_Z \triangleq \frac{\mathbb{E}[Y(t)^2]}{\sigma_Z^2} \quad (5.15)
\]
denotes the SNR of $\tilde{y}_k$ in the absence of clock jitter and

\[
\text{SNR}_J \triangleq \frac{\mathbb{E}[Y(t)^2]}{\mathbb{E}[J_k^2]} = \frac{\mathbb{E}[Y(t)^2]}{\sigma_D^2 \mathbb{E}[\dot{Y}(t)^2]} \quad (5.16)
\]
denotes the SNR of $\tilde{y}_k$ if only the jitter noise is present.

The SNR of the estimate $\hat{y}(t_k) \triangleq c^T \hat{x}(t_k)$ of $Y(t_k)$ is denoted as

\[
\text{SNR}_{\hat{y}} \triangleq \frac{\mathbb{E}[Y(t_k)^2]}{\mathbb{E}[\left(Y(t_k) - \hat{Y}(t_k)\right)^2]} \quad (5.18)
\]

Assuming the linear system is stable (all poles have negative real parts), the quantity $\mathbb{E}[Y(t)^2]$ can be calculated using (2.23):

\[
\mathbb{E}[Y(t)^2] = c^T \mathbb{E}[X(t)^2] c \quad (5.19)
\]
\[
= c^T \overrightarrow{V}_{X(\infty)} c \quad (5.20)
\]
and thus

\[
\mathbb{E}[\dot{Y}(t)^2] = c^T \mathbb{E}[\dot{X}(t)^2] c \quad (5.21)
\]
\[
= c^T A \overrightarrow{V}_{X(\infty)} (c^T A)^T. \quad (5.22)
\]
5.5 The Algorithm

We now describe an iterative algorithm to estimate the signal \( Y(t_k) \) for all \( k \). Note that the sampling instances \( t_k \) need not be equidistant; however, we will assume that they are in order so that \( t_k < t_{k'} \) for \( k < k' \).

Let \( x^{(\ell)} \) denote the value of some variable \( x \) at the \( \ell \)-th iteration. Then at each iteration \( \ell \), the estimate \( \hat{x}(t_k)^{(\ell)} \) of \( X(t_k) \) is calculated as described in Section 5.3. The estimates of \( Y(t_k) \) and \( \dot{Y}(t_k) \) are

\[
\hat{y}(t_k)^{(\ell)} \triangleq c^T \hat{x}(t_k)^{(\ell)} \tag{5.23}
\]

\[
\hat{\dot{y}}(t_k)^{(\ell)} \triangleq c^T A \hat{x}(t_k)^{(\ell)} \tag{5.24}
\]

At the first iteration no estimates of the derivatives are available. Thus, \( \sigma^2 J(t_k)^{(1)} \) is initialized to the expected average variance of \( J_k \) that is constant for all \( k \) and can be calculated using (5.7) and (5.22):

\[
\sigma^2 J(t_k)^{(1)} \triangleq \sigma^2_D c^T A \mathbf{V}_X(\infty) (c^T A)^T. \tag{5.25}
\]

Thus, at the first iteration, the noise added by clock jitter is modeled as discrete-time white Gaussian noise, which means that the first estimate of \( Y(t_k) \) is an LMMSE estimate as described in Chapter 2. Hence, the first iteration is similar to the algorithm described in [25], the main difference is that the model in Chapter 2 does not imply that \( Y(t) \) is strictly bandlimited.

At iteration \( \ell > 1 \), the value for \( \sigma^2 J(t_k)^{(\ell)} \) is set based on the estimate of \( \hat{Y}(t_k) \) of the previous iteration:

\[
\sigma^2 J(t_k)^{(\ell)} \triangleq \sigma^2_D \left( \hat{y}(t_k)^{(\ell-1)} \right)^2 = \sigma^2_D \left( c^T A \hat{x}(t_k)^{(\ell-1)} \right)^2 \quad \ell = 2, 3, \ldots. \tag{5.26}
\]

Changing \( \sigma^2 J(t_k) \) for each iteration causes a change of the total noise energy in the model, and thus changes the value of \( \text{SNR}_{\hat{y}} \) that is assumed by the algorithm (see Section 2.5.2). Since \( \sigma_U^2 \) is proportional to \( \text{E}[Y(t)^2] \) (cf. (2.14) and (2.23)), \( \text{SNR}_{\hat{y}} \) is proportional to

\[
\frac{\sigma^2_U^{(\ell)}}{\sigma_Z^2 + \sigma_D^2 \frac{1}{N} \sum_{k=1}^N \left( c^T A \hat{x}(t_k)^{(\ell-1)} \right)^2}, \tag{5.28}
\]
where the denominator of (5.28) represents the total average noise energy added to the samples \( Y(t_k) \). Thus, \( \sigma^2_U^{(\ell)} \) is chosen to keep (5.28) constant over all iterations.

For the numerical results given in Section 5.6, the values for \( \sigma_D^2 \), \( \sigma_Z^2 \) and \( \sigma_U^{(1)} \) are set equal to the values used for signal generation.

5.6 Numerical Results

To test the algorithm\(^1\), sample signals \( \tilde{y}_k \) where generated as described in Section 2.5.1 and Section 2.5.2. For the continuous-time system a low-pass Butterworth filter of order 8 was used (see Appendix C for a brief summary of Butterworth filters). Recall that \( Y(t) \) is not strictly band-limited but its spectrum is shaped by the spectrum of the continuous-time filter. We characterize the “bandwidth” of \( Y(t) \) by the \(-3 \) dB-frequency \( f_c \) of the Butterworth filter.

For signal generation, the probabilistic delay \( D_k \) of the sampler was chosen to be uniformly distributed in the interval \( \left( -\frac{0.25}{f_s}, \frac{0.25}{f_s} \right) \). For the estimation, however, it is assumed that the time delay is a Gaussian random variable with

\[
E[D_k] = 0 \quad (5.29)
\]
\[
E[D_k^2] = \left( \frac{0.25}{f_s} \right)^2. \quad (5.30)
\]

The algorithm stops after the \( \ell \)-th iteration if

\[
\frac{\text{SNR}_{\text{out}}^{(\ell)}[\text{dB}] - \text{SNR}_{\text{out}}^{(\ell-1)}[\text{dB}]}{\text{SNR}_{\text{out}}^{(\ell)}[\text{dB}] - \text{SNR}_{\text{out}}^{(0)}[\text{dB}]} < 0.01. \quad (5.31)
\]

Simulations were performed for three different values of the \(-3 \) dB-frequency \( f_c \) (thus, different “bandwidths” of \( Y(t) \)). The results are shown in Figure 5.2, where \( \text{SNR}_{\hat{y}} \) of the estimates after the first and the last

\(^1\)The authors would like to thank Mr. Daniel Baumann for the initial implementation of the algorithms of this chapter and for providing first simulation results.
iteration are plotted for different values for SNR$_Z$; the difference between these lines shows the improvement due to iterative processing. To show the overall improvement, SNR$_{\tilde{y}}$ of the observed, noisy samples $\tilde{y}_k$ is plotted as well.

Not surprisingly, the numerical results of the first iteration resemble the results in [25]. As mentioned in the introduction, the first iteration of our proposed algorithm is similar to the algorithm proposed in [25]. Since [25] assumes strictly bandlimited signals an exact comparison does not make sense.

![SNR Graph]

**Figure 5.2:** Empirical SNR$_{\tilde{y}}$ and SNR$_{\hat{y}}$ for different “bandwith” of $Y(t)$. Each point in the plot was generated by averaging the result of simulating 100 signals with 100 000 samples. The algorithm always stopped after three or four iterations.

The higher the sampling rate $f_s$, the larger the gap between SNR$_{\tilde{y}}$ and SNR$_{\hat{y}}$ of the first iteration, where iterative processing only seems to help when SNR$_Z$ is large. This is not surprising, the average energy of the jitter noise dependends on $f_c/f_s$ (and not on $\sigma_Z^2$). In the case of a
Butterworth filter of order 8, this means

\[
\text{SNR}_J = \begin{cases} 
17.4 \text{ dB}, & \text{if } f_c/f_s = 2^{-2} \\
29.5 \text{ dB}, & \text{if } f_c/f_s = 2^{-4} \\
41.5 \text{ dB}, & \text{if } f_c/f_s = 2^{-6} 
\end{cases} \tag{5.32}
\]

Thus, if SNR$_Z$ is small, the jitter noise energy is comparatively small and therefore clock jitter correction can not improve the signal quality. On the right half of the plot in Figure 5.2, the jitter noise is the dominant noise source, and thus iterative processing significantly improves SNR$_g$. 
Chapter 6

Analog-to-Digital Conversion Using Unstable Linear Filters

6.1 Overview

In this chapter we illustrate how unstable linear filters can be used in A/D conversion. An example of such a system is the sigma-delta converter [21], where the unstable linear filter is an integrator. In Figure 6.1 a simple sigma-delta converter is illustrated: the output of the integrator is sampled and quantized to +1 or −1. The negative feedback from these samples $\tilde{y}_k$ ensures that the output of the integrator $y(t)$ does not diverge, if the input signal is bounded, i.e., $|u(t)| \leq 1$. Using this architecture, doubling the sampling rate $f_s$ decreases the SNR in the passband of the quantized signal by 9 dB, as opposed to 3 dB with only a 1-bit quantizer [21].

A more general view on how unstable linear systems can be used in A/D converters was introduced in [26]. An “Unstable Linear Filter ADC” consists of an analog part and a digital part. The analog part contains some unstable linear system. Some discrete-time, quantized (and thus
Figure 6.1: A simple sigma-delta converter.

Figure 6.2: The general scheme of an “Unstable Linear Filter ADC”.

digital) observations of the system are fed back as separate inputs to ensure that its internal variables stay in a predefined range. Some other discrete-time quantized observations are passed to the digital part, which then tries to estimate the input signal of the analog part. In this chapter we show how the model and algorithms introduced in Chapter 2 can be used to implement the digital part.

See Figure 6.2 for the general concept of the analog part with digital feedback: a continuous-time signal $u(t)$ is fed into an unstable linear system. Some discrete-time observations of the system are quantized to 1-bit resolution and fed back into the system. These feedback values $s_{m,k}$, together with other discrete-time, quantized observations $\tilde{y}_{\ell,k}$, are passed to the digital part, where samples $\tilde{y}_{\ell,k}$ may be quantized to a higher resolution than one bit.

In this chapter we show an example on how such an “Unstable Linear
6.2 A Single Integrator

As mentioned in Section 6.1, a simple example of an “Unstable Linear Filter ADC” is the sigma-delta converter. In Figure 6.3 the system from Figure 6.1 is extended and the notation is adjusted to match Figure 6.2. For simplicity, let’s assume that the samples of $y(t)$ are taken at uniform rate $f_s$, and thus at sampling instances $t_k = k/f_s = kT$, with $T \triangleq 1/f_s$.

The unstable linear system itself is an integrator. Feedback from the output of the integrator to its input ensures that the output signal $y(t)$ will stay in a finite range if the input signal $u(t)$ is bounded by 1, i.e.,

$$|u(t)| \leq 1 \quad \forall t \in \mathbb{R}. \quad (6.1)$$

The output $y(t)$ of the integrator at time $t_k < t < t_{k+1}$ evolves as

$$y(t) = y(t_k) + \int_0^{t-t_k} (u(t_k + \tau) - s_k) \, d\tau \quad (6.2)$$

$$= y(t_k) - s_k(t - t_k) + \int_0^{t-t_k} u(t_k + \tau) \, d\tau, \quad (6.3)$$

where

$$s_k \triangleq \begin{cases} +1 & \text{if } y(t_k) \geq 0 \\ -1 & \text{if } y(t_k) < 0 \end{cases}. \quad (6.4)$$
Figure 6.4: A simple sigma-delta converter, where the input signal is bounded by a value larger than one.

Thus, if $u(t)$ is bounded by 1, the output $y(t)$ will be bounded by $2T$:

$$|y(t)| \leq 2T = 2/f_s. \hspace{1cm} (6.5)$$

If, due to hardware restrictions, $|y(t)|$ has to be smaller than some known $y_{\text{max}} > 0$, the sampling rate $f_s$ needs to satisfy

$$f_s \geq 2/y_{\text{max}}. \hspace{1cm} (6.6)$$

If the input signal is not bounded by 1, but by some larger value $u_{\text{max}} > 1$, the output of the integrator in Figure 6.3 is not guaranteed to stay bounded. Assume, e.g., the constant input signal $u(t) = u_{\text{max}}$. Since the feedback value is either +1 or −1, the input of the integrator will always be positive and its output will increase over time. The feedback needs to be able to compensate the maximal possible value of the input signal. Thus, the feedback needs to be amplified by some $e$, as illustrated in Figure 6.4, which satisfies

$$e \geq u_{\text{max}}, \hspace{1cm} (6.7)$$

and thus

$$|y(t)| \leq T(u_{\text{max}} + e) \quad \forall t \in \mathbb{R}. \hspace{1cm} (6.8)$$

If hardware restrictions demand that $|y(t)| \leq y_{\text{max}}$, the sampling rate $f_s$ needs to satisfy

$$f_s \geq \frac{u_{\text{max}} + e}{y_{\text{max}}} \hspace{1cm} (6.9)$$

$$\geq \frac{2u_{\text{max}}}{y_{\text{max}}}, \hspace{1cm} (6.10)$$

where (6.10) follows from (6.7).
6.3 A Linear System of Order Two

More complex systems than just an integrator can be used in an “Unstable Linear Filter ADC”. Since any linear system of even order can be split into a series of sections of order two, we now address a general system of order two.

In particular consider a system with a state space representation as

\[
\dot{x}(t) = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} x(t) + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} u(t) \tag{6.11}
\]

\[
y(t) = \begin{pmatrix} c_1 & c_2 \end{pmatrix} x(t) + du(t). \tag{6.12}
\]

Note that there is a direct path from \(u(t)\) to \(y(t)\), thus the system is not necessarily a lowpass filter.

If the eigenvalues of \(A\) (the poles of the system) both lie in the right half-plane, i.e., have positive real parts, the section is not stable and a feedback path needs to be added to the system to ensure that the state values will stay in a defined range. Figure 6.5 illustrates how a system of order two as in (6.11)–(6.12) may look like with feedback paths amplified with \(e_{11}\) and \(e_{22}\).

The output \(x_i(t)\) of integrator \(i\) is sampled with rate \(f_s = 1/T\), quantized to +1 or −1, amplified with \(e_{ii}\) and subtracted from the input of integrator \(i\). The samples \(\tilde{y}_{i,k}\) that are passed to the digital part of the A/D converter, are quantized representations of \(x_i(t_k)\) (where \(t_k = Tk\)). Note that \(\tilde{y}_{i,k}\) may be quantized to a larger resolution than just one bit.

Adding the feedback paths inflicts a change in the state space equations (6.11)–(6.12). At time \(t\) with \(t_k < t < t_{k+1}\), where \(t_k\) and \(t_{k+1}\) are consecutive sampling instants, the state vector \(\dot{x}(t)\) evolves in time according to

\[
\dot{x}(t) = Ax(t) + bu(t) - Es_k \tag{6.13}
\]

with

\[
E \triangleq \begin{pmatrix} e_{11} & 0 \\ 0 & e_{22} \end{pmatrix} \tag{6.14}
\]
Figure 6.5: A general second-order system with feedback.
\[ s_k \triangleq \begin{pmatrix} s_{1,k} \\ s_{2,k} \end{pmatrix}, \quad (6.15) \]

and

\[ s_{i,k} = \begin{cases} +1 & \text{if } x_i(t_k) \geq 0 \\ -1 & \text{if } x_i(t_k) < 0 \end{cases}. \quad (6.16) \]

The values \( e_{ii} \) required in order to stabilize the system are found similarly as for the sigma-delta converter in Section 6.2: if the input \( u(t) \) is bounded by some \( u_{\text{max}} > 0 \) and the output of integrator \( i \) needs to be bounded by some \( x_{i,\text{max}} > 0 \) (e.g., due to hardware restrictions), then \( e_{ii} \) has to be able to compensate the largest possible value of the input of integrator \( i \), and thus values for \( e_{ii} \) are allowed that satisfy

\[ e_{11} \geq u_{\text{max}} |b_1| + x_{1,\text{max}} a_{11} + x_{2,\text{max}} |a_{12}| \quad (6.17) \]
\[ e_{22} \geq u_{\text{max}} |b_2| + x_{1,\text{max}} |a_{21}| + x_{2,\text{max}} a_{22}. \quad (6.18) \]

By solving the differential equation (6.13), \( \mathbf{x}(t) \) can be calculated. Let \( t_k < t < t_{k+1} \), then

\[
\mathbf{x}(t) = e^{A(t-t_k)} \mathbf{x}(t_k) + \int_{t_k}^{t} e^{A(t-t_k-\tau)} (\mathbf{b}u(t_k + \tau) - ES_k) \, d\tau
\]

\[ = e^{A(t-t_k)} \mathbf{x}(t_k) + \int_{0}^{t-t_k} e^{A(t-t_k)} \mathbf{b}u(t_k + \tau) \, d\tau
\]

\[ - A^{-1} \left( e^{A(t-t_k)} - I_2\right) ES_k, \quad (6.20) \]

where \( I_2 \) is the unit matrix in \( \mathbb{R}^{2 \times 2} \).

From (6.20) it is not as simple as in Section 6.2 to calculate the largest value that can be reached by \( |x_i(t)| \) (the output of integrator \( i \)). It is clear, that the largest value of the integrator outputs can only be reached if the input signal \( u(t) \) is constant at either \( +u_{\text{max}} \) or \( -u_{\text{max}} \), in which case the integral in (6.20) is a constant. The output \( x_i(t) \) of integrator \( i \) is then an exponential function in \( t \). Since the eigenvalues of \( A \) all have positive real parts, \( x_i(t) \) either increases or decreases monotonically in time. Thus, if

\[ |x_i(t_{k+1})| \leq x_{i,\text{max}}, \quad (6.21) \]
for any $x(t_k)$ so that $|x_i(t_k)| \leq x_{i,\text{max}}$, we are assured that $x(t)$ never leaves its legal range for $t_k < t < t_{k+1}$.

We define

$$x^{(+)}(t_{k+1}) \triangleq e^{A T} x(t_k) + A^{-1} (e^{A T} - I_2) (+b u_{\text{max}} - E s_k) \tag{6.22}$$

$$x^{(-)}(t_{k+1}) \triangleq e^{A T} x(t_k) + A^{-1} (e^{A T} - I_2) (-b u_{\text{max}} - E s_k), \tag{6.23}$$

and

$$x^*_i(t_{k+1}) \triangleq \max \left\{ |x^{(+)}_i(t_{k+1})|, |x^{(-)}_i(t_{k+1})| \right\} \geq |x_i(t)|, \tag{6.24}$$

for $i \in \{1, 2\}$ and $t_k < t < t_{k+1}$.

The elements of $x^{(+)}(t_{k+1})$ and $x^{(-)}(t_{k+1})$ are linear functions with parameters $x_1(t_k)$ and $x_2(t_k)$. Since $s_{i,k}$ depends on the sign of $x_i(t_k)$ the maximal value of $x^*_i(t_{k+1})$ lies on a point of the discrete set

$$x(t_k) \in \{-x_{1,\text{max}}; -\varepsilon; 0; x_{1,\text{max}}\} \times \{-x_{2,\text{max}}; -\varepsilon; 0; x_{2,\text{max}}\} \tag{6.25}$$

with $\varepsilon \approx 0$ and $\varepsilon > 0$.

Thus, we can check for any sampling rate $f_s$ if the system is feazible by testing if

$$x^*_i(t_{k+1}) \leq x_{i,\text{max}} \tag{6.26}$$

for both states $i$ and all the 8 points in (6.25).

Recall that the state space representation is not unique for a given system. If some sampling rate $f_s$ is feasible for some given representation, it does not imply that $f_s$ is also feasible for any other representation of the same filter.

### 6.4 Factor Graph and Estimation Algorithm

The message passing rules derived in Section 2.4 are extended in this section by adding the feedback that is needed to stabilize the system. If the input signal $U(t)$ is modeled as continuous-time white Gaussian noise
as in Chapter 2, the state variables of the system are random variables that, for \( t_k < t < t_{k+1} \), evolve according to (6.13)

\[
\dot{X}(t) = AX(t) + bU(t) - Es_k,
\]

which solves to (6.20)

\[
X(t) = e^{A(t-t_k)}X(t_k) + \int_0^{t-t_k} e^{A(t-t_k)}bU(t_k + \tau) d\tau - A^{-1} \left( e^{A(t-t_k)} - I_2 \right) Es_k.
\]

For any fixed \( X(t_k) = x(t_k) \) (6.28) induces a probability density function \( f(x(t_{k+1})|x(t_1)) \), which is also Gaussian. As in Section 2.3, this conditional density can be used as a building block in a Forney factor graph. Specifically it can be represented by the factor graph in Figure 6.6, where

\[
m_k = -A^{-1} \left( e^{AT} - I_2 \right) Es_k
\]

\[
V_S = \sigma_U^2 \int_0^T e^{A\tau} b b^T (e^{A\tau})^T d\tau,
\]

with \( T = t_{k+1} - t_k \). The message passing rules with the feedback path are summarized in Table 6.1. Note that the calculation of \( m_U(t) \) (vi.i.5) did not change compared to (ii.i.5) in Table 2.1 when adding constant feedback since the constant input \(-Es_0\) is already included in the message \( \mu X(t_0) \) (which is needed to calculate \( m_U(t) \)). The message passing rules in Table 6.1 can easily be verified by adding the constant input \(-Es_0\) to the proofs in Appendix B.

### 6.5 Implementation of an Example System

#### 6.5.1 Generation of the System

Let's illustrate an example of an “Unstable Linear Filter ADC” where the linear system is a Butterworth filter of order four with the poles
Table 6.1: Messages through node / factor \( f(x(t_1)|x(t_0), s_0) \), where \( t_1 > t_0 \) and \( s_0 \) is the constant feedback in the interval \( t_0 < t < t_1 \) as in Figure 6.5.

\[
\begin{align*}
\vec{m}_{X(t_1)} &= e^{A(t_1-t_0)} \vec{m}_{X(t_0)} - A^{-1} \left( e^{A(t_1-t_0)} - I_n \right) ES_0 \\
\vec{V}_{X(t_1)} &= e^{A(t_1-t_0)} \vec{V}_{X(t_0)} e^{A^T(t_1-t_0)} \\
&\quad + \sigma_U^2 \int_0^{t_1-t_0} e^{A\tau} bb^T e^{A^T\tau} d\tau \\
&\quad \underbrace{Q\Theta(t_1-t_0)Q^H}_{\text{see (2.13)}} \\
\vec{m}_{X(t_0)} &= e^{-A(t_1-t_0)} \left( \vec{m}_{X(t_1)} + A^{-1} \left( e^{A(t_1-t_0)} - I_n \right) ES_0 \right) \\
\vec{V}_{X(t_0)} &= e^{-A(t_1-t_0)} \vec{V}_{X(t_1)} e^{-A^T(t_1-t_0)} \\
&\quad + \sigma_U^2 \int_0^{t_1-t_0} e^{-A\tau} bb^T e^{-A^T\tau} d\tau \\
&\quad \underbrace{Q\Theta(t_1-t_0)Q^H}_{\text{see (2.15)}} \\
m_{U(t)} &= \sigma_U^2 b^T \left( \vec{V}_{X(t)} + \vec{V}_{X(t)} \right)^{-1} \left( \vec{m}_{X(t)} - \vec{m}_{X(t)} \right) \\
V_{U(t)} &\to \infty
\end{align*}
\]
6.5. Implementation of an Example System

\[ X(t_k) \xrightarrow{\mathcal{N}(m_k, V_S)} e^{AT} \xrightarrow{+} X(t_{k+1}) \]

Figure 6.6: Forney factor graph of \( f(x(t_{k+1})|x(t_1)) \), extended for an additional constant input according to (6.27)–(6.30).

\[ p_1 = \rho e^{i\varphi_1}, \quad \overline{p}_1 = \rho e^{-i\varphi_1}, \quad p_2 = \rho e^{i\varphi_2} \quad \text{and} \quad \overline{p}_2 = \rho e^{-i\varphi_2}, \]

where

\[ \rho = 2\pi f_c \quad (6.31) \]
\[ \varphi_1 = \frac{7\pi}{8} \quad (6.32) \]
\[ \varphi_2 = \frac{5\pi}{8} \quad (6.33) \]

and \( f_c \) denotes the \(-3\) dB-frequency of the Butterworth filter (cf. Appendix C).

This filter can be split into two sections, where each section contains one of the conjugate complex pole pairs. For a stable Butterworth filter of order four the state space equation is

\[
\begin{pmatrix}
\dot{x}_1(t) \\
\dot{x}_2(t)
\end{pmatrix} =
\begin{pmatrix}
A_1 & 0 \\
b_2c_1^T & A_2
\end{pmatrix}
\begin{pmatrix}
x_1(t) \\
x_2(t)
\end{pmatrix} +
\begin{pmatrix}
b_1 \\
0
\end{pmatrix} u(t),
\]

and the output \( y(t) \) of the filter is

\[
y(t) = \left( 0 \quad c_2^T \right) \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix},
\]

with

\[
A_m = \rho^2 \begin{pmatrix}
\cos(\varphi_m) & -\sin(\varphi_m) \\
\sin(\varphi_m) & \cos(\varphi_m)
\end{pmatrix}
\]
\[
b_m = \begin{pmatrix}
1 \\
-1
\end{pmatrix} \frac{\rho^2}{\sin(\varphi_m)}
\]
\[
c_m^T = \begin{pmatrix}
0.5 \\
0.5
\end{pmatrix}.
\]

Note that this filter is a lowpass filter and there is no direct path from \( u(t) \) to \( y(t) \). In Section 6.3 systems where described that are not necessarily
lowpass filters. Since the estimation of the input signal will be done with message passing algorithms as described in Section 2.5.5 we require the linear system to be a lowpass filter, which does not imply that each second order section itself is also a lowpass filter.

Recall that the state space representation of a filter is not unique. The particular choice of the input vectors $b_m$ and the rotation matrix $A_m$ helps to evenly distribute the energy in the system among the states. Of the representations that were implemented in the scope of this thesis, this representation allowed to stabilize an unstable section with the lowest sampling rate.

If section $m$ is to be unstable, the poles are moved to the right half plane with $p_m = \rho e^{i\varphi_m} \mapsto \rho e^{i(\pi - \varphi_m)}$, and thus the matrix $A_m$ changes to

$$A'_m = \rho^2 \begin{pmatrix} -\cos(\varphi_m) & -\sin(\varphi_m) \\ \sin(\varphi_m) & -\cos(\varphi_m) \end{pmatrix},$$

(6.39)

and a feedback path as in (6.27) is added. Thus, e.g., if the first section is stable, and the second section is unstable, the state space equations for the total systems for $t_k < t < t_{k+1}$ are

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} A_1 & 0 \\ b_2 c_1^T & A_2' \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} b_1 \\ 0 \end{pmatrix} u(t)$$

(6.40)

$$- \begin{pmatrix} 0 & 0 \\ 0 & E_2 \end{pmatrix} \begin{pmatrix} 0 \\ s_k \end{pmatrix}.$$  

(6.41)

The elements of the vector $s_k$ are defined as in (6.16) dependant on the elements of the vector $x_2(t_k)$. The feedback matrix is

$$E_2 = \begin{pmatrix} e_{11} & 0 \\ 0 & e_{22} \end{pmatrix}$$

(6.42)

with feedback values set to the smallest possible value from (6.17)–(6.18). The maximal values for $u(t)$ and the state variables in the numerical example in this section are set to 1, thus

$$e_{11} = \frac{\rho^2}{\sin(\varphi_2)} - \cos(\varphi_2) + \sin(\varphi_2)$$

(6.43)

$$e_{22} = \frac{\rho^2}{\sin(\varphi_2)} + \sin(\varphi_2) - \cos(\varphi_2).$$

(6.44)
The observations $\tilde{y}_k$ that are passed to the Kalman filter, are the sampled and quantized outputs of the integrators of the second section:

\[
\begin{pmatrix}
\tilde{y}_{1,k} \\
\tilde{y}_{2,k}
\end{pmatrix} = \begin{pmatrix} 0 & I_2 \end{pmatrix} \begin{pmatrix} x_1(t_k) \\
x_2(t_k) \end{pmatrix} + Z_k \\
= x_2(t_k) + Z_k,
\]

(6.45)

(6.46)

where $Z_k$ is the quantization error. In Figure 6.7 the value of $\tilde{y}_{i,k}$ depending on $x_i(t_k)$ for a 3-bit quantizer is illustrated.

If the quantization error is uniformly distributed in the interval between $+\frac{1}{2^n}$ and $-\frac{1}{2^n}$, where $n$ is the number of bits of the quantizer (which is a widely used assumption, cf. [21]), and the elements of $Z_k$ are uncorrelated, the mean and covariance matrix of $Z_k$ are

\[
E[Z_k] = 0 \quad (6.47)
\]

\[
V_Z \triangleq E[Z_k Z_k^T] = \frac{1}{3} \left( \frac{1}{2^n} \right)^2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (6.48)
\]

Recall that the Kalman smoother will assume the quantization error to be white Gaussian noise, which, of course, is an approximation.

Note, that only the states of the second section are passed to the Kalman filter. Using observations of the first section would result in a non-smooth estimate of the input signal $U(t)$ as mentioned in Section 2.5.5.

### 6.5.2 Simulation of the System

We show how an “Unstable Linear Filter ADC” can be simulated with an input signal

\[
u(t) = \sin(2\pi ft)
\]

(6.49)

for different frequencies $f$. Recall that the state $x(t)$ for $t_k < t < t_{k+1}$ evolves as

\[
\dot{x}(t) = Ax(t) + bu(t) - Es_k
\]

(6.50)
as in (6.13), and thus

\[
x(t) = e^{A(t-t_k)} x(t_k) + \int_{0}^{t-t_k} e^{A(t-t_k)} b u(t_k + \tau) d\tau \\
- A^{-1} \left( e^{A(t-t_k)} - I_n \right) Es_k,
\]

as in (6.20). For \( u(t) = \sin(2\pi ft) \) the integral in (6.51) can be solved analytically, and the value for \( x(t_{k+1}) \), given \( x(t_k) \) (which also defines \( s_k \)), can be calculated. The quantized observations \( \tilde{y}_{k+1} \) can then easily be calculated from \( x(t_{k+1}) \).

### 6.5.3 Training the Estimation Algorithm

Using (6.48), a value for the variance of the quantization error under which the quantized samples are observed can be calculated from the system parameters. Though the optimal choice for \( \sigma_U^2 \), the penalty on the energy of the estimate of the input signal, is not clear. Using a training signal, the value for \( \sigma_U^2 \) can be calculated using a numerical
algorithm\textsuperscript{1}.

The training signal $u(t)$ that was chosen for the numerical results in this chapter was a sum of sine waves

$$u(t) = \frac{1}{15}\sum_{\ell=1}^{15}\sin(2\pi f_{\ell}t).$$

(6.52)

The frequencies $f_{\ell}$ where linearly spread in the interval $0 < f_{\ell} < f_c$, where $f_c$ is the $-3$ dB-frequency of the Butterworth filter that was used as the linear system in the "Unstable Linear Filter ADC".

In Section 6.5.2 we showed how the "Unstable Linear Filter ADC" can be simulated with a sine wave as its input signal. Since the unstable system is linear, the simulation of the system with a sum of sine waves as its input signal is straight forward. Given the observations generated from $u(t)$, an estimate $\hat{u}(t_k)$ can be calculated using the factor graph in Figure 2.6, but calculating the messages through $f(x(t_{k+1})|x(t_1))$ as described in Table 6.1. The value for $\sigma_U^2$ that minimizes the MSE

$$\frac{1}{K}\sum_{k=1}^{K}(u(t_k) - \hat{u}(t_k))^2,$$

(6.53)

where $K$ is the number of samples of the training signal, can be found numerically.

### 6.5.4 Performance of the Example System

To test the performance of the "Unstable Linear Filter ADC", the input signal was estimated by direct input estimation as introduced in Section 2.5.5, where $\sigma_U^2$ was set to the value that was calculated as described in Section 6.5.3.

The input signal was set to

$$u(t) = \sin(2\pi ft)$$

(6.54)

for different frequencies $f$ and the system chosen was a Butterworth filter of order four as described in Section 6.5. The $-3$ dB-frequency $f_c$ was

\textsuperscript{1}In the simulations of this chapter the “Golden Section Search” as described in [23] was used.
Figure 6.8: Empirical SNR dependent on $f$ in the input signals $u(t) = \sin(2\pi ft)$ where the system is a Butterworth of order four with only stable sections. The length of $u(t)$ was 30 times the period $T = 1/f$, where the first and last 5 periods where cut off for the calculation of the SNR.

32·$f_s$. Note that we cannot expect that the algorithm can reconstruct frequencies larger than $f_c$ (cf. Chapter 4), which implies an oversampling ratio of 16. For the system chosen in this section, a lower sampling rate did not result in a stable system (cf. Section 6.3).

Two Butterworth filters were tested, one where all poles are in the left half-plane (thus the system is stable), and one where the poles of the second system where mirrored to the right half-plane and stabilized using feedback paths.

In Figure 6.8 and Figure 6.9 the empirical SNR of the estimate $\hat{u}(t)$ of the input signal is plotted for different frequencies $f$ of the input signal (6.54). The SNR of the estimate $\hat{u}(t)$ is defined as

$$\text{SNR}_{\hat{a}} \triangleq \frac{\mathbb{E}[U(t)^2]}{\mathbb{E}[(U(t) - \hat{U}(t))^2]}.$$  \hspace{1cm} (6.55)

In the plot in Figure 6.8 the Butterworth filter was a stable system, and in Figure 6.9 the second section of the Butterworth filter was unstable. The estimate of the input signal $u(t)$ improves when an unstable section is introduced. Especially if 1-bit quantizers are used to generate the samples $\tilde{y}_k$, the stable system performs very poorly and there is a great benefit of introducing an unstable section.
Figure 6.9: Empirical SNR dependent on \( f \) in the input signals \( u(t) = \sin(2\pi ft) \) where the system is a Butterworth of order four where the second section is unstable. The length of \( u(t) \) was 30 times the period \( T = 1/f \), where the first and last 5 periods where cut off for the calculation of the SNR. The dash-dotted line is the SNR of a simple implementation of the sigma-delta converter.

The performance of a simple implementation of the sigma-delta converter as in Figure 6.1, with only one 1-bit quantizer, is also shown in Figure 6.9. The same input signals (6.54) where used and the bit stream \( \tilde{y}_k \) was filtered with a FIR lowpass filter of order 501. The filter was constructed from an ideal lowpass filter with cut-off frequency \( f_c \), multiplied in the time domain with a Hamming window. This filter is not ideal, but the sigma-delta converter still outperforms the seemingly more complex “Unstable Linear Filter ADC”. Note that an anti-aliasing filter is usually added to a sigma-delta converter, where it can already be included in the architecture of the “Unstable Linear Filter ADC”.

As seen in Figure 6.9 there is only very little improvement by using 2-bit quantizers as opposed to 1-bit quantizers. This is due to the fact that the values of the output of the integrators \( x(t_k) \) rarely reached a value larger than 0.5 (or smaller than \(-0.5\)). The system was designed to never run into saturation as long as the input signal is bounded by 1. Thus, if the input is a simple sine wave the integrators never reach large values. Two approaches to address this problem are:

- Depending on the application, the input signal might not be as nice
as a simple sine wave, and thus the whole range of the quantizers might be used. Simulations should be implemented where the input signal is chosen to be more practical depending on an actual application.

- If the sampling rate is reduced, a larger range of the integrator outputs would be used, but the integrators might sometimes run into saturation where the system is non-linear. If this only happens rarely, it could still be beneficial to reduce the sampling rate.

In this chapter we introduced a general view on the widely used sigma-delta converter. The numerical experiments described in this section show that an “Unstable Linear Filter ADC” can be implemented as proposed and that the model and estimation algorithm introduced in Chapter 2 are directly applicable. Further experiments can now be performed with filters also different from Butterworth filters.
Chapter 7

Conclusion and Future Work

A new way to model a continuous-time signal was introduced, where the signal is viewed as the output of a continuous-time linear system / filter that is driven by continuous-time white Gaussian noise. Thus, the signal is not strictly bandlimited but its spectrum is shaped by the transfer function of the continuous-time linear system / filter. A digital signal can then be viewed as the noisy, discrete-time observations of the continuous-time signal.

We showed how this model can be represented in a Forney factor graph and how message passing algorithms, which essentially result in a Kalman smoother, can be used to calculate an estimate of the noiseless, continuous-time signal at arbitrary time instances. We also showed how an estimate of the input signal of a continuous-time system / filter can be calculated at arbitrary time instances given noisy, discrete-time observations of its output signal. The estimate of the input signal appears to be new and can be viewed as a generalized bandlimited version of the input signal.

We illustrated two applications where the model and algorithms introduced in Chapter 2 can be used in A/D conversion to improve the quality of the digital signal. In Chapter 5 we showed how the error in-
duced by clock jitter can be reduced. In Chapter 6 we introduced a
general view on the sigma-delta converter, where we argued that more
complex unstable linear systems than just integrators can be used in
A/D conversion. One example using a Butterworth filter was described
in detail.

In the simulations implemented in the scope of this thesis, the filter
was always assumed to be known. The effect of model mismatch on
the result was not addressed. Also, the SNR of the noisy, discrete-time
samples was always assumed to be known. Algorithms for the estimation
of the SNR are investigated by Reller in [27]. If and how these algorithms
can be applied to the algorithms introduced in this thesis should be
investigated.

When Butterworth filters of order larger than 8 where implemented,
numerical problems arose and the estimates where not useful. The root
of these problems should be investigated to be able to use the algorithms
for systems with high order filters.

Depending on the application the Butterworth filter might not be
a good choice, especially if the model represents a given piece of hard-
ware. In all the simulations implemented in the scope of this thesis
the continuous-time system was a Butterworth filter. Simulations with
other architectures need to be implemented to get a deeper insight on
the model and algorithms that was proposed in this thesis.
Appendix A

Connections to Control Theory

A.1 Offline Control

A basic problem in control theory is to choose an input signal so that a system follows a desired state trajectory or produces a desired output signal with minimum mean squared error [17].

Assume that a state space representation as described in Chapter 2 of the system is present. For a given target output signal \( \tilde{y}_k \) at discrete time-instances \( t_k \) for \( k = 1, \ldots, K \), the corresponding input signal \( \hat{u}(t) \) can be found by minimizing the cost function

\[
\frac{1}{\sigma^2_U} \int_{t_0}^{t_K} \hat{u}(t)^2 \, dt + \frac{1}{\sigma^2_Z} \sum_{k=1}^{K} (\tilde{y}_k - c^T \tilde{x}(t_k))^2,
\]

which is equivalent to (2.17).

The regularization \( \sigma^2_U / \sigma^2_Z \) determines the trade-off between the input-signal power and the accuracy of the output-signal. If it is the task that the system reaches a defined state trajectory \( \tilde{x}_k \), the state vector \( C \) is set to the unit matrix and then the target “output” signal \( \tilde{y}_k \) in (A.1) is simply set to \( \tilde{y}_k = C^T \tilde{x}_k = \tilde{x}_k \).
A.2 Online Control

In the previous example the input signal $\hat{u}(t)$ is calculated offline, i.e., the whole signal is calculated and then fed to the system. There is no feedback to the controller on how well the target signal is reached. If observations of the system are taken at each time instant $t_k$, the actual output signal can be compared to the target output signal, and the input signal $\hat{u}(t)$ could be adjusted after each sampling instant.

To perform this task the model and algorithm only changes slightly from the example shown above. Assume that the controller reached sampling instant $t_{k^*}$. Observations $\tilde{y}_\text{obs}(t)$ are now present for $t = t_1, \ldots, t_{k^*}$. The new input signal $\hat{u}(t)$ can be found by minimizing the cost function

$$\frac{1}{\sigma_U^2} \int_{t_k^*}^{t_{k^*}} \hat{u}(t)^2 \, dt + \frac{1}{\sigma_{Z_{\text{obs}}}^2} \sum_{k=1}^{k^*} (\tilde{y}_\text{obs}(t_k) - c^T \hat{x}(t_k))^2 + \frac{1}{\sigma_{Z_{\text{tar}}}^2} \sum_{k=(k^*+1)}^{K} (\tilde{y}_\text{tar}(t_k) - c^T \hat{x}(t_k))^2,$$

where $\tilde{y}_\text{tar}(t_k)$ denotes the target output signal and $\sigma_{Z_{\text{obs}}}^2$ denotes the power of the noise under which the observations where taken. The values $\sigma_U^2$ and $\sigma_{Z_{\text{tar}}}^2$ can be used to either put more weight on the input power constraint, or to ensure that the target signal is reached with high accuracy. Note that the integral over $\hat{u}(t)$ is now taken from $t_{k^*}$ since the input signal for $t < t_{k^*}$ has been determined by the controller and is therefore known.

Thus, the factor graph shown in Chapter 2, which is used to minimize the cost function (A.1), only needs to be changed slightly. The observations $\tilde{y}_k$ in the factor are set to $\tilde{y}_\text{obs}(t_k)$ for $k \leq k^*$, and to $\tilde{y}_\text{tar}(t_k)$ for $k > k^*$. The node $f(x(t_{k+1})|x(t_k))$ for $t_{k+1} < t_{k^*}$ is the deterministic function

$$x(t_{k+1}) = e^{AT} x(t_k) + \int_0^T e^{A(T-\tau)} b u(t_k + \tau) \, d\tau,$$

where $u(t)$ is the known input signal, and $T = t_{k+1} - t_k$. 

Note that it is also possible that, e.g., the target is defined as a state trajectory, but the observations are taken from the output signal of the system. The cost function then changes to

\[
\frac{1}{\sigma_U^2} \int_{t_{k^*}}^{t_K} \dot{u}(t)^2 \, dt \\
+ \frac{1}{\sigma_{Z_{obs}}^2} \sum_{k=1}^{k^*} (\tilde{y}_{obs}(t_k) - c^T \hat{x}(t_k))^2 \\
+ \frac{1}{\sigma_{Z_{tar}}^2} \sum_{k=(k^*+1)}^{K} (\tilde{y}_{tar}(t_k) - \hat{x}(t_k))^2. \tag{A.4}
\]

The adjustments to the factor graph are straightforward.

In the special case where the goal is not to reach a defined output signal but to reach a fixed state, the regularization parameter \( \sigma_{Z_{obs}}^2 / \sigma_U^2 \) can be used to either put more weight on the input power constraint, or to allow the system to reach the target state very quickly.
Appendix B

Additional Proofs

B.1 Proof of the Discrete-Time Decomposition in Figure 2.3

We prove the discrete-time decomposition in Figure 2.3 for a non-scalar input, e.g., \( U(t) \in \mathbb{R}^\eta \) as in Section 2.6. The scalar case in Figure 2.3 is then just a special case with \( V_U = \sigma_U^2 \) and \( B = b \).

From (2.40), for a piecewise continuous input signal \( U(t) \) we have

\[
X(t_1) \approx e^{AT}X(t_0) + \sum_{k=1}^{N-1} e^{A(T-kT/N)} B \int_{kT/N}^{(k+1)T/N} U(t_0 + \tau) d\tau, \quad (B.1)
\]

and thus

\[
X(t_1) \approx e^{AT}X(t_0) + \sum_{k=1}^{N} e^{A(T-kT/N)} B \frac{T}{N} U(t_0 + kT/N), \quad (B.2)
\]

where the approximation (B.1) becomes exact in the limit \( N \to \infty \). The factor graph in Figure B.1, represents (B.2). By decomposing \( e^{AT} \) as in Figure B.2, we then obtain Figure B.3.
Appendix B. Additional Proofs

Figure B.1: Discrete-time decomposition (into $N$ sections) of open system (2.38) according to (B.2) with $T \triangleq t_1 - t_0$. The decomposition is exact in the limit $N \to \infty$.

Figure B.2: Decomposition of $e^{AT}$ into $N$ sections.

However, as described in Section 2.6, the integrals in (B.1) are modeled as Gaussian random variables with mean zero and covariance matrix $V_UT/N$. The corresponding factor graph of (B.1) is shown in Figure B.4 (where we have used the decomposition of $e^{AT}$ as in Figure B.3). From Figure B.4, we obtain Figure 2.3 by setting $B = b$ and $V_U = \sigma_U^2$, and moving scale factors around so that the snapshots of the continuous-time signal $U(t)$ appear with the same scaling as the input signal in Figure B.3.
B.1. Proof of the Discrete-Time Decomposition in Figure 2.3

\[ X(t_0) e^{ATN} + B^T \mathcal{N}(0, VU^{T/N}) + \ldots + e^{ATN} + B^T \mathcal{N}(0, VU^{T/N}) = X(t_1) \]

**Figure B.3:** Discrete-time decomposition (into \( N \) sections) of open system (2.1) obtained from Figure B.1 and Figure B.2.

\[ N(0, VU^{T/N}) \]

\[ \tilde{U}_1 \]

\[ B \]

\[ X(t_0), e^{ATN} + \ldots + e^{ATN} = X(t_1) \]

**Figure B.4:** Discrete-time decomposition (into \( N \) sections) of the noise-driven system according to (B.1) provided that \( U(t) \) is white Gaussian noise as described in Section 2.6. The decomposition is exact in the limit \( N \to \infty \).
B.2 Derivation of $\tilde{\Theta}(t)$ and $\tilde{\Theta}(t)$ in (2.14), (2.16), (2.43) and (2.44)

Let

$$\Lambda \triangleq \begin{pmatrix} \lambda_1 & 0 \\ \vdots & \ddots \\ 0 & \lambda_n \end{pmatrix}.$$  \hspace{1cm} (B.3)

From (2.12), we have

$$e^{A\tau} = Qe^{\Lambda\tau}Q^{-1}$$ \hspace{1cm} (B.4)

and

$$e^{A^{T}\tau} = (e^{A\tau})^{T} = (e^{\Lambda\tau})^{H} = (Q^{-1})^{H}e^{\Lambda^{T}\tau}Q^{H},$$ \hspace{1cm} (B.5)

and thus

$$\int_{0}^{t} e^{A\tau}BVU^{T}e^{A^{T}\tau}dt = Q \left( \int_{0}^{t} e^{\Lambda\tau}\Psi e^{\Lambda^{T}\tau}dt \right) Q^{H} \hspace{1cm} (B.6)$$

with

$$\Psi \triangleq Q^{-1}BVU(Q^{-1}B)^{H}. \hspace{1cm} (B.7)$$

The element in row $k$ and column $\ell$ of the matrix under the integral is

$$\left( e^{\Lambda\tau}\Psi e^{\Lambda^{T}\tau} \right)_{k,\ell} = \psi_{k,\ell} e^{(\lambda_{k}+\lambda_{\ell})\tau}, \hspace{1cm} (B.8)$$

where $\psi_{k,\ell}$ refers to the elements of the matrix $\Psi$, and elementwise integration yields

$$\left( \int_{0}^{t} e^{\Lambda\tau}\Psi e^{\Lambda^{T}\tau}dt \right)_{k,\ell} = \frac{\psi_{k,\ell}}{\lambda_{k} + \lambda_{\ell}} \left( e^{(\lambda_{k}+\lambda_{\ell})t} - 1 \right), \hspace{1cm} (B.9)$$

which validates definition (2.43). In the definition (2.14) the input signal is scalar and thus $V_{U} = \sigma_{U}^{2}$ and $B = b$.

The definitions (2.16) and (2.44) follow from noting that changing $e^{A\tau}$ into $e^{-A\tau}$ amounts to a sign change of $\Lambda$. 

Consider the factor graph in Figure B.5, which shows the relevant part of Figure 2.3 for a non-scalar input signal augmented with additional named variables. From [4, eq. (54) and (III.5)], we have

\[
W_U(t) = \hat{W}'_U(t) + \hat{W}'_U(t) - \nabla_U \hat{m}_U(t) B_T \hat{W}'_U(t) B; \tag{B.10}
\]

\[
W_U(t) = V_U^{-1} \frac{T}{N} + \left( \frac{T}{N} \right)^2 B_T \hat{W}'_U(t) B; \tag{B.11}
\]

taking the limit \( N \to \infty \) yields \( W_U(t) = 0 \) (and thus the covariance matrix of \( U(t) \) diverges).

From [4, eq. (55)], we have

\[
W_U(t) m_U(t) = \hat{W}'_U(t) \hat{m}_U(t) + \hat{W}_U(t) \hat{m}_U(t); \tag{B.12}
\]

inserting \( \hat{m}_U(t) = 0 \) and using [4, eq. (III.6)] yields

\[
W_U(t) m_U(t) = T \frac{B_T \hat{W}'_U(t) \hat{m}_U(t)}{N}. \tag{B.13}
\]

Using (B.11) and (B.13), we obtain

\[
m_U(t) = (W_U(t))^{-1} (W_U(t) m_U(t)) \tag{B.14}
\]
and the approximation (B.16) becomes exact in the limit $N \to \infty$.

Using [4, eq. (II.10)], we have

$$\hat{m}_{U(t)} = \hat{m}_{X(t)} - \hat{m}_{X'(t)}$$

and using [4, eq. (II.8)], we have

$$\hat{W}_{U'(t)} = \left(\hat{V}_{U'(t)}\right)^{-1}$$

Again, the approximations (B.18) and (B.21) both become exact in the limit $N \to \infty$. Inserting (B.18) and (B.21) into (B.16) yields

$$m_{U(t)} = V_{U} B^T \left(V_{X(t)} + \hat{V}_{X(t)}\right)^{-1} (\hat{m}_{X(t)} - \hat{m}_{X(t)})$$

and since $\hat{u}(t) \triangleq m_{U(t)}$ is the LMMSE estimate of $U(t)$ this proves (ii.ii.5) and (ii.i.5) is just a special case of (ii.ii.5) with $V_{U} = \sigma_{U}^2$ and $B = b$.

### B.4 Proof of (2.17) and (2.48)

Recall the factor graph representation of a least squares problem as in Figure B.6, where the large box on top expresses the given constraints. Clearly, maximizing the function represented by Figure B.6 amounts to computing

$$\arg\max_{z_1, \ldots, z_n} \prod_{k=1}^n e^{-z_k V_k^{-1} z_k^T} = \arg\min_{z_1, \ldots, z_n} \sum_{k=1}^n z_k V_k^{-1} z_k^T$$

(B.23)
subject to the constraints. The right-hand side of (B.23) will be called “cost function.” Recall that sum-product message passing in cycle-free linear Gaussian factor graphs maximizes the left-hand side of (B.23) (subject to the constraints) and thus minimizes the cost function [4].

Now plugging Figure 2.3 into the factor graph in Figure 2.1 results in a factor graph as in Figure B.6 with cost function

$$
\sum_{k=1}^{K} \left( z_k V_Z^{-1} z_k^T + \sum_{\ell=1}^{N} u(t_{k-1} + \ell \frac{T_k}{N}) (V_U)^{-1} \left( u(t_{k-1} + \ell \frac{T_k}{N}) \right)^T \frac{T_k}{N} \right)
$$

with $T_k \triangleq t_k - t_{k-1}$. Taking the limit $N \to \infty$, (B.24) becomes

$$
\sum_{k=1}^{K} \left( z_k V_Z^{-1} z_k^T + \int_{t_{k-1}}^{t_k} u(t) (V_U)^{-1} (u(t))^T dt \right),
$$

which is (2.48). For scalar $z_k$ and $u(t)$, just substitute $V_Z = \sigma_Z^2$ and $V_U = \sigma_U^2$ to obtain (2.17).
Appendix C

The Butterworth Filter

The Butterworth filter is easy to construct and has a nice transfer function. In this section we summarize the construction and properties of the Butterworth filter [28]. A Butterworth filter of order $N$ has $N$ poles $p_n$, where

$$p_n = 2\pi f_c e^{i \frac{\pi}{N} (2n+N-1)},$$  \hspace{1cm} (C.1)

as illustrated in Figure C.1.

Figure C.1: Poles of a Butterworth of order $N = 4$ (left) and $N = 8$ (right).
The Fourier spectrum of the Butterworth filter has the property

\[ |\mathcal{F}_H(2\pi if)|^2 = \frac{1}{1 + (f/f_c)^{2N}}. \] (C.2)

See Figure C.2 for some examples of the amplitude of the Fourier spectrum of a Butterworth filter. The amplitude is plotted in dB (i.e. \(10 \cdot \log_{10}(|\mathcal{F}_H(2\pi if)|^2)\)).

In Figure C.2, it is clearly visible how the amplitude response of the Butterworth filter is very flat for frequencies smaller than \(f_c\). The damping at \(f = f_c\) is \(-3\) dB, and thus \(f_c\) is also called the \(-3\) dB-frequency. For frequencies larger than \(f_c\) the damping depends on the filter order.
Appendix D

Additions to Chapter 6

D.1 The Sigma-Delta Converter with a Pre-Filter

In Section 6.2, an implementation of the sigma-delta converter is shown where the unstable filter is a single integrator. If the input estimation as in Section 6.4 is preformed for a single integrator, the estimate of the input signal $U(t)$ is useless because $c^Tb \neq 0$ (cf. Section 2.5.5).

A useful estimate still can be found by adding a prefilter to the sigma-delta converter (which can either model an anti-aliasing filter which is also implemented in hardware, or additional spectral shaping as introduced in Section 4.3).

Let the prefilter be

\[
\dot{x}_1(t) = A_1 x_1(t) + b_1 u(t) \quad \text{(D.1)}
\]
\[
y_1(t) = c_1^T x_1(t). \quad \text{(D.2)}
\]

The system with the single integrator looks like

\[
\dot{x}_2(t) = y_1(t) \quad \text{(D.3)}
\]
\[
y(t) = x_2(t). \quad \text{(D.4)}
\]
Thus the total system with the prefilter and the integrator is

\[ \dot{x}(t) = \begin{pmatrix} A_1 & 0 \\ c_1 & 0 \end{pmatrix} + \begin{pmatrix} b_1 \\ 0 \end{pmatrix} u(t) \] (D.5)

\[ y(t) = \begin{pmatrix} 0, \ldots, 0, 1 \end{pmatrix}^\top x(t). \] (D.6)

The new system matrix \( A_{\text{tot}} \) cannot be diagonalized as in (2.12). Thus, if the input reconstruction as introduced in Chapter 2 is implemented, the messages cannot be calculated as described in Section 2.4.

Nevertheless, the integral in (2.13)

\[ \int_0^t e^{A_{\text{tot}} \tau} b_{\text{tot}} b_{\text{tot}}^\top e^{A_{\text{tot}}^\top \tau} d\tau \] (D.7)

can be calculated. The matrix exponential in the integral is

\[ e^{A_{\text{tot}} \tau} = e^{\begin{pmatrix} A_1 & 0 \\ c_1 & 0 \end{pmatrix} \tau} \] (D.8)

\[ = \sum_{k=0}^{\infty} \frac{1}{k!} \begin{pmatrix} A_1 \tau & 0 \\ c_1 \tau & 0 \end{pmatrix}^k \] (D.9)

\[ = \begin{pmatrix} I_n & 0 \\ 0 & 1 \end{pmatrix} + \sum_{k=1}^{\infty} \frac{1}{k!} \begin{pmatrix} A_1 \tau & 0 \\ c_1 \tau & 0 \end{pmatrix}^k \] (D.10)

\[ = \begin{pmatrix} I_n & 0 \\ 0 & 1 \end{pmatrix} + \sum_{k=1}^{\infty} \frac{1}{k!} \begin{pmatrix} A_1^k \tau^k & 0 \\ c_1 A_1^{k-1} \tau^k & 0 \end{pmatrix} \] (D.11)

\[ = \begin{pmatrix} I_n & 0 \\ 0 & 1 \end{pmatrix} + \sum_{k=1}^{\infty} \frac{1}{k!} \begin{pmatrix} (A_1 \tau)^k & 0 \\ c_1 A_1^{-1} (A_1 \tau)^k & 0 \end{pmatrix} \] (D.12)

\[ = \begin{pmatrix} 0 & 0 \\ -c_1 A_1^{-1} & 1 \end{pmatrix} + \sum_{k=0}^{\infty} \frac{1}{k!} \begin{pmatrix} (A_1 \tau)^k & 0 \\ c_1 A_1^{-1} (A_1 \tau)^k & 0 \end{pmatrix} \] (D.13)

\[ = \begin{pmatrix} c_1 A_1^{-1} (e^{A_1 \tau} - I_n) & 0 \\ 0 & 1 \end{pmatrix}, \] (D.14)

where \( I_n \) is the unit matrix in \( \mathbb{R}^{n \times n} \).
If we substitute
\begin{align}
P_1(\tau) & \triangleq e^{A_1 \tau} \\
P_2(\tau) & \triangleq c_1 A_1^{-1} \left( e^{A_1 \tau} - I_n \right),
\end{align}
the integral (D.7) is
\begin{align}
\int_0^t \begin{pmatrix} P_1(\tau) & 0 \\ P_2(\tau) & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ 0 \end{pmatrix} \begin{pmatrix} b_1^T & 0 \end{pmatrix} \begin{pmatrix} P_1(\tau)^T & P_2(\tau)^T \\ 0 & 1 \end{pmatrix} d\tau
\end{align}
\begin{align}
= \int_0^t \begin{pmatrix} P(\tau)b_1b_1^T P_1(\tau)^T & P_1(\tau)b_1b_1^T P_2(\tau)^T \\ P_2(\tau)b_1b_1^T P_1(\tau)^T & P_2(\tau)b_1b_1^T P_2(\tau)^T \end{pmatrix} d\tau.
\end{align}

If we insert the definitions of \( P_1(\tau) \) and \( P_2(\tau) \) and expand everything, two types of integrals appear. Some integrals are in the form
\begin{align}
\int_0^t e^{A_1 \tau} b_1b_1^T e^{A_1^T \tau} d\tau,
\end{align}
which can be calculated as described in Section 2.4. Other integrals appear in the form
\begin{align}
C \int_0^t e^{A_1 \tau} d\tau = CA_1^{-1} \left( e^{A_1 t} - I_n \right),
\end{align}
and thus, the integral (D.7) can be solved, and the message passing rules can be calculated.
Bibliography


Series in Signal and Information Processing
edited by Hans-Andrea Loeliger


Vol. 5: Markus Erne, Signal Adaptive Audio Coding using Wavelets and Rate Optimization. ISBN 3-89649-625-5


Vol. 7: Heinz Mathis, Nonlinear Functions for Blind Separation and Equalization. ISBN 3-89649-728-6

Vol. 8: Daniel Lippuner, Model-Based Step-Size Control for Adaptive Filters. ISBN 3-89649-755-3

Vol. 9: Ralf Kretzschmar, A Survey of Neural Network Classifiers for Local Wind Prediction. ISBN 3-89649-798-7


Vol. 11: Pascal O. Vontobel, Algebraic Coding for Iterative Decoding. ISBN 3-89649-865-7

Vol. 12: Qun Gao, Fingerprint Verification using Cellular Neural Networks. ISBN 3-89649-894-0


Hartung-Gorre Verlag Konstanz – http://www.hartung-gorre.de