Optimization of Coding of AR Sources for Transmission Across Channels with Loss

Arildsen, Thomas

DOI (link to publication from Publisher):
10.6084/m9.figshare.709006

Publication date:
2010

Document Version
Accepted author manuscript, peer reviewed version

Aalborg University
Department of Electronic Systems
Multimedia Information and Signal Processing
Niels Jernes Vej 12, A6-3
DK-9220 Aalborg Ø

Ph.D Thesis

Optimization of Coding of AR Sources for Transmission Across Channels with Loss

Thomas Arildsen
<tha@es.aau.dk>

Aalborg, 12. januar 2011

Arildsen, Thomas

Optimization of Coding of AR Sources for Transmission Across Channels with Loss

Aalborg University
Department of Electronic Systems
Multimedia Information and Signal Processing
Niels Jernes Vej 12, A6-3
DK-9220 Aalborg Ø

Supervisors:

Prof., Ph.D.	Søren Holdt Jensen	Aalborg University
Associate Prof., Ph.D.	Manohar N. Murthi	University of Miami
Director of Research, Ph.D.	Søren Vang Andersen	Skype

This thesis was typeset using LaTeX and the Memoir class.

Preface

I began working on the Ph.D. study that forms the basis of this thesis in April 2006. The research was initiated as part of the research project "Cross-Layer Design for Multi-Media Applications on Mobile Phones" (aka "X3MP") headed by Associate Professors Frank H. P. Fitzek and Søren Vang Andersen, the latter then part owner and manager of the VoIP company Sonorit in Nørresundby; Søren Vang Andersen was my main supervisor. Shortly after commencing the research project, Søren Vang Andersen's company Sonorit was acquired by Skype, after which he had to scale down his involvement in the X3MP research project at AAU. Consequently, Søren Holdt Jensen took over the role as main supervisor, and I turned the topic of my research in a slightly different direction than originally planned. Søren Vang Andersen remained associated as a co-supervisor.

Associate Professor Manohar N. Murthi from University of Miami was associated as co-supervisor in 2006. In the winter of 2007, I visited University of Miami as a visiting researcher under Assoc. Prof. Murthi. In late 2007, my wife and I became the proud parents of our first son and consequently, I was away on parental leave from my Ph.D. position during two periods of 2008. This prolonged my Ph.D. study by approximately three months.

At the end of my original Ph.D. scholarship period mid-2009, I was generously awarded an extension of my employment as a Ph.D. student by my supervisor Søren Holdt Jensen through the end of 2009. By the end of 2009, the research of my Ph.D. study was concluded. I finished the thesis on the side while working in a new position in a research project formed as a joint effort between the Multimedia Information and Signal Processing (MISP) Section and the Technology Platforms Section (TPS) at AAU.

I would like to express my sincere gratitude to the supervisors involved in my Ph.D. study: Søren Vang Andersen, Manohar N. Murthi, Søren Holdt Jensen, and Frank H. P. Fitzek as well as my collaborators on several publications and numerous colleagues of the MISP Section. I would also like to dedicate this thesis to my wonderful wife Sidsel and my two sons Malthe and Asger and especially thank my wife for her patience and support during my Ph.D. study.

Aalborg, 2010

Abstract

Source coding concerns the representation of information in a source signal using as few bits as possible. In the case of lossy source coding, it is the encoding of a source signal using the fewest possible bits at a given distortion or, at the lowest possible distortion given a specified bit rate. Channel coding is usually applied in combination with source coding to ensure reliable transmission of the (source coded) information at the maximal rate across a channel given the properties of this channel.

In this thesis, we consider the coding of auto-regressive (AR) sources which are sources that can be modeled as auto-regressive processes. The coding of AR sources lends itself to linear predictive coding. We address the problem of joint source/channel coding in the setting of linear predictive coding of AR sources. We consider channels in which individual source coded signal samples can be lost during channel transmission. The optimization of linear predictive coding for such lossy channel behaviour is not well understood in the literature.

We review basics of source and channel coding, differential pulse code modulation (DPCM), state-space models, minimum mean squared error (MMSE) estimation, and quantization. On this background we propose a new algorithm for optimization of predictive coding of AR sources for transmission across channels with loss.

The optimization algorithm takes as its starting point a re-thinking of the source coding operation as an operation producing linear measurements of the source signal. The source process and source encoder are formulated as a state-space model, enabling the use of Kalman filtering for decoding the source signal.

The optimization algorithm is a greedy approach that designs the filter coefficients of a generalized DPCM encoder. The objective of the optimization problem (design of the filter coefficients) is to minimize the decoder state error covariance. This is done iteratively in a greedy sense, minimizing the trace of the state error covariance at each iteration until convergence.

Furthermore, it is proved that employing fixed-lag smoothing at the decoder is guaranteed to reduce the estimated source signal mean squared error (MSE) under mild constraints on the encoder filter coefficients.

Extensive Monte Carlo simulation studies show that the proposed algorithm improves the signal-to-noise ratio (SNR) of decoded source signals substantially compared to the case where the encoder is unaware of channel loss.

We finally provide an extensive overview of cross-layer communication issues which are important to consider due to the fact that the proposed algorithm interacts with the source coding and exploits channel-related information typically available from different layers of network protocol stacks.

Resumé

Kildekodning omhandler repræsentation af informationen i et kildesignal med færrest mulige bits. Ved kildekodning med tab, er målet at kode kildesignalet med færrest mulige bits ved et givet niveau af forvrængning, eller med mindst mulig forvrængning givet en bestemt bitrate. Kanalkodning benyttes typisk i sammenhæng med kildekodning for at sikre pålidelig transmission af den (kildekodede) information ved den maksimale rate gennem en kanal, givet kanalens egenskaber.

I denne afhandling beskæftiger vi os med kodning af auto-regressive (AR) kilder, som er kilder, der kan modelleres som auto-regressive processer. Lineær prediktiv kodning passer naturligt til kodning af AR kilder. Vi behandler problemstillingen om kombineret kilde- og kanalkodning indenfor rammerne af lineær prediktiv kodning af AR kilder. Vi betragter kanaler, i hvilke de individuelle kodede signalværdier kan gå tabt under transmission gennem kanalen. Optimering af lineær prediktiv kodning til denne type kanal er ikke velbeskrevet i litteraturen.

Vi gennemgår de basale elementer af kilde- og kanalkodning, differential pulse code modulation (DPCM), state-space modeller, minimum mean squared error (MMSE)-estimation og kvantisering. På denne baggrund foreslår vi en ny algoritme til optimering af prediktiv kodning af AR kilder til transmission gennem kanaler med tab.

Optimeringsalgoritmens udgangspunkt er en gentænkning af kildekodningen som en operation, der tager lineære målinger af kildesignalet. Kildeprocessen og kildekoderen formuleres som en state-space model, hvilket muliggør brugen af Kalman-filtrering til dekodning af signalet.

Optimeringsalgoritmen bruger en såkaldt "grådig" tilgang til optimeringen, som designer filterkoefficienterne til en generaliseret DPCM-koder. Målet for optimeringsproblemet (design af filterkoefficienterne) er at minimere dekoderens kovarians af state-fejlen. Dette gøres iterativt i en "grådig" forstand, hvilket minimerer trace af kovariansen af state-fejlen i hver iteration indtil algoritmen konvergerer.

Endvidere bevises det, at brugen af estimations-"udglatning" med en fast forsinkelse i dekoderen garanterer, at kvadratet på fejlen i det dekodede kildesignal reduceres i gennemsnit (mean squared error (MSE)) under milde

krav til kildekoderens filterkoefficienter.

Omfattende Monte Carlo-simulationer viser, at den foreslåede algoritme forbedrer signal-støj-forholdet (SNR) af dekodede kildesignaler betragteligt i sammenligning med det tilfælde, hvor kildekoderen ikke har kendskab til kanaltabene.

Endelig giver vi et omfattende overblik over "cross-layer" kommunikation, som er et vigtigt aspekt at holde sig for øje pga. det faktum, at den foreslåede algoritme både interagerer med kildekodningen og udnytter kanal-tabsinformation, som typisk er tilgængelige på forskellige lag i netværksprotokolstakke.

List of Publications

This is a complete list of publications published in the course of the Ph.D. work presented in this thesis. The following papers and book chapter are included in the thesis:

A Thomas Arildsen, Manohar N. Murthi, Søren Vang Andersen, and Søren Holdt Jensen. „On Predictive Coding for Erasure Channels Using a Kalman Framework". In: *IEEE Transactions on Signal Processing* 57.11 (Nov. 2009), pp. 4456–4466. ISSN: 1053-587X. DOI: 10.1109/TSP.2009.2025796

B Thomas Arildsen, Manohar N. Murthi, Søren Vang Andersen, and Søren Holdt Jensen. „On Predictive Coding for Erasure Channels Using a Kalman Framework". In: *Proceedings of the 17th European Signal Processing Conference (EUSIPCO-2009)*. Eurasip. EUSIPCO 2009, Glasgow Ltd, Aug. 2009

C Thomas Arildsen, Jan Østergaard, Manohar N. Murthi, Søren Vang Andersen, and Søren Holdt Jensen. „Fixed-Lag Smoothing for Low-Delay Predictive Coding with Noise Shaping for Lossy Networks". In: *Proceedings of Data Compression Conference (DCC-2010)*. Los Alamitos, CA: IEEE Computer Society, Mar. 2010, pp. 279–287. DOI: 10.1109/DCC.2010.33

D Thomas Arildsen and Frank H. P. Fitzek. „Cross Layer Protocol Design for Wireless Communication". In: *Mobile Phone Programming and its Application to Wireless Networking*. Ed. by Frank H. P. Fitzek and Frank Reichert. 1st ed. Dordrecht, The Netherlands: Springer, 2007. Chap. 17, pp. 343–362. ISBN: 978-1-4020-5968-1. DOI: 10.1007/978-1-4020-5969-8_17

The following book chapters are not included in the thesis:

- Thomas Arildsen and Morten L. Jørgensen. „Qtopia Greenphone". In: *Mobile Phone Programming and its Application to Wireless Networking*. Ed. by Frank H. P. Fitzek and Frank Reichert. 1st ed. Dordrecht,

The Netherlands: Springer, 2007. Chap. 6, pp. 159–174. ISBN: 978-1-4020-5968-1. DOI: 10.1007/978-1-4020-5969-8_6

- Thomas Arildsen and Frank H. P. Fitzek. „The C-Cube Concept : Combining Cross-Layer Protocol Design, Cognitive-, and Cooperative Network Concepts“. In: *Cognitive Wireless Networks: Concepts, Methodologies and Visions Inspiring the Age of Enlightenment of Wireless Communications*. Ed. by Frank H. P. Fitzek and Marcos Katz. 1st ed. Dordrecht, The Netherlands: Springer, 2007. Chap. 21, pp. 423–433. ISBN: 978-1-4020-5978-0. DOI: 10.1007/978-1-4020-5979-7_21

Contents

Chapter 1

Introduction

1.1 Coding System Overview

In 1948, Shannon published his landmark paper on information theory [1]. A main result of this work ("The Fundamental Theorem for a Discrete Channel with Noise") is the fact that if a discrete source has an entropy below the channel capacity, there exists a coding system that allows it to be transmitted through the channel with arbitrarily small error; conversely, if the source has an entropy greater than the channel capacity, it cannot be transmitted reliably through the channel. Since the publication of Shannon's results, it has been common to consider coding/communication systems of the structure depicted in Figure 1.1. A consequence of the results of [1] is that

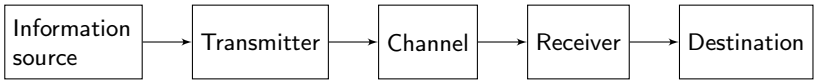


Figure 1.1: General communication system structure.

reliable transmission of a source can in principle be achieved by separate source and channel coders. The source coder can remove redundancy from the source signal without considering the channel characteristics while the channel coder adds redundancy appropriate for the channel to the resulting coded source signal without considering the characteristics of the original source signal. This principle is popularly known as the source-channel separation principle [2]. It corresponds to the structure depicted in Figure 1.2.

There are examples where the above-mentioned theorem from [1] does not hold. One often-mentioned example in the literature is multiple-access communications [3]. The shortcomings of the theory are also demonstrated in [2]. We shall not get into further details on the matter here, but recent

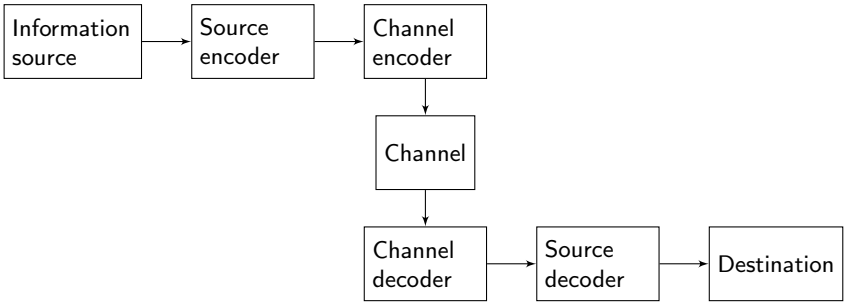


Figure 1.2: General communication system structure with separate source/channel coding.

work on generalization of Shannon's famous theory can be found in [2, 4, 5].

Another detail of Shannon's theory is that the probability of correctly decoding a transmitted source sequence of duration T goes to 1 as $T \to \infty$ [1]. In practice this limits the applicability of the source-channel separation principle, especially in the case of communications with low delay requirements, since it imposes a coding/transmission delay.

Because of the mentioned shortcomings of the theory and the sequence length requirement, joint source-channel coding approaches are actively investigated despite the fact that Shannon's theorem at first glance suggests that source and channel coding should be separated.

1.2 Differential Pulse Code Modulation

A well-known, classic source coding technique is differential pulse code modulation (DPCM), dating back to 1952 [6, 7]. DPCM is a form of linear predictive coding employed in a feedback quantization scheme. The original principle is illustrated in Figure 1.3.

The principle behind DPCM is to remove redundancy from a source signal s_n containing temporal correlation. This is done by forming an Nth order linear prediction $\hat{s}_n$–i.e., a linear estimate of s_n from past samples $\{s_i | i = n-1, \ldots, n-N\}$–and subtracting the prediction from the source sequence s_n

$$d_n = s_n - \hat{s}_n. \tag{1.1}$$

The prediction residual d_n will contain less information than the original source sequence s_n and can thus be quantized more efficiently.

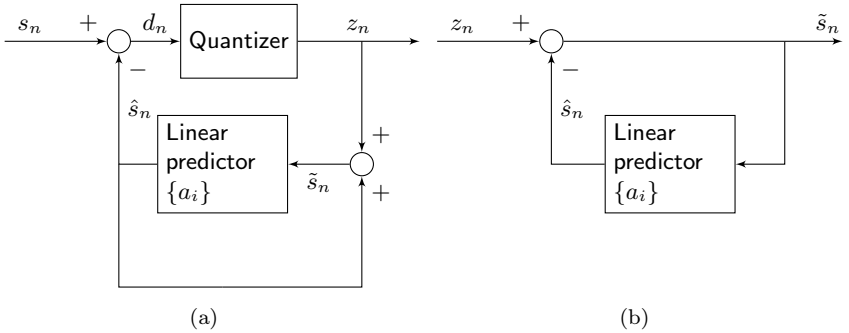


Figure 1.3: DPCM: (a) encoder and (b) decoder.

Ideally, the linear prediction would be calculated from past samples of the original source sequence

$$\hat{s}_n = \sum_{i=1}^{N} a_i s_{n-i}. \tag{1.2}$$

However, the original source sequence will not be available to the decoder which would consequently not be able to reconstruct the source signal. Therefore, the prediction is calculated from a reconstruction $\tilde{s}_n$ of the source signal based on the quantized prediction residual z_n

$$z_n = Q(d_n) \tag{1.3}$$

$$\tilde{s}_n = \hat{s}_n + z_n \tag{1.4}$$

$$\hat{s}_n = \sum_{i=1}^{N} a_i \tilde{s}_{n-i}, \tag{1.5}$$

where $\tilde{s}_n$ is also the output of the decoder, shown as $\tilde{s}_n$ in fig. 1.3b.

Selection of the predictor coefficients as treated in for example [8, Sec. 6.3] is based on the assumption that the quantization noise is negligible and the calculation of the optimal coefficients is basically developed from the approximation $\tilde{s}_n \approx s_n$. This for example leads to the result that the optimal predictor coefficients for an auto-regressive (AR) source signal are equal to the filter coefficients of the source process.

The DPCM coding scheme, although simple in structure, turns out to be quite difficult to design optimally. This is due to the non-linear quantizer and the recursive nature of DPCM where the optimal quantizer will depend on the source and the predictor, and the optimal predictor will depend on the source and quantizer. Thus the predictor and quantizer must

be optimized together, which turns out to be very challenging. An extensive rate-distortion analysis of DPCM and optimization of the quantizer is presented in [9]. Optimization of the quantizer for a given predictor and source is treated in [10] which finds that the optimal predictor does not generally match the source. That is, this contradicts the result mentioned above, stemming from simpler assumptions, that the predictor coefficients for an AR source signal equal the coefficients of the source process.

DPCM also turns out not to be very (rate-distortion) efficient for coding at low bit rates. An attempt at solving this is made in, e.g., [11] by using pre- and post-filters and quantization noise shaping. The optimal predictor for Gaussian sources is found in [12] which is the first paper to show how the rate-distortion function can be achieved at all bit rates. The issue of DPCM's low-rate efficiency is also addressed in [13] which employs pre-filtering and down-sampling to improve rate-distortion performance for sources with monotonic spectra. In [13] it is also demonstrated how DPCM, through remarkably simple modifications, still proves to be a very competitive and simple coding technique compared to more modern and complex coding approaches.

Stability of a DPCM coding system has been investigated in [14].

1.2.1 Noise Shaping Structure

A useful generalization of the basic DPCM encoder structure shown in fig. 1.3a is shown in fig. 1.4 [15–17]. This structure enables shaping of the quantization noise in the encoder separately from the prediction of the source signal. The structure depicted in fig. 1.4 offers more degrees of freedom in the encoder design and a convenient structure for modeling, which will be discussed in Section 1.6. It is easily verified, for example by rearrang-

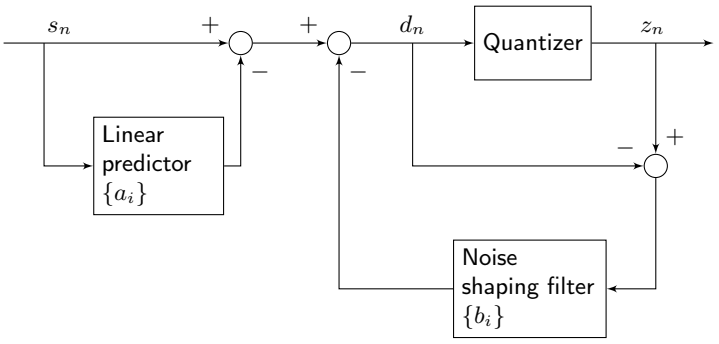


Figure 1.4: Generalized DPCM encoder.

ing the block diagram of fig. 1.4, that the structures depicted in figs. 1.3a

and 1.4 are equivalent when the noise shaping filter in fig. 1.4 equals the linear predictor, i.e., $a_i = b_i,\ \forall i$.

1.2.2 Coding of Auto-Regressive Sources

DPCM lends itself naturally to coding of AR sources. This is closely linked to the structure of the linear prediction employed (1.2). Observe that the linear prediction (1.2) has the same structure as an AR process, with zero input. Consider an AR process

$$s_n = \sum_{i=1}^{N} \alpha_i s_{n-i} + e_n, \tag{1.6}$$

where $e_n \sim \mathcal{N}(0, \sigma_e^2)$. The prediction $\hat{s}_n$ from s_{n-i}, $i \in [1, N]$ shown in (1.2) results in the prediction residual (1.7). As pointed out earlier, the prediction defined in (1.2) is not useful in practice, but the following derivation serves to demonstrate why DPCM is particularly suitable for AR sources

$$\begin{aligned} d_n &= s_n - \hat{s}_n \\ &= \sum_{i=1}^{N} \alpha_i s_{n-i} + e_n - \sum_{i=1}^{N} a_i s_{n-i} \\ &= e_n \text{ for } \alpha_i = a_i,\ \forall i. \end{aligned} \tag{1.7}$$

The prediction residual $d_n = e_n$ is optimal in the sense that it is white Gaussian and thus contains no residual redundancy.

The fact that DPCM coding is well suited for AR sources has resulted in wide application of DPCM in particularly voice coding, because an AR model models especially the vowel parts of human speech well [18].

1.2.3 Error Propagation

As pointed out in for example [8, Sec. 6.7], DPCM is subject to error propagation. Considering (1.4), it is evident that a transmission error, i.e., some type of error in the received z_n will affect the immediate decoded sample $\tilde{s}_n$. Furthermore, the effect of the error on $\tilde{s}_n$ will in principle degrade all future decoded samples $\tilde{s}_{n+i},\ i = 1, \ldots, \infty$ because of the auto-regressive nature of the decoder, cf. (1.4)-(1.5).

As pointed out in [8, Sec. 6.7.2], the propagation of channel errors can be reduced by choosing $a_1 < \alpha_1$ for a first-order predictor. However, it should be pointed out as well that this will also lead to less efficient prediction. This is discussed further in Section 1.5.3.

1.3 State-Space Modeling

A broad class of linear systems can be modeled by state space models. The work presented in Publications A to C make extensive use of state-space models of AR processes. Some basics of state-space models are therefore reviewed in this section.

A state space system can be represented very generally as[1]

$$\mathbf{x}_{n+1} = \mathbf{F}_n \mathbf{x}_n + \mathbf{G}_{1,n} \mathbf{w}_n \tag{1.8a}$$
$$\mathbf{y}_n = \mathbf{H}_n \mathbf{x}_n + \mathbf{G}_{2,n} \mathbf{v}_n. \tag{1.8b}$$

The state $\mathbf{x}_n \in \mathbb{R}^K$ evolves over time n according to (1.8a), where $\mathbf{F}_n \in \mathbb{R}^{K \times K}$ is called the state update matrix. $\mathbf{w}_n \in \mathbb{R}^{L_1}$ is typically known as the process noise and is a sequence of outcomes of a stochastic process. $\mathbf{G}_{1,n} \in \mathbb{R}^{K \times L_1}$ is a transform matrix which can for example be used to impose a specific structure on $\mathbf{G}_{1,n}\mathbf{w}_n$. The state can be observed through measurements $\mathbf{y}_n \in \mathbb{R}^M$ taken according to (1.8b), known as the measurement or observation equation. $\mathbf{H}_n \in \mathbb{R}^{M \times K}$ is typically known as the measurement or observation matrix. $\mathbf{v}_n \in \mathbb{R}^{L_2}$ is the measurement noise and is a sequence of outcomes of a stochastic process. Again, $\mathbf{G}_{2,n} \in \mathbb{R}^{K \times L_2}$ is a transform matrix introduced for convenience. No further assumptions on the elements of (1.8a) and (1.8b) are made for a general state-space model.

1.3.1 Auto-Regressive Processes

A (stationary) Nth order AR process, cf. Section 1.2.2, can be modeled by a state space model of the form (1.8) as follows

$$\mathbf{x}_{n+1} = \mathbf{F}\mathbf{x}_n + \mathbf{G}w_n \tag{1.9a}$$
$$z_n = \mathbf{h}^{\mathrm{T}}\mathbf{x}_n + v_n, \tag{1.9b}$$

[1] One can find numerous variations on the standard state-space formulation. Specifically, state-space models often include a known input $\mathbf{u}_n$ in the process equation (1.8a). This input is typically used in control applications and is omitted in this thesis to simplify the equation.

where

$$\mathbf{F} = \begin{bmatrix} -\alpha_1 & & \dots & & -\alpha_N \\ 1 & 0 & \dots & 0 & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix}_{(N\times N)} \quad \mathbf{x}_n = \begin{bmatrix} z_{n-1} \\ \vdots \\ z_{n-N} \end{bmatrix}_{(N\times 1)}$$

$$\mathbf{G} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{(N\times 1)} \quad \mathbf{h} = \begin{bmatrix} -\alpha_1 \\ \vdots \\ -\alpha_N \end{bmatrix}_{(N\times 1)}, \tag{1.10}$$

$w_n \sim \mathcal{N}(0, \sigma_w^2)$, and $v_n = w_n$. It is easily verified from the definitions of $\mathbf{F}$, $\mathbf{G}$, $\mathbf{h}$, and $\mathbf{x}_n$ that (1.9) represents an AR process of the form[2]

$$z_n = \sum_{i=1}^{N} \alpha_n z_{n-i} + w_n. \tag{1.11}$$

Comparing the state space model (1.9) to a classic tapped delay line filter model of an AR process, it can be seen that the state $\mathbf{x}_n$ and the process equation (1.9a) model the filter delay line and the temporal evolution of its content while the measurement equation (1.9b) models the tap weighting and summation of the input and tap values to form the output.

Now, if we relate the described AR process model to the DPCM encoder described in Section 1.2 and use it to model DPCM encoding of an AR source, we can let the state and the process equation (1.9a) represent the source AR model, cf. fig. 1.4. Then the finite impulse response (FIR), i.e., moving average (MA), filtering operation of the prediction error filter can be modeled by the measurement equation (1.9b) with the measurement vector $\mathbf{h}$ containing the tap weights of the prediction error filter. In this case the measurement vector can therefor not be used to model the source AR process itself as seen from (1.10). Using a different definition, it is possible to contain the entire modeling of the AR process in the process equation and leave room for modeling of the MA filtering operation in the measurement vector $\mathbf{h}$

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{F}\mathbf{x}_n + \mathbf{G}w_{n+1} \\ z_n &= \mathbf{h}^{\mathrm{T}}\mathbf{x}_n, \end{aligned} \tag{1.12}$$

[2]Notation changed compared to (1.6).

where

$$\mathbf{F} = \begin{bmatrix} -\alpha_1 & & \dots & & -\alpha_N \\ 1 & 0 & \dots & 0 & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix}_{(N\times N)} \qquad \mathbf{x}_n = \begin{bmatrix} z_n \\ \vdots \\ z_{n-(N-1)} \end{bmatrix}_{(N\times 1)}$$

$$\mathbf{G} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{(N\times 1)} \qquad \mathbf{h} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{(N\times 1)}, \tag{1.13}$$

$w_n \sim \mathcal{N}(0, \sigma_w^2)$.

Note that the model defined by (1.12) and (1.13) has a different time indexing of w than (1.9) and (1.10). This is necessary in order to accommodate the formulation (1.11) of the AR process. On the other hand, maintaining the time indexing from (1.9) and (1.10) would result in the following AR process definition:

$$z_n = \sum_{i=1}^{N} \alpha_n z_{n-i} + w_{n-1}, \tag{1.14}$$

resulting in a zero in the Z-domain transfer function of the process (as opposed to the conventional 1 in the numerator–see (1.15).) Both of the described approaches are however possible and are merely technicalities employed to enable the state space model (1.12) and (1.13). Neither approach alters the ability to estimate the state $\mathbf{x}_n$ from (1.12) and (1.13). In Publications A to C the state space model (1.12) and (1.13) is employed in a model of the complete DPCM encoder as depicted in fig. 1.4.

1.3.2 Stability and The State Transition Matrix

An AR process defined by a difference equation as (1.11) has the following Z-domain transfer function

$$H(z) = \frac{Y(z)}{W(z)} = \frac{1}{1 - \sum_{i=1}^{N} \alpha_n z^{-i}}. \tag{1.15}$$

The poles of $H(z)$, i.e., the roots of the denominator polynomial, are required to lie within the unit circle in the complex plane in order for the AR process to be stable. The state transition matrix $\mathbf{F}$ is the companion matrix of the denominator polynomial of the AR process transfer function (1.15),

or equivalently, the denominator polynomial is the characteristic polynomial of the state transition matrix. The roots of the characteristic polynomial equal the eigenvalues of $\mathbf{F}$ [19, p. 373]. As a consequence, stable AR processes–or stable filters in general–have state transition eigenvalues λ_i, where $|\lambda_i| < 1, \ \forall i$.

1.4 Minimum Mean Squared Error Estimation

Estimation can generally be formulated as the problem of estimating a stochastic variable of interest $\mathbf{x}$ by observing a related stochastic variable $\mathbf{y}$, resulting in the estimate $\hat{\mathbf{x}}$. The variables $\mathbf{x}$ and $\mathbf{y}$ must have a joint probability density function. If not, an observed outcome of $\mathbf{y}$ does not convey any information about $\mathbf{x}$. Although minimum mean squared error (MMSE) estimation applies to complex as well as real variables, we shall constrain the current exposition of the subject to real variables for simplicity.

Considering an estimate $\hat{\mathbf{x}}$ of a signal $\mathbf{x}$, it is desirable to be able to evaluate the quality of the estimate. This is evaluated as a measure of size of the error $\tilde{\mathbf{x}}$ between the true signal and its estimate

$$\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}. \tag{1.16}$$

In principle, one could consider any measure $d(\cdot)$ on the size of $\tilde{\mathbf{x}}$, but it turns out to be particularly useful to apply the mean squared error (MSE)

$$d(\tilde{\mathbf{x}}) \triangleq \mathrm{E}\left\{(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^{\mathrm{T}}\right\} = \mathbf{P}, \tag{1.17}$$

where the MSE is also the covariance matrix $\mathbf{P}$ of the estimation error $\tilde{\mathbf{x}}$. We use the above MSE (matrix) definition based on the outer product $\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{\mathrm{T}}$ rather than the inner product $\tilde{\mathbf{x}}^{\mathrm{T}}\tilde{\mathbf{x}}$, the latter of which is seen in some literature, because the latter conveys no information on the error of the individual elements of $\hat{\mathbf{x}}$ in the general case of $\hat{\mathbf{x}}$ being a vector. The matrix form also relates more directly to the Kalman filter which will be introduced in Section 1.4.3.

In order to find the optimal estimator of some variable $\mathbf{x}$, it must be defined in which sense it is optimal. When considering the MSE measure, the optimal estimator is the one that minimizes the MSE, hence the term "minimum mean squared error".

$$\hat{\mathbf{x}}^* = \arg\min_{\hat{\mathbf{x}}} \mathrm{E}\left\{(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^{\mathrm{T}}\right\}, \tag{1.18}$$

where $\hat{\mathbf{x}}^*$ denotes the MMSE estimate of $\mathbf{x}$, with $\cdot^*$ signifying the optimum. It can be shown that in general (see [20, App. 3.A])

$$\hat{\mathbf{x}}^* = \mathrm{E}\left\{\mathbf{x}|\mathbf{y}\right\}. \tag{1.19}$$

In practice it is usually most convenient and in some cases only possible to work with linear estimators, i.e., estimators of the form

$$\hat{\mathbf{x}} = \mathbf{v} + \mathbf{K}\mathbf{y}. \tag{1.20}$$

Strictly speaking, (1.20) is an *affine* estimator (because of $\mathbf{v}$), but the term *linear* estimator is more common in estimation literature. In particular, the latter is true when the variables $\mathbf{x}$ and $\mathbf{y}$ are assumed to be zero-mean stochastic variables, in which case $\mathbf{v} = 0$. In the following, we restrict the description to the case of $\mathbf{v} = 0$:

$$\hat{\mathbf{x}} = \mathbf{K}\mathbf{y}. \tag{1.21}$$

It is implicitly assumed that $\mathbf{K}$ is of appropriate dimensions with respect to $\hat{\mathbf{x}}$ and $\mathbf{y}$.

The MMSE estimator $\mathbf{K}^*\mathbf{y}$ of the form (1.21) can be found according to (1.18) by differentiating (1.17) and equating to zero, with the result that $\mathbf{K}^*$ is any solution to

$$\mathbf{K}^* \, \mathrm{E}\left\{\mathbf{y}\mathbf{y}^{\mathrm{T}}\right\} = \mathrm{E}\left\{\mathbf{x}\mathbf{y}^{\mathrm{T}}\right\}. \tag{1.22}$$

With the assumption that $\mathrm{E}\left\{\mathbf{y}\mathbf{y}^{\mathrm{T}}\right\} > 0$, the solution can be formulated as

$$\mathbf{K}^* = \mathrm{E}\left\{\mathbf{x}\mathbf{y}^{\mathrm{T}}\right\} \mathrm{E}\left\{\mathbf{y}\mathbf{y}^{\mathrm{T}}\right\}^{-1}. \tag{1.23}$$

It can be shown that

$$\mathbf{P}(\mathbf{K}) \triangleq \mathrm{E}\left\{(\mathbf{x} - \mathbf{K}\mathbf{y})(\mathbf{x} - \mathbf{K}\mathbf{y})^{\mathrm{T}}\right\} \geq \mathbf{P}(\mathbf{K}^*), \tag{1.24}$$

for any $\mathbf{K}$, which proves that $\mathbf{K}^*$ is the MMSE estimator, see [20]. In (1.24), the inequality $\mathbf{P}(\mathbf{K}) \geq \mathbf{P}(\mathbf{K}^*) \Leftrightarrow \mathbf{P}(\mathbf{K}) - \mathbf{P}(\mathbf{K}^*) \geq 0$ signifies that $\mathbf{P}(\mathbf{K}) - \mathbf{P}(\mathbf{K}^*)$ is positive-semidefinite.

The relation between $\mathbf{x}$, $\mathbf{y}$, and $\hat{\mathbf{x}}$ can be interpreted geometrically to give a more intuitive understanding of the MMSE principle. One may define the following inner product of stochastic vectors $\mathbf{x}$ and $\mathbf{y}$:[3]

$$\langle \mathbf{x}, \mathbf{y} \rangle \triangleq \mathrm{E}\left\{\mathbf{x}\mathbf{y}^{\mathrm{T}}\right\} \tag{1.25}$$

From (1.22) follows

$$\mathrm{E}\left\{(\mathbf{x} - \mathbf{K}^*\mathbf{y})\,\mathbf{y}^{\mathrm{T}}\right\} = 0, \tag{1.26}$$

and from (1.25)

$$\langle \tilde{\mathbf{x}}^*, \mathbf{y} \rangle = 0. \tag{1.27}$$

[3]Note that although an inner product is usually understood to be a scalar, none of the conditions (linearity, symmetry, positive semi-definiteness) in the definition of real inner product spaces prevent it from being a matrix as seen here [19, p. 361].

The geometric interpretation of (1.27) is that the minimum mean squared error of the estimate $\hat{\mathbf{x}}^*$ is orthogonal to the observation $\mathbf{y}$. This is typically referred to as the *orthogonality principle*. The orthogonality principle is depicted geometrically in fig. 1.5. $\mathbf{x}$, $\mathbf{y}$, and $\hat{\mathbf{x}}$ all lie in the space V of all functions of $\mathbf{x}$ and $\mathbf{y}$ while $\mathbf{y}$ and all estimates $\hat{\mathbf{x}}$ (also) lie in the subspace W (of V) of all linear functions of $\mathbf{y}$. From a geometric perspective it is immediately clear that the error vector $\tilde{\mathbf{x}}^*$ orthogonal to $\hat{\mathbf{x}}^*$ is the smallest such vector

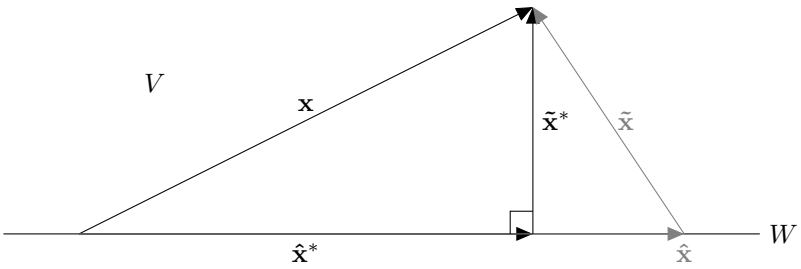


Figure 1.5: The orthogonality principle in MMSE estimation. The estimate $\hat{\mathbf{x}}$ and associated error $\tilde{\mathbf{x}}$ represent any other non-MMSE linear estimator. The plane represents the space V while the horizontal line on which the estimates are drawn represents the subspace W.

1.4.1 Wiener Filtering

The theory of Wiener filtering encompasses MMSE sequence estimation of stochastic processes [21]. While the Wiener filter is generally a technique for continuous-time stochastic processes, we shall only consider the discrete-time version here. Note also that we shall only consider the case of scalar stochastic processes here. The theory extends to the vector case as well, but since we primarily treat the Wiener filter here as background information for the Kalman filter introduced in Section 1.4.3, we shall not get into the details of the vector case here; cf. [20, Sec. 7.8] for more information.

Consider the zero-mean jointly wide-sense stationary random processes x_n and y_n with known first- and second-order moments, i.e., expectation and variance. The (discrete-time) Wiener filter is the MMSE-optimal linear estimator of x_n given all observations $\{y_n\}_{n=-\infty}^{\infty}$:

$$\hat{x}_n = \sum_{i=-\infty}^{\infty} k_{n-i} y_n \tag{1.28}$$

The equivalent of (1.22) for finding the optimal estimator in terms of $k_i,\ i \in (-\infty, \infty)$ becomes ([20, Sec. 7.3.1])

$$\sum_{m=-\infty}^{\infty} k_{i-m}\, \mathrm{R}_{yy}(m) = \mathrm{R}_{xy}(i), \tag{1.29}$$

where $\mathrm{R}_{xy}(m) = \mathrm{E}\{x(n)y(n-m)\} = \mathrm{E}\{x(n+l)y(n+l-m),\ \forall l\}$ where the second equality is due to the assumption of wide-sense stationarity. In (1.29), the optimal estimator given by $\hat{x}^*(i),\ \forall i$ cannot be found by inversion of R_{yy} as done in (1.23). The solution can however be found in the discrete Fourier transform domain as:

$$K\left(e^{j\omega}\right) = \frac{S_{xy}\left(e^{j\omega}\right)}{S_{yy}\left(e^{j\omega}\right)} \tag{1.30}$$

Seen from the perspective of practical implementability, the Wiener filter as given by (1.28) is not practical due to the dependence of the estimator on observations into the infinite future, i.e., the estimator is not causal. If we restrict the Wiener filter to rely on observations up to and including the current time, we can formulate it as (1.31).

$$\hat{x}_{n|n} = \sum_{i=-\infty}^{n} k_{n-i} y_n \tag{1.31}$$

A useful ingredient in the derivation of the optimal estimator of the type in (1.31) is the concept of innovations which we shall describe in Section 1.4.2.

1.4.2 Innovations

Based on (1.22) it can be seen that $\mathbf{K}^*$ would be easy to calculate, if $\mathrm{R}_{\mathbf{yy}}$ were a diagonal matrix, i.e., if the observations were uncorrelated. In general, this is a very restrictive assumption, but there is a way to achieve this desirable property.

The innovations are defined as ([20, Ch. 4])

$$\tilde{\mathbf{y}}_n \triangleq \mathbf{y}_n - \hat{\mathbf{y}}_{n|n-1}, \tag{1.32}$$

where

$$\hat{\mathbf{y}}_{n|n-1} \triangleq \sum_{i=0}^{n-1} \langle \mathbf{y}_n, \tilde{\mathbf{y}}_i \rangle \langle \tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_i \rangle^{-1} \tilde{\mathbf{y}}_i, \tag{1.33}$$

the projection of $\mathbf{y}_n$ onto the vector space spanned by $\{\tilde{\mathbf{y}}_0, \tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_{n-1}\}$. The innovations can be seen as the part of an observation $\mathbf{y}_n$ that cannot be predicted from the past observations and thus represents the *new information* in $\mathbf{y}_n$.

By the definitions (1.32) and (1.33) (subtraction of the orthogonal projection onto $\{\tilde{\mathbf{y}}_0, \tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_{n-1}\}$), each innovation $\tilde{\mathbf{y}}_n$ is orthogonal to all previous innovations $\tilde{\mathbf{y}}_0 \ldots \tilde{\mathbf{y}}_{n-1}$, resulting in the innovations sequence being uncorrelated (white), whereby $\mathrm{R}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}$ is diagonal. The observations and the innovations are equivalent in the sense that

$$\mathrm{E}\left\{\mathbf{x}|\mathbf{y}_0, \ldots, \mathbf{y}_n\right\} = \mathrm{E}\left\{\mathbf{x}|\tilde{\mathbf{y}}_0, \ldots, \tilde{\mathbf{y}}_n\right\}.$$

The advantage of the innovations is that the signal of interest can be estimated from the uncorrelated innovations sequence in stead of the generally correlated observations sequence. This can be exploited in the causal Wiener filter (1.31), based on the innovations in stead ([20, Sec. 7.7]):

$$\hat{x}_{n|n} = \sum_{i=-\infty}^{n} g_{n-i}\tilde{y}_n. \tag{1.34}$$

By the orthogonality principle (1.27),

$$\langle x_n - \hat{x}_{n|n}, \tilde{y}_i\rangle = 0, \text{ for } i \in (-\infty, n]. \tag{1.35}$$

Since the process x and thereby $\tilde{y}$ are stationary,

$$\langle x_n, \tilde{y}_i\rangle \triangleq \mathrm{R}_{x\tilde{y}}(n-i). \tag{1.36}$$

From (1.35) one obtains

$$\mathrm{R}_{x\tilde{y}}(n-i) = \sum_{j=-\infty}^{n} g_{n-j}\langle\tilde{y}_j, \tilde{y}_i\rangle = g_{n-j}r_{\tilde{y}\tilde{y}}, \text{ for } i \leq n, \tag{1.37}$$

since $\langle\tilde{y}_j, \tilde{y}_i\rangle = 0$, for $i \neq j$ and $r_{\tilde{y}\tilde{y}} \triangleq \langle\tilde{y}_i, \tilde{y}_i\rangle$. g_{n-j} can now be straightforwardly isolated from (1.37), demonstrating the usefulness of the innovations. Cf. [20, Sec. 7.7] for details on calculation of $\mathrm{R}_{x\tilde{y}}(n-i)$.

Wiener filtering can be taken further and can for example be defined for recursive estimation of state sequences from observations thereof governed by a stationary state-space model of the form (1.8), i.e., with constant matrices $\mathbf{F}$, $\mathbf{G}_1$, $\mathbf{H}$, and $\mathbf{G}_2$ and jointly stationary zero-mean random processes $\mathbf{w}_n$ and $\mathbf{v}_n$. This line of thought leads to the recursive linear minimum mean-squared error (LMMSE) estimation based on a certain class of (generally non-stationary) state-space models. Such an estimation framework, known as Kalman filtering, is described in the following section.

1.4.3 Kalman Filtering

A major step in estimation theory was taken in 1960 by Kalman with his paper on a new approach to linear estimation [22]. The paper presented a

filtering, or estimation, framework in which the state $\mathbf{x}$ of a system represented by a very general state space model such as (1.8) can be estimated optimally in the MMSE sense. Here we simplify the state space model slightly compared to (1.8) to a form commonly encountered in Kalman filtering literature, see (1.38)[4].

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{F}_n \mathbf{x}_n + \mathbf{G}_n \mathbf{w}_n \\ \mathbf{y}_n &= \mathbf{h}_n^{\mathrm{T}} \mathbf{x}_n + \mathbf{v}_n, \end{aligned} \tag{1.38}$$

When the Kalman filter is applied for estimation of the state $\mathbf{x}_n$, it is required, in order for the Kalman filter to be MMSE-optimal, that $\mathbf{w}_n \sim \mathcal{N}(0, \mathbf{Q}_n)$ and $\mathbf{v}_n \sim \mathcal{N}(0, \mathbf{R}_n)$, and $\mathbf{x}_0 \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{P}_0)$ with $\mathbf{w}_n$, $\mathbf{v}_n$, and $\mathbf{x}_0$ independent. In many formulations of the Kalman filter, it is typically required that $\mathrm{E}\left\{\mathbf{w}_n {\mathbf{v}_n}^{\mathrm{T}}\right\} = \mathbf{0}$. The Kalman filter can be formulated for correlated process and measurement noise as well, i.e., $\mathbf{S}_n \triangleq \mathrm{E}\left\{\mathbf{w}_n {\mathbf{v}_n}^{\mathrm{T}}\right\} \neq 0$, see [23].

Note that the system in (1.38) is generally time-variant ($\mathbf{F}_n$, $\mathbf{h}_n$, $\mathbf{Q}_n$, and $\mathbf{R}_n$ may be time-varying) and is therefor not necessarily stationary.

The Kalman filter has gained immense popularity since its introduction, perhaps particularly in the area of control theory. It is treated extensively in for example [20, 23–27]. The estimation principles and accompanying equations are summarized in the following.

The Kalman estimator is defined by two sets of equations (1.39), commonly referred to as measurement update and time update, respectively. These steps can also be interpreted as estimation and prediction.

Measurement update:

$$\hat{\mathbf{x}}_n = \hat{\mathbf{x}}_n^- + \mathbf{K}_n \left(\mathbf{y}_n - \mathbf{h}_n^{\mathrm{T}} \hat{\mathbf{x}}_n^-\right) \tag{1.39a}$$

$$\mathbf{P}_n = \mathbf{P}_n^- - \mathbf{K}_n \mathbf{h}_n^{\mathrm{T}} {\mathbf{P}_n^-}^{\mathrm{T}} \tag{1.39b}$$

where

$$\mathbf{K}_n \triangleq \mathbf{P}_n^- \mathbf{h}_n \left(\mathbf{h}_n^{\mathrm{T}} \mathbf{P}_n^- \mathbf{h}_n + \mathbf{R}_n\right)^{-1}$$

Time update:

$$\hat{\mathbf{x}}_{n+1}^- = \mathbf{F}_n \hat{\mathbf{x}}_n + \mathbf{G}_n \mathbf{S}_n {\mathbf{R}_n}^{-1} \left(\mathbf{y}_n - \mathbf{h}_n^{\mathrm{T}} \hat{\mathbf{x}}_n\right) \tag{1.39c}$$

$$\begin{aligned} \mathbf{P}_{n+1}^- = {} & \left(\mathbf{F}_n - \mathbf{G}_n \mathbf{S}_n {\mathbf{R}_n}^{-1} \mathbf{h}_n^{\mathrm{T}}\right) \mathbf{P}_n \left(\mathbf{F}_n - \mathbf{G}_n \mathbf{S}_n {\mathbf{R}_n}^{-1} \mathbf{h}_n^{\mathrm{T}}\right)^{\mathrm{T}} \\ & + \mathbf{G}_n \left(\mathbf{Q}_n - \mathbf{S}_n {\mathbf{R}_n}^{-1} {\mathbf{S}_n}^{\mathrm{T}}\right) {\mathbf{G}_n}^{\mathrm{T}} \end{aligned} \tag{1.39d}$$

[4] Not all of the elements given in (1.38) are part of [22]. Formulation for a more general state space model available in for example [23].

Equation (1.39) requires some explanation of notation:

$\hat{\mathbf{x}}_n$ State *estimate* at time n given observations up to and including time n, i.e., $\mathrm{E}\{\mathbf{x}_n|\mathbf{y}_0,\ldots,\mathbf{y}_n\}$.

$\hat{\mathbf{x}}_n^-$ State *prediction* at time n given observations up to and including time $n-1$, i.e., $\mathrm{E}\{\mathbf{x}_n|\mathbf{y}_0,\ldots,\mathbf{y}_{n-1}\}$. This corresponds to the projection of $\mathbf{x}_n$ onto the space spanned by $\{\mathbf{y}_0,\ldots,\mathbf{y}_{n-1}\}$.

$\mathbf{K}_n$ The so-called Kalman gain. This updates the prediction of the state, $\hat{\mathbf{x}}_n^-$, with the new information contained in the innovation $\tilde{\mathbf{y}}_n = \mathbf{y}_n - \mathbf{h}_n^{\mathrm{T}}\hat{\mathbf{x}}_n^-$.

$\mathbf{P}_n$ State estimation error covariance $\mathbf{P}_n = \mathrm{E}\left\{(\mathbf{x}_n - \hat{\mathbf{x}}_n)(\mathbf{x}_n - \hat{\mathbf{x}}_n)^{\mathrm{T}}\right\}$

$\mathbf{P}_n^-$ State prediction error covariance $\mathbf{P}_n^- = \mathrm{E}\left\{(\mathbf{x}_n - \hat{\mathbf{x}}_n^-)(\mathbf{x}_n - \hat{\mathbf{x}}_n^-)^{\mathrm{T}}\right\}$.

As can be seen from (1.39) the Kalman recursions consist of two equations, (1.39a) and (1.39c), calculating the actual estimates and predictions of the state $\mathbf{x}_n$, as well as two equations, (1.39b) and (1.39d), updating the covariances involved in the calculation of (1.39a) and (1.39c).

One interesting fact about the Kalman filter is that for a stationary signal model (1.38) satisfying the assumptions for the Kalman filter, the Kalman filter is equivalent to the Wiener filter (1.31) as $n \to \infty$ [23, 28].

Kalman filtering has also found its way into source coding. Early examples of the use of Kalman filtering in source coding are found in [29–31]. Here, the Kalman filter is not directly incorporated into the signal path of the source signal coding; it is employed in estimating and updating the predictor coefficients of a linear predictor FIR filter. In later examples, the Kalman filter is incorporated directly into the coding signal path. [32] employs the Kalman filter as the actual predictor in the encoder, effectively seeing the quantized prediction residual as Kalman innovations. See Section 1.6.5 for further discussion of the Kalman filter as predictor. The decoder is a replica of the encoder's Kalman filter, receiving its innovations from the encoder. A related use of the Kalman filter is seen in [33] which is a transform coding framework. The Kalman innovations represent the quantized prediction residuals, which are here quantized vectors. The Kalman measurement matrix represents the linear transform. Still a predictive coding scheme, [34] brings Kalman filtering to the class of analysis-by-synthesis source coding. Again the Kalman filter is employed as predictor, but the prediction error is not quantized directly, rather matched to a codebook of

excitation signals in order for the signal synthesized from the excitation to match the original source signal best, i.e. the code-excited linear prediction (CELP) principle.

A very recent example of Kalman filtering in source coding is [35]. Here, the Kalman filter is used as predictor in a Gaussian mixture model (GMM)-based predictive coding framework.

1.5 Quantization

Quantization is often associated with analog-to-digital (A/D) conversion, which consists of time discretization (sample-and-hold) and value discretization (quantization). Quantization is the conversion of a continuous value x to a discrete value y. The discrete value is selected from a–for practical purposes, finite–set $Y = \{y_1 \dots y_N\}$. The quantizer performing this quantization selects the quantized value $y \in Y$ from the discrete set Y as the closest approximation, in some sense, to the original analog value x. The measure by which the closest approximation is selected is in general often the MSE. In specific applications such as audio or image coding, one often uses other measures of error tailored to match the percepual characteristics of human hearing resp. vision [36–38].

Quantization can be thought of in a broader sense than just discretization of continuous values. For example, quantization is often performed in a digital signal processing system where a higher-precision discrete value is converted (quantized) to a discrete value of smaller precision. This is done in source coding systems in order to reduce the required bit-rate for transmission of the source signal. The history of quantization is treated thoroughly in [39].

Quantization can be generalized from scalars to vectors, where the input values and quantized values are vectors. Since the work presented in this thesis is based on classic DPCM-type source coding, which uses scalar quantization, we shall not get into the details of vector quantization here. For further details, we refer to [40].

1.5.1 Scalar Quantizers

A quantizer can be completely defined in terms of its partition cells (input levels) $\{R_i | i = 1, 2, \dots, N\}$ and codebook (output levels) $\{y_i | 1, 2, \dots, N\}$. If we consider scalar quantization, the partition cells are intervals of the type $(x_{i-1}, x_i]$ with decision points x_i. A quantizer is termed regular if its cells R_i are contiguous intervals of this form, and $y_i \in (x_{i-1}, x_i)$ [40]. For unbounded inputs x, the quantizer's partition cells consist of overload cells $R_1 = (-\infty, x_1]$ and $R_N = (x_{N-1}, \infty)$; and granular cells $R_i = (x_{i-1}, x_i]$, $i =$

$2, \ldots, N-1$. The quantizer operates by selecting output values as follows:

$$y = y_i \text{ if } x_{i-1} < x \leq x_i. \tag{1.40}$$

Quantization of a value x results in a quantization index i according to (1.40). The index i is converted to a sequence of bits and perhaps entropy coded before the value is transmitted to a receiver which in turn reconstructs the quantized value $y = y_i$ from the codebook based on i.

The design of quantizers, i.e., the selection of decision points x_i and output values y_i defining the quantizer, can in practice be based on statistical properties of the input x or "trained" from representative examples of real input data.

A classic type of scalar quantizer is the Lloyd-Max quantizer, discovered independently by Lloyd and Max, as presented in [41, 42]. This type of scalar quantizer is fitted to the probability density function (p.d.f.) of the quantizer input, either by assuming a known input p.d.f. or by basing it on training data. The output levels are the centroids of the partition cells. The so-called *Lloyd I* and *Lloyd II* algorithms are available for designing the quantizer [40–42]. An example of a Lloyd-Max quantizer is given in fig. 1.6 for a Gaussian input $x \sim \mathcal{N}(0, 1)$.

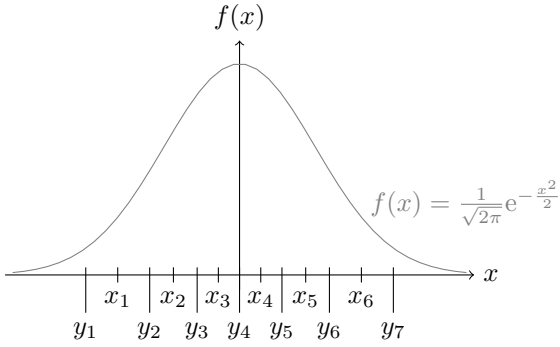


Figure 1.6: Lloyd-Max partition cells and output levels for a Gaussian input p.d.f.. Decision points x_i and output levels y_i are plotted along the x-axis.

A different type of scalar quantizer is the uniform quantizer. As the name suggests, its decision points are uniformly distributed

$$R_i = (x_{i-1}, x_{i-1} + \Delta]. \tag{1.41}$$

Generally, a uniform quantizer has uniformly distributed decision points (1.41), whereas the output levels are not necessarily uniformly distributed but could be selected as, typically, the centroids or midpoints of the partition cells. When referring to uniform quantizers in this thesis, we shall consider

the term to cover uniform quantizers with midpoint output levels. Consequently, the output levels are uniformly distributed as well

$$y_i = \frac{x_{i-1} + x_i}{2} = x_{i-1} + \frac{\Delta}{2} = y_{i-1} + \Delta. \tag{1.42}$$

Uniform scalar quantizers can be designed, i.e., calculation of Δ, according to [43]. An example of a bounded uniform quantizer is given in fig. 1.7 for a Gaussian input $x \sim \mathcal{N}(0, 1)$.

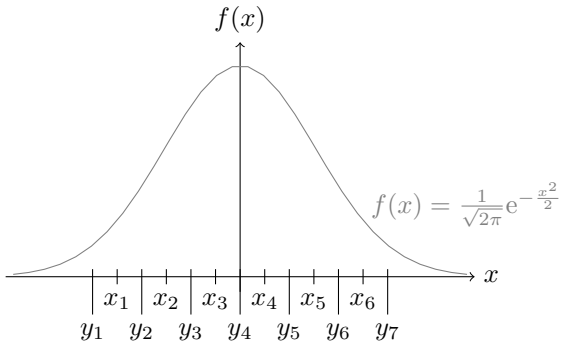


Figure 1.7: Uniform partition cells and output levels for a Gaussian input p.d.f.. Decision points x_i and output levels y_i are plotted along the x-axis.

1.5.2 Lloyd-Max vs. Uniform Quantizers

In terms of computational complexity, one has to consider that Lloyd-Max quantizers have to compare the input value to potentially all decision levels to determine the index, while uniform quantizers can be implemented much simpler using scaling and rounding. On the other hand, uniform quantizers do not quantize the input as efficiently as Lloyd-Max quantizers, i.e., they introduce a larger quantization error on average at a given resolution / number of bits per quantized sample. This, however, only holds under the assumption of fixed-rate quantization and non-uniform input p.d.f..

A Lloyd-Max quantizer performs fixed-rate quantization, i.e., it assigns a fixed number of bits per sample. This is a result of the design of the quantizer which calculates the decision points with equal input value probability in each partition cell. Therefore all possible quantizer indices $i \in [1, N]$ are equally probable and so, it makes sense to spend an equal number of bits on all indices.

Except for uniform input p.d.f.'s, the quantization indices produced by a uniform quantizer are generally not equally likely. Therefore, the rate of this type of quantizer can be reduced by entropy coding without affecting the

distortion. The basic principle of entropy coding is to code the quantization indices using a variable number of bits, such that the most probable indices are assigned to the shortest index code words and the least probable indices are assigned to the longest code words. This minimizes the *average* required bit-rate for transmission of the quantization indices. However, it implies that the resulting transmission bit rate is variable as opposed to fixed in the example of the Lloyd-Max quantizer. In this work, we consider only fixed-rate quantization and do not employ entropy coding.

1.5.3 Quantization Noise and Modeling

Clearly, cf. (1.40), a quantizer's output is a nonlinear (and discontinuous) function of the input. This is inconvenient for most practical applications since it makes the quantizer difficult to model.

Analysis of the behavior of quantization noise started from [44]. It is typically assumed that quantization noise can be considered uniform white for sufficiently high rates. A lot of information-theoretical work on quantization is actually based on the assumption of 'high rate', in theory tending to infinity. It turns out that reasonably low rates can be considered 'high rate' in practice [45]. However, at low rates the assumption of uniform white quantization noise does not generally hold. In fact quantization noise can be shown to approach a Gaussian distribution for optimal lattice vector quantizers when the vector dimension approaches infinity [46]. The latter is however not relevant to scalar quantizers. Quantization noise spectra for various cases were analyzed extensively for uniform quantizers in [47].

The error introduced by quantizing the input x can be defined as additive noise $q(n)$

$$y(n) = x(n) + q(n). \tag{1.43}$$

In statistical signal processing applications such as (linear) estimation we are interested in a linear model of the quantizer. It is not necessary to model the quantization operation itself, a statistical model of the quantizer is sufficient. It is convenient to work with a statistical model in which the noise is uncorrelated with the input, i.e.

$$\mathrm{E}\left\{y^2(n)\right\} = \mathrm{E}\left\{x^2(n)\right\} + \mathrm{E}\left\{q^2(n)\right\} \Leftrightarrow \mathrm{E}\left\{x(n)q(n)\right\} = 0. \tag{1.44}$$

However, for quantizers with centroid output levels (for example Lloyd-Max quantizers), the following holds

$$\mathrm{E}\left\{y(n)q(n)\right\} = 0 \tag{1.45}$$

which implies, cf. [40, p. 181],

$$\mathrm{E}\left\{(x(n) - y(n))^2\right\} = \mathrm{E}\left\{x^2(n)\right\} - \mathrm{E}\left\{y^2(n)\right\}. \tag{1.46}$$

Actually, it has been shown in [48] for a uniform quantizer that $\mathrm{E}\{y(n)q(n)\} = 0$ for centroid output levels–as in (1.45)–while $\mathrm{E}\{x(n)q(n)\} = 0$ for mid-point output levels, under high-rate assumptions.

As a consequence of (1.46), the model (1.43) is not valid under the assumption (1.44). To accommodate (1.46), one can consider the following model, typically referred to as the gain-plus-additive noise quantizer model

$$y(n) = \rho x(n) + r(n). \tag{1.47}$$

Equation (1.47) allows modeling the quantization noise $q(n)$ as correlated with the input $x(n)$:

$$q(n) = y(n) - x(n) = (\rho - 1)x(n) + r(n). \tag{1.48}$$

The model (1.48) accomodates (1.46) for $\rho < 1$, where ρ is calculated as [49]

$$\rho = 1 - \frac{\mathrm{E}\left\{q^2(n)\right\}}{\mathrm{E}\left\{x^2(n)\right\}}. \tag{1.49}$$

Furthermore, it follows from the derivations in [49] that

$$\mathrm{E}\left\{r^2(n)\right\} = \rho(1-\rho)\,\mathrm{E}\left\{x^2(n)\right\}. \tag{1.50}$$

Equation (1.50) captures the fact that the quantization noise variance grows as the input to the quantizer grows by modeling the variance of the additive quantization noise as proportional to the input variance. In Section 1.2.3, it was mentioned that the effect of error propagation in DPCM can be reduced by choosing the prediction coefficient a_1 smaller than the AR model coefficient α_1. In light of the dependency of quantization noise on the input to the quantizer, as modeled in (1.50), we can see from Section 1.2.2–especially (1.7)–that the optimal choice of $a_1 < \alpha_1$ to reduce the effect of error propagation is a trade-off with respect to quantization noise since the effect of the latter will be increased by such a choice of a_1. This goes for a general choice $a_i \neq \alpha_i,\ \forall i$ as well.

The linear quantization noise model (1.47) was chosen for the framework presented in Publications A to C. It has the advantage of being simple and above all linear, thus making it possible to integrate the quantization noise model into the standard Kalman filter equations as described in Publications A to C. While it is simple, it could also be argued that is has a drawback of not characterizing the quantizer accurately enough. Especially in the design of the quantizer as presented in Publication A, the additive noise $r(n)$ is considered white and Gaussian, both of which are simplifying assumptions.

1.5.4 Alternative Approaches to Quantization Noise Modeling

Alternative approaches related to Kalman filtering exist which could possibly take into account non-white and/or non-Gaussian noise.

A well-known approach to handling non-linear systems in Kalman filtering is the "extended Kalman filter" [23, Sec. 8.2]. This approach is based on the standard Kalman filter, but the filter is in this case applied to a state-space model that is a linear approximation of the underlying non-linear system. The linear approximation is obtained by Taylor expansion (cf. e.g. [50, Sec. 9.19]) of the non-linear functions of the system. Due to the Taylor expansion, this approximation is only expected to work well for sufficiently smooth non-linear functions. As a quantizer is generally not only non-linear but also discontinuous, we consider the extended Kalman filter (EKF) merely a complicated way of obtaining an alternative linear approximation–to that of (1.47)–that we do not expect to perform significantly better.

A more recent alternative to the EKF, reported to work much better in practice, is the "unscented Kalman filter" [51, 52]. This modification of Kalman filtering is based on the so-called "unscented transform". Unfortunately, filter recursions of the unscented Kalman filter (UKF) take a form that does not allow the approach for optimization considered in Publications A to C.

A result in Kalman filtering for non-Gaussian measurement noise can be found in [53]. Here, the Kalman estimate is modified by means of a score function based on the measurement p.d.f.. This approach is however also associated with several difficulties. It introduces the difficulty of finding or estimating the measurement p.d.f. which would again complicate the quantizer design due to the inter-relation of quantizer and encoder filter design pointed out in Section 1.2. The Kalman statistics recursions are subject to a modification involving the mentioned score function which we believe would introduce some non-linear dependencies on the encoder filters preventing the optimization considered in Publications A to C. The score function concept is simplified somewhat in [54] through approximation of the score function. The score function concept is also developed for the case of generalized Gaussian noise in [55]; the generalized Gaussian p.d.f. includes, e.g., the Laplacian and the uniform as well as the Gaussian p.d.f.s and is also suited for modeling more heavy-tailed distributions than the Gaussian p.d.f.. The score function-based approaches do however not facilitate the optimization in Publications A to C.

Finally, one could also consider the concept of robust estimation for handling the quantization noise issue. The idea here is that the exact probability distributions of stochastic components in the system are not known. The solution is to design estimators in such a way that the estimation per-

formance is degraded as little as possible when the assumptions made about the distributions are not correct or adequate. Results on robust estimation applied within the framework of Kalman filtering can be found in [56, 57]. The approach in [56] primarily changes the estimation step of the Kalman filter by deriving it according to a more statistically robust error minimization criterion than the mean-squared error criterion leading to the classic Kalman filter. It does suggest as one possible solution to keep the original Kalman expressions for the statistics updates and could as such still accommodate the optimization approach used in Publications A to C. Such an approach could be a topic of further research.

1.6 Overview of Conducted Research

In most data communication environments, loss of transmitted data is an unavoidable impairment. Wired network environments are known to introduce losses due to congestion in network nodes. On wireless links, interference and adverse signal-to-noise conditions can cause erroneously received data as well. Several types of error-robustness techniques can be employed to deal with transmission impairments. Overall, these techniques can be classified as error detection, error recovery, or error concealment.

Error detection techniques provide a mechanism to detect whether received data are subject to errors by transmitting parity data along with the payload data [58]. Error detection does not provide a mechanism to correct the detected errors, so upon reception of data, the receiver can only use the parity data to check the integrity of the data (i.e., detect errors) and subsequently decide how to treat the erroneous data.

One typical error recovery mechanism to ensure correct reception of transmitted data is the closed-loop approach of retransmission, for example seen in the form of automatic repeat request (ARQ) [59]. Network transport protocols such as Transmission Control Protocol (TCP) employ retransmission of lost data [60]. Retransmission schemes need to employ an error detecting code in order to determine which data has to be retransmitted due to errors. However, retransmission naturally imposes further delay on transmitted data and protocols intended for transmission of data with real-time requirements, e.g., Real-time Transport Protocol (RTP) / RTP Control Protocol (RTCP) [61], typically do not employ this mechanism and utilize User Datagram Protocol (UDP) for network transport [62]. Another possibility for error recovery is the open-loop approach of forward error correction (FEC) where error correcting data is transmitted along with the payload data to enable reconstruction of the latter in case of errors. FEC however also introduces some delay and increases the required bandwidth for transmitting data as it introduces transmission of error correcting data. Both retransmission and FEC are separate channel coding measures in the

sense discussed in Section 1.1. As also mentioned in Section 1.1, it can also make sense to combine source and channel coding to mitigate effects of data lost in transmission.

Yet another possibility for dealing with transmission losses is to attempt to conceal the effects of lost data (error concealment.) Examples of this approach are [63, 64]. For an overview and taxonomy of error robustness techniques, see [65, Sec. 3.3].

The motivation for the work in this project stems from the hypothesis that it is possible to optimize the coding of DPCM-coded source signals to take channel losses into account. As such, it can be seen as a form of joint source/channel coding.

1.6.1 Earlier Work on Channel Optimization of Predictive Coding

As mentioned in Section 1.2, several publications treat the optimization of DPCM. In [66] a classic DPCM system with a first order predictor is optimized for noisy channels, but not losses as such.

Sayood and Borkenhagen introduced the idea in [67] that a DPCM encoder leaves redundancy, i.e., residual temporal correlation, in the prediction error residual, either due to imperfect knowledge of the source or simplifying assumptions. This residual redundancy can be exploited in the decoder together with knowledge of the channel to optimize the decoding operation in a manner akin to decoding of convolutional channel codes. Some remarkable improvements compared to standard DPCM were demonstrated in [67]. The main contribution of this work is the decoder that exploits the residual redundancy in the source signal. It does however also employ optimization of the predictor by means of a method from [66].

Eriksson, Lindén, and Skoglund introduced a "safety-net" concept in [68, 69]. The main idea in this concept is that some of the source samples are predictively quantized–as in DPCM–and some are quantized directly. We shall not get into the details of this combination of techniques here, but the interesting part is that the directly quantized samples help improve performance under transmission across noisy channels. Direct quantization can actually be seen as predictive quantization in the extreme case of leaving redundancy in the prediction residual, i.e., there is no prediction at all and the prediction residual is thus the source signal itself–hence the direct quantization. This underlines the principle suggested in [67].

Chan considered DPCM coding in a system with packet losses which may result in collective loss of several transmitted indices [70]. This work investigates optimization of a system with a first-order predictor.

As the cited references in this section indicate, error resilience can be achieved in predictive coding by somehow leaving redundancy in the prediction residual. This generally translates into choosing predictor coefficients

that 'mis-match' the source, as exemplified in (1.7) when $\alpha_i \neq a_i$. In this way the optimization is actually related to bandwidth expansion, known from speech coding, where the linear predictive coding (LPC) coefficients are scaled to smooth the spectrum represented by the LPC coefficients:

$$a_i' = \lambda^i a_i, \tag{1.51}$$

which has the effect of 'broadening' the peaks of the spectrum. The principle is mentioned in for example [71]. In fact, the same principle can be found in video coding, known as leaky prediction. See for example [72–74]. When applied as leaky prediction in video coding, the objective is to reduce error propagation in the decoded video signal. This was discussed in Section 1.2.3.

The principle of leaving redundancy in the prediction residual to achieve error resilience is a main motivation for the research presented in this thesis.

1.6.2 Motivation and Main Hypothesis

Based on existing approaches for optimization of DPCM with respect to adverse channel conditions, it seems likely that further steps could be taken to design a linear predictive coding framework to take transmission losses into account. On this basis, the hypothesis motivating the research conducted in this project is:

> It is possible to optimize the filtering operations of a general linear predictive coding framework for transmission across lossy channels to improve decoded signal quality under transmission loss significantly compared to frameworks not optimized for loss. This can be done through optimization of the encoder so as to leave appropriate redundancy in the prediction residual.

In the current work, it was decided to take as starting point a DPCM coding framework where the decoding is based on Kalman estimation since this provides LMMSE estimation given the observed data, i.e., quantized prediction residuals reconstructed at the decoder. As mentioned in Section 1.4.3 several predictive coding approaches employ Kalman filtering. However, it was decided to use a different approach than the existing approaches (employing a Kalman filter as the predictor), in order to facilitate the modeling of losses of the transmitted data.

The chosen DPCM encoder structure is basically the noise shaping structure described in Section 1.2.1. It allows for state-space modeling of the encoder, as described in detail in Publications A to C, in such a way that the quantized prediction errors reconstructed from the quantization indices transmitted to the decoder are interpreted as noisy measurements. The (linear) state-space model of the encoder employs the gain-plus-additive noise quantizer model (1.47) described in Section 1.5.3.

Decoding at the receiver side is accomplished by means of a Kalman estimate of the state of the encoder. The state of the encoder contains a number of the most recent samples of the original source sequence, from which the decoded source signal is formed at the receiver.

1.6.3 Optimization in Kalman Estimation

So far, the described framework consists of a DPCM encoder and a Kalman estimator-based decoder. The encoder is modeled in such a way that the measurement vector in the state space model of the encoder represents the FIR predictor and noise feedback filters. Optimizing the encoder's filtering operations can thus be seen as optimizing the measurement vector in Kalman estimation terminology. Ramabadran and Sinha present an optimization approach in [75, 76] that aims to minimize the state estimation error covariance through optimization of the measurement vector. This idea has inspired the approach detailed in Publications A to C.

1.6.4 Estimation From Lossy Observations

Considering the possible loss of transmitted information, a suitable approach for modeling such transmission losses is required. One approach would be to model the channel loss process as a Markovian jump linear system (JLS). A JLS is a linear system, modeled for example as in (1.8), where some or all of the system parameters ($\mathbf{F}_n$, $\mathbf{H}_n$, $\mathbf{G}_{1,n}$, $\mathbf{G}_{2,n}$, $\mathbf{Q}_n$, $\mathbf{R}_n$) change over time within a set of different values for each parameter, e.g.,

$$\mathbf{F}_n = \mathbf{F}_{\theta(n)} \in \{\mathbf{F}_\theta | \theta = 1, \dots N\}.$$

The evolution of system parameter values is governed by a Markov chain with states $\theta \in [1, \dots N]$, hence the name *Markovian* JLS. Examples of this modeling approach are found in [77–79]. JLS modeling potentially has a lot of different applications. When used for transmission loss modeling, i.e., modeling of lost observations in the state-space sense, a straight-forward interpretation of loss is to define

$$\mathbf{H}_\theta = \mathbf{0} \qquad\qquad \mathbf{G}_{2,n} = \mathbf{0},$$

for the state(s) θ representing loss. Markov modeling enables quite general loss models with this type (JLS) of state-space model as exemplified in [79]. However, this also makes them rather challenging to optimize [78, 80, 81].

For a simple two-state loss model modeling correct reception vs. loss of transmitted information, it is realistic to assume the loss state (θ) is known, i.e. in typical transmission scenarios it is possible to determine whether transmitted information has been lost. In the case of more complex loss models however, it can be difficult or unrealistic to assume the loss state θ is

known. This could for example be a model representing loss statistics that vary over time so that, although one can determine whether information has been lost, it can be difficult if not impossible to reliably estimate the current loss *statistics*, modeled via θ. The research by Costa et al. is an example of estimation where the Markov state θ is unknown [77, 82, 83]. The work in [78, 80, 81] on the other hand considers the state θ known. In [81], the framework they consider is applied to linear predictive coding, as is the case for the current research. In [81] an optimization approach is presented for this predictive coding framework, however with a sub-optimal estimator (in the LMMSE sense, compared to the Kalman filter). In the work presented here, the decoding is based on (LMMSE-optimal) Kalman estimation. The work in [81] represents the tradition of Kalman filtering in predictive coding where the Kalman filter is employed as predictor in the encoder (strictly speaking, the filters in [81] are not Kalman filters but filters with a similar structure but fixed gains in stead of the dynamic Kalman gain.) The work presented here, on the other hand only employs Kalman estimation as decoder.

Fairly recent work on Kalman filtering with erroneous measurements is found in [84, 85] by Mostofi and Murray. The framework presented here does not merely consider measurements lost or received; it considers measurements possibly received with errors and investigates how/when to use the erroneous measurements and their impact on stability of the estimator. This approach is however not directly applicable to the work presented in this thesis, because it requires an invertible measurement matrix and assumes measurement noise negligible compared to the noise incurred by transmission errors. The requirement of an invertible measurement matrix is not fulfilled here, cf. the state-space model of AR source process–see Section 1.3.1. The assumption of negligible measurement noise compared to the transmission noise does not hold generally as the measurement noise in the work presented in this thesis corresponds to quantization noise which is not negligible, particularly in case of low-resolution quantization and low transmission loss probability.

Another example of optimization of a Kalman-based source coding framework for transmission losses is found in [86, 87] by Subasingha, Murthi, and Andersen. This framework is a GMM-based predictive coding framework. It uses Kalman estimation in the role of the predictor as seen in the abovementioned [81]. The optimization for losses is achieved by selecting different fixed Kalman gains for different loss modes of the decoder, in principle familiar to [78].

In the current research it was chosen to follow a different loss modeling approach, as described by Sinopoli et al. in [88]. In this work, the modeling of losses is based on the idea that lost observations can be modeled as infinitely uncertain measurements by letting the measurement noise covariance $\mathbf{R}_n \to \infty$ for n at which the measurement is lost. This results

in a very convenient and intuitive modification to the Kalman filter that allows estimation with lost measurements. One drawback to this approach is that it is more restricted in its ability to model losses and as such only considers independent identically distributed (i.i.d.) losses. This approach is being employed in, e.g., [88–90]. However, the latter works do not consider the framework applied as a coding system and they do not consider optimization of the framework with respect to estimation under losses.

In [91], Schenato generalized the concept of lost measurements to delayed measurements as well. Estimation performance with delayed measurements is also investigated in [92, 93]. This further generalization to delayed measurements was not taken in the current research. Considering estimation from lost observations [88], a situation can arise where the estimation becomes unstable if the loss rate becomes too high. This stability issue is investigated in [88, 94–97]. The estimator stability is however only an issue when the system is unstable, i.e., some or all eigenvalues λ_i of $\mathbf{F}_n$ are $|\lambda_i| > 1$. As explained at the end of Section 1.3, $|\lambda_i| < 0,\ \forall i$ when the system under consideration is a stable AR process. Therefore, it is not an issue in the current work.

The work by Jin, Gupta, and Murray in [98] can be seen as a generalization of the work of Sinopoli et al. from [88] to the two-description case, the simplest incarnation of multiple description coding (MDC). Another example in MDC is [99] by Blind et al. which considers the design of a precoding matrix that robustifies the Kalman filter to packet losses. This line of research has not been investigated in relation to this thesis since it was decided not to extend the work to MDC.

1.6.5 Kalman Filter as Predictor vs. Kalman Filter as Estimator

As mentioned in Section 1.4.3, the Kalman filter can be employed as a predictor in a linear predictive source coder. As a result, the encoder quantizes a prediction residual that is in fact Kalman innovations. As mentioned in Section 1.4.2, the innovations sequence is white which is desirable from a quantization perspective. However, as sketched in Section 1.6.1, optimization of predictive coding for channel impairments points in the direction of leaving residual temporal correlation in the prediction residual, i.e., not whitening it completely. For this reason, it is not desirable to employ the Kalman filter as a predictor in the encoder since this makes a colored prediction residual difficult to achieve by nature of the innovations.

An addition to the above argument against the Kalman filter as a predictor in the encoder, the framework for handling lost observations in Kalman estimation from [88] mentioned in Section 1.6.4 requires the ability to model the source-encoder system by a state-space model of the form (1.38) due to the application of the Kalman estimator in the decoder. Employment of

the Kalman filter as predictor in the encoder renders this state space formulation impossible.

For these reasons, it was chosen to consider a predictive encoder with fixed FIR filters in the encoder as this facilitates control of the coloring of the prediction residual and enables use of the framework from [88] for modeling losses.

It should be mentioned that application of the Kalman filter as predictor in the encoder and the resulting quantization of innovations enables different approaches to modeling of the quantization noise compared to the current work (as described in Publication A.) Some notable examples of such quantization noise modeling are found in [33, 81, 100].

1.7 Cross-Layer Design

Modifying the source coding in such a way as to counter impairments of the transmission channel can be seen as joint source-channel coding, as mentioned in Section 1.6.1. The principle of joint source-channel coding is related to the more general principle of cross-layer protocol design.

The cross-layer protocol design principle stems from the fact that most modern communication protocols are structured according to the ISO Open Systems Interconnection (OSI) model [101]. The OSI model divides communication protocols into seven separate layers of different functionality, the collective implementation of which is commonly referred to as a protocol stack. The seven layers are: application, presentation, session, transport, network, link, and physical. A layer within the protocol stack only communicates directly with its two adjacent layers; the one above and the one below. This principle provides a very modular communication model in which individual layers only need to be concerned with the interfaces to adjacent layers and enables a framework in which a protocol stack can easily be composed from a variety of different protocols at each layer. The same principle does however also limit the potential optimality of the protocol stack somewhat, because a protocol at one layer in principle does not know explicitly what operations take place inside other layers and hence cannot take this into account in configuring its own operation.

In many applications, the above-mentioned difficulties associated with the strong modularity of the OSI model have inspired solutions where adherence to the limited standard interfaces has been violated in order to achieve various benefits through joint optimization of or more extensive collaboration between different layers of the protocol stack. This leads to design or optimization of protocols across several layers of the protocol stack, hence the term 'cross-layer'.

Joint source-channel coding is a principle that by its nature violates the OSI layered model. This is due to the fact that source coding is usually seen

as an application layer operation, while channel coding typically takes place at lower layers such as transport or link layers. It thereby either joins the functionality of two or more layers or at the very least requires some internal information from one layer in order to perform the joint source-channel coding at another. In the work presented here, we specifically consider a modification to the source coding that optimizes the source coding to mitigate channel effects, i.e., loss of transmitted data. This means that the source coding, at the application layer, requires information on the channel conditions from lower layers that are more directly involved with the communication channel. Consider for example a source coding protocol at the application layer transmitting data across a network. In order to optimize the source(-channel) coding for the current probability of packet loss, it will need information to estimate the packet loss rate from, e.g., the transport or network layers. In this way, the work presented in this thesis also needs cross-layer information.

Publication D provides an overview of the cross-layer principle, describes different types of cross-layer communication, and reviews a substantial number of examples from the research literature and classifies them according to the involved layers. At the time of publication of Publication D there were relatively few examples involving source coding and cross-layer optimization; most schemes are centered around the lower layers, i.e., physical, link, network, transport. Publication D lists some examples, many of which involve video coding, see Sections 2.3.2, 2.3.5 and 2.3.6. Other surveys of literature on the cross-layer principle are found in for example [102, 103].

Recent examples of cross-layer optimization involving source coding include [104–107]. In [104] the authors deal with video coding with cross-layer interaction between the application and physical layers. An algorithm is presented for joint optimization of application layer source and channel coding and physical layer rate adaptation. The topic in [105] is video coding as well. This paper presents an optimization that incorporates both physical, media access control (MAC) (link), and application layers in a scheme where the source coding and retransmission requests depend on channel state information. In [106], the topic is distributed source coding in a sensor network where the MAC layer (in the link layer) is adatpted to information from the source coding at the application layer. Likewise, [107] is related to distributed source coding and seeks to optimize data quality under energy efficiency and latency constraints.

Perhaps the most interesting recent publications related to cross-layer design, with relation to the work presented in this thesis, are [84, 85]; also mentioned in Section 1.6.4 as examples of Kalman estimation with erroneous measurements. What makes these papers interesting in relation to cross-layer design is that they specifically recognize the need for cross-layer information in their work as their Kalman estimator (at the application layer) needs information on the current link quality, seen as communication

noise, which is only available at another layer. Consequently, they investigate the optimum strategies for their algorithm in the two cases that such cross layer information is/is not available, respectively.

The subject of joint-source channel coding and cross-layer with a focus on video broascasting is treated extensively in a recent book by Duhamel and Kieffer [108]. Publications such as [108] show that the research area of cross-layer algorithm/protocol design is maturing.

Most examples of cross-layer optimization approach smaller individual problems within the classic OSI model and introduce mechanisms addressing specific details in existing protocols. In [109], the authors approach the cross-layer, or more generally network protocol stack, optimization problem in a more unified way where the communication network is modeled by a network utility maximization problem. The layering of the protocol stack is used as a decomposition of the optimization problem. The paper collects several years' research from different authors and pieces the methods together in a framework that contributes to advancing network protocol design toward a mathematical theory of network architectures.

The work presented in this thesis can, as mentioned, be considered joint-source channel coding, or perhaps more correctly channel-optimized source coding since it is developed from the source coding perspective and modified to account for loss of data in the transmission channel. As such it belongs in the application layer of the OSI model. As pointed out above, it is likely to require cross-layer interaction with lower layers such as transport, network, or link layers. It was chosen in this work to focus on the theoretical aspects of optimizing the source coding framework for transmission loss and so, not to focus on the practical details of implementation in combination with existing protocols at other layers of the protocol stack and the possible interaction with these. This section was provided to shed light on some of the more practical aspects related to the presented work.

1.8 Research Contributions

This section summarizes the main research contributions in Publications A to D.

1.8.1 Publication A

Although published second, this was the first publication written on the main results of this thesis. This paper published in IEEE Transactions on Signal Processing introduces a source coding framework based on generalized DPCM. The framework applies to coding of stationary AR sources of any order.

A design method for optimization of predictive coding at the encoder is introduced for application in combination with a Kalman filter at the

decoder. The optimization algorithm uses a greedy iterative approach. The presented coding framework is used to demonstrate the application of the proposed optimization method.

Results of several Monte Carlo simulations based on the above method are presented for a wide range of i.i.d. transmission loss probabilities in combination with several encoder quantization bit rates and two different scalar quantizer types (Lloyd-Max and uniform). The results demonstrate substantial performance improvements in terms of decoded source signal signal-to-noise ratio (SNR) compared to the same coding framework optimized for no loss (unaware of loss).

1.8.2 Publication B

This paper published in Proceedings of the 17th European Signal Processing Conference (EUSIPCO-2009) is based on the same method and coding framework as in Publication A. Publication B presents results of additional Monte Carlo simulations for correlated transmission losses. The losses under consideration are generated from a Gilbert-Elliott process with mean error burst lengths of 2 and 3. The results show that, although some degradation in SNR compared to the i.i.d. loss case is observed, the method is robust to correlated losses despite the fact that the optimization method explicitly assumes i.i.d. transmission losses.

1.8.3 Publication C

The third paper, published in Proceedings of Data Compression Conference (DCC-2010) is based on the same method as in Publication A. Publication C furthermore generalizes the source coding system model to a formulation allowing source model, predictor and noise feedback filter at the encoder to be of any and generally different orders.

The main contribution of this paper is, however, the investigation of fixed-lag smoothing at the decoder. The system model provides built-in possibility for fixed-lag smoothing. Using fixed-lag smoothing further improves SNR of the decoded source signal at the decoder at the cost of modest additional decoding delay. The paper proves that the presented smoothing approach is guaranteed to decrease the decoded signal MSE under certain requirements on the encoder filters. These requirements are shown by examples to be naturally met by encoder filters optimized by the method presented in Publication A.

1.8.4 Publication D

Publication D is a book chapter published in *Mobile Phone Programming and its Application to Wireless Networking* [113]. This chapter is not di-

rectly in line with Publications A to C, but in stead puts the main results in perspective regarding practical application of the proposed algorithm. Publication D provides an overview of cross-layer network protocol design as mentioned in Section 1.7.

The chapter presents, to the best of our knowledge, the most extensive overview of cross-layer protocol design and optimization at its time of publication, until the publication of [102]. We present an overview of different types of cross-layer interaction, existing frameworks for cross-layer information exchange, and a survey of literature related to wireless networking in which cross-layer optimization was used.

1.8.5 Conclusion

The research conducted in this Ph.D. work has resulted in an algorithm for optimization of predictive coding of AR sources for transmission across channels with loss.

The optimization algorithm takes as its starting point a re-thinking of the source coding operation as an operation producing linear measurements of the source signal. The source process and source encoder are formulated as a state-space model, enabling the use of Kalman filtering for estimating the encoder state from received measurements, thus decoding the source signal. The encoder is represented by the state-space measurement equation at the decoder. Channel loss is modeled as the possible loss of individual measurements.

The optimization algorithm is a greedy off-line approach that designs the filter coefficients of a generalized DPCM encoder. The objective of the optimization problem (design of the filter coefficients) is to minimize the Kalman estimator (decoder) state error variance. This is done iteratively in a greedy sense, minimizing the trace of the state error covariance at each iteration. Although global optimality of the solution can not be guaranteed, Monte Carlo simulations of the proposed algorithm show that it provides substantial improvements in decoded source signal SNR compared to the case where the encoder is unaware of channel loss and thus optimized for no loss.

The algorithm has been demonstrated to be robust to correlated channel losses even though the optimization algorithm assumes i.i.d. losses.

It has been proved that employing fixed-lag smoothing at the decoder, i.e., estimating $\hat{s}_{n-l}$ based on measurements of the source signal up to s_n at time n with a lag l, is guaranteed to reduce the estimated source signal MSE under mild constraints on the encoder filter coefficients which have been demonstrated to be met in practice by the proposed optimization algorithm.

We have furthermore provided an extensive overview of cross-layer optimization. Cross-layer communication issues are important to consider due

to the fact that the proposed algorithm interacts with the source coding, typically at the application layer of a network protocol stack, while requiring channel loss information which will be available from the transport, network, and/or link layers of a protocol stack. Thus the overview of cross-layer optimization puts the main results presented in this thesis in perspective in relation to their practical application.

Based on the proposed optimization algorithm we can conclude that "*it is possible to optimize the filtering operations of a general linear predictive coding framework for transmission across lossy channels to improve decoded signal quality under transmission loss significantly compared to frameworks not optimized for loss*," as stated in the main hypothesis on page 24.

References

[1] C. E. Shannon. „A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (July 1948), 379–423; C. E. Shannon. „A mathematical theory of communication". In: *The Bell System Technical Journal* 27.4 (Oct. 1948), 623–656.

[2] S. Vembu, S. Verdú, and Y. Steinberg. „The source-channel separation theorem revisited". In: *IEEE Transactions on Information Theory* 41.1 (Jan. 1995), pp. 44–54. ISSN: 0018-9448. DOI: 10.1109/18.370119.

[3] T. Cover, A. E. Gamal, and M. Salehi. „Multiple access channels with arbitrarily correlated sources". In: *IEEE Transactions on Information Theory* 26.6 (1980), pp. 648–657. ISSN: 0018-9448.

[4] T. S. Han and S. Verdú. „Approximation theory of output statistics". In: *IEEE Transactions on Information Theory* 39.3 (May 1993), pp. 752–772. ISSN: 0018-9448. DOI: 10.1109/18.256486.

[5] S. Verdú and T. S. Han. „A general formula for channel capacity". In: *IEEE Transactions on Information Theory* 40.4 (July 1994), pp. 1147–1157. ISSN: 0018-9448. DOI: 10.1109/18.335960.

[6] B. M. Oliver. „Efficient Coding". In: *Bell Systems Technical Journal* (July 1952), pp. 724–756.

[7] P. Elias. „Predictive coding–I". In: *IRE Transactions on Information Theory* 1.1 (Mar. 1955), pp. 16–24. ISSN: 0096-1000. DOI: 10.1109/TIT.1955.1055126; P. Elias. „Predictive coding–II". In: *IRE Transactions on Information Theory* 1.1 (Mar. 1955), pp. 24–33. ISSN: 0096-1000. DOI: 10.1109/TIT.1955.1055116.

[8] N. S. Jayant and P. Noll. *Digital Coding of Waveforms - Principles and Applications to Speech and Video*. Prentice Hall, 1984. ISBN: 0-13-211913-7.

[9] N. Farvardin and J. Modestino. „Rate-distortion performance of DPCM schemes for autoregressive sources“. In: *IEEE Transactions on Information Theory* 31.3 (1985), pp. 402–418. ISSN: 0018-9448.

[10] M. Naraghi-Pour and D. L. Neuhoff. „Mismatched DPCM encoding of autoregressive processes“. In: *IEEE Transactions on Information Theory* 36.2 (1990), pp. 296–304. ISSN: 0018-9448. DOI: 10.1109/18.52476.

[11] O. G. Guleryuz and M. T. Orchard. „On the DPCM compression of Gaussian autoregressive sequences“. In: *IEEE Transactions on Information Theory* 47.3 (2001), pp. 945–956. ISSN: 0018-9448. DOI: 10.1109/18.915650.

[12] R. Zamir, Y. Kochman, and U. Erez. „Achieving the Gaussian Rate–Distortion Function by Prediction“. In: *IEEE Transactions on Information Theory* 54.7 (July 2008), pp. 3354–3364. ISSN: 0018-9448. DOI: 10.1109/TIT.2008.924683.

[13] A. N. Kim and T. A. Ramstad. „Improving the Rate-Distortion Performance of DPCM Using Multirate Processing With Application in Low-Rate Image Coding“. In: *IEEE Transactions on Signal Processing* 55.10 (2007), pp. 4958–4968. ISSN: 1053-587X. DOI: 10.1109/TSP.2007.897901.

[14] C. Uhl and O. Macchi. „Stability of a DPCM transmission system with an order t predictor“. In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 40.1 (1993), pp. 50–55. ISSN: 1057-7122. DOI: 10.1109/81.215342.

[15] E. Kimme and F. Kuo. „Synthesis of Optimal Filters for a Feedback Quantization System“. In: *IEEE Transactions on Circuit Theory* 10.3 (Sept. 1963), pp. 405–413. ISSN: 0018-9324.

[16] S. Tewksbury and R. Hallock. „Oversampled, linear predictive and noise-shaping coders of order N > 1“. In: *IEEE Transactions on Circuits and Systems* 25.7 (July 1978), pp. 436–447. ISSN: 0098-4094.

[17] B. Atal and M. Schroeder. „Predictive coding of speech signals and subjective error criteria“. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27.3 (1979), pp. 247–254. ISSN: 0096-3518.

[18] J. Flanagan et al. „Speech Coding“. In: *IEEE Transactions on Communications* 27.4 (Apr. 1979), pp. 710–737. ISSN: 0090-6778.

[19] E. Kreyszig. *Advanced Engineering Mathematics*. 8th ed. New York: John Wiley & Sons, Inc., 1999.

[20] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, 2000. ISBN: 0130224642.

[21] N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. New York: John Wiley & Sons, 1949.

[22] R. E. Kalman. „A New Approach to Linear Filtering and Prediction Problems“. In: *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960). This version is a transcription by John Lukesh from 20 January 2002., pp. 35–45.

[23] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1979, repr. Minneola, New York: Dover Publications Inc., 2005. ISBN: 0-486-43938-0.

[24] A. V. Balakrishnan. *Kalman Filtering Theory*. Los Angeles, CA, USA: Optimization Software, Inc., 1987. ISBN: 0-911-57549-9.

[25] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.

[26] M. S. Grewal and A. P. Andrews. *Kalman Filtering: Theory and Practice*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993. ISBN: 0-13-211335-X.

[27] D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006. ISBN: 0471708585.

[28] V. Gómez. „Wiener-Kolmogorov Filtering and Smoothing for Multivariate Series With State-Space Structure“. In: *Journal of Time Series Analysis* 28.3 (2007), pp. 361–385. DOI: 10.1111/j.1467-9892.2006.00514.x.

[29] J. Gibson, S. Jones, and J. Melsa. „Sequentially Adaptive Prediction and Coding of Speech Signals“. In: *IEEE Transactions on Communications* 22.11 (1974), pp. 1789–1797. ISSN: 0096-2244.

[30] J. Gibson. „Sequentially Adaptive Backward Prediction in ADPCM Speech Coders“. In: *IEEE Transactions on Communications* 26.1 (1978), pp. 145–150. ISSN: 0090-6778.

[31] J. Gibson, V. Berglund, and L. Sauter. „Kalman Backward Adaptive Predictor Coefficient Identification in ADPCM with PCQ“. In: *IEEE Transactions on Communications* 28.3 (Mar. 1980), pp. 361–371.

[32] S. Crisafulli, J. D. Mills, and R. R. Bitmead. „Kalman filtering techniques in speech coding“. In: *1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. 1992, 77–80 vol.1. DOI: 10.1109/ICASSP.1992.225968.

[33] S. V. Andersen, S. H. Jensen, and E. Hansen. „Quantization noise modeling in low-delay speech coding“. In: *Proc. IEEE Workshop on Speech Coding For Telecommunications*. 1997, pp. 65–66. DOI: 10.1109/SCFT.1997.623898.

[34] S. V. Andersen et al. „Analysis-by-synthesis speech coding with quantization noise modeling“. In: *Conference Record of the Thirty-Second Asilomar Conference on Signals, Systems & Computers*. Vol. 1. 1998, 333–337 vol.1. DOI: 10.1109/ACSSC.1998.750881.

[35] S. Subasingha, M. N. Murthi, and S. V. Andersen. „Gaussian Mixture Kalman Predictive Coding of Line Spectral Frequencies“. In: *IEEE Transactions on Audio, Speech, and Language Processing* 17.2 (Feb. 2009), pp. 379–391. ISSN: 1558-7916. DOI: 10.1109/TASL.2008.2008735.

[36] W. R. Gardner and B. D. Rao. „Theoretical analysis of the high-rate vector quantization of LPC parameters“. In: *IEEE Transactions on Speech and Audio Processing* 3.5 (Sept. 1995), pp. 367 –381. ISSN: 1063-6676. DOI: 10.1109/89.466658.

[37] T. Linder, R. Zamir, and K. Zeger. „High-resolution source coding for non-difference distortion measures: multidimensional companding“. In: *IEEE Transactions on Information Theory* 45.2 (Mar. 1999), pp. 548 –561. ISSN: 0018-9448. DOI: 10.1109/18.749002.

[38] J. Li, N. Chaddha, and R. M. Gray. „Asymptotic performance of vector quantizers with a perceptual distortion measure“. In: *IEEE Transactions on Information Theory* 45.4 (May 1999), pp. 1082 – 1091. ISSN: 0018-9448. DOI: 10.1109/18.761252.

[39] R. M. Gray and D. L. Neuhoff. „Quantization“. In: *IEEE Transactions on Information Theory* 44.6 (Oct. 1998), pp. 2325–2383. ISSN: 0018-9448. DOI: 10.1109/18.720541.

[40] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[41] J. Max. „Quantizing for minimum distortion“. In: *IEEE Transactions on Information Theory* 6.1 (1960), pp. 7–12. ISSN: 0018-9448.

[42] S. Lloyd. „Least squares quantization in PCM“. In: *IEEE Transactions on Information Theory* 28.2 (1982). The material in this paper was presented in part at the Institute of Mathematical Statistics Meeting, Atlantic City, NJ, September 10-13, 1957., pp. 129–137. ISSN: 0018-9448.

[43] J. Bucklew and J. Gallagher N. „Some properties of uniform step size quantizers (Corresp.)“ In: *IEEE Transactions on Information Theory* 26.5 (1980), pp. 610–613. ISSN: 0018-9448.

[44] W. R. Bennett. „Spectra of Quantized Signals“. In: *Bell Systems Technical Journal* 27 (1948), pp. 446–472.

[45] V. K. Goyal. „High-rate transform coding: how high is high, and does it matter?“ In: *IEEE International Symposium on Information Theory, 2000. Proceedings.* 2000, pp. 207–. DOI: 10.1109/ISIT.2000.866505.

[46] R. Zamir and M. Feder. „On lattice quantization noise“. In: *IEEE Transactions on Information Theory* 42.4 (July 1996), pp. 1152–1159. ISSN: 0018-9448. DOI: 10.1109/18.508838.

[47] R. M. Gray. „Quantization noise spectra“. In: *IEEE Transactions on Information Theory* 36.6 (Nov. 1990), pp. 1220–1244. ISSN: 0018-9448. DOI: 10.1109/18.59924.

[48] D. Marco and D. L. Neuhoff. „The validity of the additive noise model for uniform scalar quantizers“. In: *IEEE Transactions on Information Theory* 51.5 (2005), pp. 1739–1755. ISSN: 0018-9448. DOI: 10.1109/TIT.2005.846397.

[49] P. H. Westerink, J. Biemond, and D. E. Boekee. „Scalar quantization error analysis for image subband coding using QMFs“. In: *IEEE Transactions on Signal Processing* 40.2 (Feb. 1992), pp. 421–428. ISSN: 1053-587X. DOI: 10.1109/78.124952.

[50] T. M. Apostol. *Mathematical Analysis.* 2nd ed. Addison Wesley, 1974. ISBN: 0201002884.

[51] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. „A new method for the nonlinear transformation of means and covariances in filters and estimators“. In: *IEEE Transactions on Automatic Control* 45.3 (Mar. 2000), pp. 477 –482. ISSN: 0018-9286. DOI: 10.1109/9.847726.

[52] S. J. Julier and J. K. Uhlmann. „Unscented filtering and nonlinear estimation“. In: *Proceedings of the IEEE* 92.3 (2004), pp. 401–422. ISSN: 0018-9219. DOI: 10.1109/JPROC.2003.823141; S. J. Julier and J. K. Uhlmann. „Corrections to “Unscented Filtering and Nonlinear Estimation”“. In: *Proceedings of the IEEE* 92.12 (2004), p. 1958. ISSN: 0018-9219. DOI: 10.1109/JPROC.2004.837637.

[53] C. Masreliez. „Approximate non-Gaussian filtering with linear state and observation relations“. In: *IEEE Transactions on Automatic Control* 20.1 (1975), pp. 107–110. ISSN: 0018-9286.

[54] W.-R. Wu and A. Kundu. „Recursive filtering with non-Gaussian noises“. In: *IEEE Transactions on Signal Processing* 44.6 (1996), pp. 1454–1468. ISSN: 1053-587X. DOI: 10.1109/78.506611.

[55] W. Niehsen. „Robust Kalman filtering with generalized Gaussian measurement noise“. In: *IEEE Transactions on Aerospace and Electronic Systems* 38.4 (Oct. 2002), pp. 1409–1412. ISSN: 0. DOI: 10.1109/TAES.2002.1145765.

[56] Ž. M. Đurović and B. D. Kovačević. „Robust estimation with unknown noise statistics“. In: *IEEE Transactions on Automatic Control* 44.6 (1999), pp. 1292–1296. ISSN: 0018-9286. DOI: 10.1109/9.769393.

[57] M. A. Gandhi and L. Mill. „Robust Kalman Filter Based on a Generalized Maximum Likelihood-Type Estimator“. In: *IEEE Transactions on Signal Processing* 58.2 (2010). Accepted for publication. ISSN: 1053-587X. DOI: 10.1109/TSP.2009.2039731.

[58] T. Kløve and V. I. Korzhik. *Error Detecting Codes, General Theory and their Application in Feedback Communication Systems*. Kluwer Academic Publishers, 1995. ISBN: 0-7923-9629-4.

[59] G. Fairhurst and L. Wood. *Advice to link designers on link Automatic Repeat reQuest (ARQ)*. Request for Comments 3366. Internet Engineering Task Force, Aug. 2002.

[60] Information Sciences Institute, University of Southern California. *Transmission Control Protocol*. Request for Comments 793. Internet Engineering Task Force, Sept. 1981.

[61] H. Schulzrinne et al. *RTP: A Transport Protocol for Real-Time Applications*. Request for Comments 3550. Internet Engineering Task Force, July 2003.

[62] J. Postel. *User Datagram Protocol*. Request for Comments 768. Internet Engineering Task Force, Aug. 1980.

[63] C. A. Rødbro et al. „Hidden Markov model-based packet loss concealment for voice over IP“. In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.5 (Sept. 2006), pp. 1609–1623. ISSN: 1558-7916. DOI: 10.1109/TSA.2005.858561.

[64] D. Persson, T. Eriksson, and P. Hedelin. „Packet Video Error Concealment With Gaussian Mixture Models“. In: *IEEE Transactions on Image Processing* 17.2 (Feb. 2008), pp. 145–154. ISSN: 1057-7149. DOI: 10.1109/TIP.2007.914151.

[65] Z.-H. Tan, P. Dalsgaard, and B. Lindberg. „Automatic speech recognition over error-prone wireless networks“. In: *Speech Communication* 47.1-2 (2005). In Honour of Louis Pols, pp. 220–242. ISSN: 0167-6393. DOI: 10.1016/j.specom.2005.05.007.

[66] Y. Chang and R. Donaldson. „Analysis, Optimization, and Sensitivity Study of Differential PCM Systems Operating on Noisy Communication Channels“. In: *IEEE Transactions on Communications* 20.3 (June 1972), pp. 338–350. ISSN: 0090-6778.

[67] K. Sayood and J. C. Borkenhagen. „Use of residual redundancy in the design of joint source/channel coders“. In: *IEEE Transactions on Communications* 39.6 (1991), pp. 838–846. ISSN: 0090-6778. DOI: 10.1109/26.87173.

[68] T. Eriksson, J. Lindén, and J. Skoglund. „Exploiting interframe correlation in spectral quantization: a study of different memory VQ schemes“. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1996.* Vol. 2. 1996, 765–768 vol. 2. DOI: 10.1109/ICASSP.1996.543233.

[69] T. Eriksson, J. Lindén, and J. Skoglund. „Interframe LSF quantization for noisy channels“. In: *IEEE Transactions on Speech and Audio Processing* 7.5 (1999), pp. 495–509. ISSN: 1063-6676. DOI: 10.1109/89.784102.

[70] M. H. Chan. „The performance of DPCM operating on lossy channels with memory“. In: *IEEE Transactions on Communication* 43.234 (Feb-Mar-Apr 1995), pp. 1686–1696. DOI: 10.1109/26.380240.

[71] P. Kabal. „Ill-conditioning and bandwidth expansion in linear prediction of speech“. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03).* Vol. 1. 2003, I–824–I–827 vol.1.

[72] M. Ghanbari and V. Seferidis. „Efficient H.261-based two-layer video codecs for ATM networks“. In: *IEEE Transactions on Circuits and Systems for Video Technology* 5.2 (1995), pp. 171–175. ISSN: 1051-8215. DOI: 10.1109/76.388066.

[73] A. Fuldseth and T. A. Ramstad. „Robust subband video coding with leaky prediction“. In: *Proc. IEEE Digital Signal Processing Workshop.* 1996, pp. 57–60. DOI: 10.1109/DSPWS.1996.555459.

[74] S. Han and B. Girod. „Robust and efficient scalable video coding with leaky prediction“. In: *Proc. International Conference on Image Processing 2002.* Vol. 2. 2002, II–41–II–44 vol.2. DOI: 10.1109/ICIP.2002.1039882.

[75] T. V. Ramabadran and D. Sinha. „On the selection of measurements in least-squares estimation“. In: *IEEE International Conference on Systems Engineering.* 1989, pp. 221–226. DOI: 10.1109/ICSYSE.1989.48659.

[76] T. V. Ramabadran and D. Sinha. „Speech data compression through sparse coding of innovations“. In: *IEEE Transactions on Speech and Audio Processing* 2.2 (1994), pp. 274–284. ISSN: 1063-6676. DOI: 10.1109/89.279276.

[77] O. L. V. Costa. „Linear minimum mean square error estimation for discrete-time Markovian jump linear systems“. In: *IEEE Transactions on Automatic Control* 39.8 (1994), pp. 1685–1689. ISSN: 0018-9286. DOI: 10.1109/9.310052.

[78] S. C. Smith and P. Seiler. „Estimation with lossy measurements: jump estimators for jump systems“. In: *IEEE Transactions on Automatic Control* 48.12 (Dec. 2003), pp. 2163–2171. ISSN: 0018-9286. DOI: 10.1109/TAC.2003.820140.

[79] A. K. Fletcher, S. Rangan, and V. K. Goyal. „Estimation from lossy sensor data: jump linear modeling and Kalman filtering“. In: *Proceedings of the third international symposium on information processing in sensor networks*. Berkeley, California, USA: ACM Press, 2004, pp. 251–258. DOI: 10.1145/984622.984659.

[80] A. K. Fletcher et al. „Causal and Strictly Causal Estimation for Jump Linear Systems: An LMI Analysis“. In: *40th Annual Conference on Information Sciences and Systems 2006*. Mar. 2006, pp. 1302–1307. DOI: 10.1109/CISS.2006.286665.

[81] A. K. Fletcher et al. „Robust Predictive Quantization: Analysis and Design Via Convex Optimization“. In: *IEEE Journal of Selected Topics in Signal Processing* 1.4 (Dec. 2007), pp. 618–632. ISSN: 1932-4553. DOI: 10.1109/JSTSP.2007.910622.

[82] O. L. V. Costa and S. Guerra. „Stationary filter for linear minimum mean square error estimator of discrete-time Markovian jump systems“. In: *IEEE Transactions on Automatic Control* 47.8 (2002), pp. 1351–1356. ISSN: 0018-9286. DOI: 10.1109/TAC.2002.800745.

[83] O. L. V. Costa and S. Guerra. „Robust linear filtering for discrete-time hybrid Markov linear systems“. In: *International Journal of Control* 75.10 (2002), pp. 712–727. ISSN: 0020-7179. DOI: 10.1080/00207170210139502.

[84] Y. Mostofi and R. M. Murray. „To Drop or Not to Drop: Design Principles for Kalman Filtering Over Wireless Fading Channels“. In: *IEEE Transactions on Automatic Control* 54.2 (Feb. 2009), pp. 376–381. ISSN: 0018-9286. DOI: 10.1109/TAC.2008.2008331.

[85] Y. Mostofi and R. M. Murray. „Kalman filtering over wireless fading channels - How to handle packet drop“. In: *International Journal of Robust and Nonlinear Control* 19.18 (2009), pp. 1993–2015. DOI: 10.1002/rnc.1398.

[86] S. Subasingha, M. N. Murthi, and S. V. Andersen. „On GMM Kalman predictive coding of LSFS for packet loss“. In: *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009.* Apr. 2009, pp. 4105–4108. DOI: 10.1109/ICASSP.2009.4960531.

[87] S. Subasingha, M. N. Murthi, and S. V. Andersen. „A Kalman filtering approach to GMM predictive coding of LSFS for packet loss conditions“. In: *16th International Conference on Digital Signal Processing, 2009*. July 2009, pp. 1–6. DOI: 10.1109/ICDSP.2009.5201111.

[88] B. Sinopoli et al. „Kalman filtering with intermittent observations“. In: *IEEE Transactions on Automatic Control* 49.9 (2004), pp. 1453–1464. ISSN: 0018-9286. DOI: 10.1109/TAC.2004.834121.

[89] X. Liu and A. Goldsmith. „Kalman filtering with partial observation losses“. In: *43rd IEEE Conference on Decision and Control*. Vol. 4. 2004, 4180–4186 Vol.4.

[90] L. Schenato et al. „Foundations of Control and Estimation Over Lossy Networks“. In: *Proceedings of the IEEE* 95.1 (2007), pp. 163–187. ISSN: 0018-9219. DOI: 10.1109/JPROC.2006.887306.

[91] L. Schenato. „Optimal Estimation in Networked Control Systems Subject to Random Delay and Packet Drop“. In: *IEEE Transactions on Automatic Control* 53.5 (June 2008), pp. 1311–1317. ISSN: 0018-9286. DOI: 10.1109/TAC.2008.921012.

[92] M. Epstein et al. „Probabilistic performance of state estimation across a lossy network“. In: *Automatica* 44.12 (2008), pp. 3046 – 3053. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2008.05.026.

[93] L. Shi, L. Xie, and R. M. Murray. „Kalman filtering over a packet-delaying network: A probabilistic approach“. In: *Automatica* 45.9 (2009), pp. 2134 –2140. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2009.05.018.

[94] M. Huang and S. Dey. „Stability of Kalman filtering with Markovian packet losses“. In: *Automatica* 43.4 (Apr. 2007), pp. 598–607. DOI: 10.1016/j.automatica.2006.10.023.

[95] A. S. Leong, S. Dey, and J. S. Evans. „On Kalman Smoothing With Random Packet Loss“. In: *IEEE Transactions on Signal Processing* 56.7 (July 2008), pp. 3346–3351. ISSN: 1053-587X. DOI: 10.1109/TSP.2008.920470.

[96] Y. Mo and B. Sinopoli. „A characterization of the critical value for Kalman filtering with intermittent observations“. In: *47th IEEE Conference on Decision and Control, 2008. CDC 2008.* Dec. 2008, pp. 2692–2697. DOI: 10.1109/CDC.2008.4739119.

[97] K. Plarre and F. Bullo. „On Kalman Filtering for Detectable Systems With Intermittent Observations“. In: *IEEE Transactions on Automatic Control* 54.2 (Feb. 2009), pp. 386–390. ISSN: 0018-9286. DOI: 10.1109/TAC.2008.2008347.

[98] Z. Jin, V. Gupta, and R. M. Murray. „State estimation over packet dropping networks using multiple description coding“. In: *Automatica* 42.9 (2006), pp. 1441 –1452. ISSN: 0005-1098. DOI: DOI:10.1016/j.automatica.2006.03.020.

[99] R. Blind et al. „Robustification and optimization of a Kalman filter with measurement loss using linear precoding“. In: *American Control Conference, 2009. ACC '09.* June 2009, pp. 2222–2227. DOI: 10.1109/ACC.2009.5160322.

[100] E. J. Msechu et al. „Decentralized Quantized Kalman Filtering With Scalable Communication Cost“. In: *IEEE Transactions on Signal Processing* 56.8 (2008), pp. 3727–3741. ISSN: 1053-587X. DOI: 10.1109/TSP.2008.925931.

[101] *Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.* 7498-1. Version 2. ISO/IEC, Nov. 1994.

[102] F. Foukalas, V. Gazis, and N. Alonistioti. „Cross-layer design proposals for wireless mobile networks: a survey and taxonomy“. In: *IEEE Communications Surveys & Tutorials* 10.1 (Jan. 2008), pp. 70–85. ISSN: 1553-877X. DOI: 10.1109/COMST.2008.4483671.

[103] M. Razzaque, S. Dobson, and P. Nixon. „Cross-Layer Architectures for Autonomic Communications“. In: *Journal of Network and Systems Management* 15.1 (Mar. 2007), pp. 13–27. DOI: 10.1007/s10922-006-9051-8.

[104] A. Argyriou. „Distortion-optimized video encoding and streaming in multi-rate wireless lans“. In: *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008.* Mar. 2008, pp. 2169–2172. DOI: 10.1109/ICASSP.2008.4518073.

[105] S. Mathialagan and S. Shanmugavel. „Cross-Layer Optimization Using MIMO System for Wireless Networks“. In: *2009 International Conference on Signal Processing Systems.* May 2009, pp. 651–655. DOI: 10.1109/ICSPS.2009.77.

[106] F. Oldewurtel, J. Ansari, and P. Mähönen. „Cross-Layer Design for Distributed Source Coding in Wireless Sensor Networks“. In: *Sensor Technologies and Applications, 2008. SENSORCOMM '08. Second International Conference on.* Aug. 2008, pp. 435–443. DOI: 10.1109/SENSORCOMM.2008.10.

[107] W. Wang et al. „Cross-layer multirate interaction with Distributed Source Coding in Wireless Sensor Networks“. In: *IEEE Transactions on Wireless Communications* 8.2 (Feb. 2009), pp. 787–795. ISSN: 1536-1276. DOI: 10.1109/TWC.2009.071009.

[108] P. Duhamel and M. Kieffer. *Joint Source-Channel Decoding. A Cross-Layer Perspective with Applications in Video Broadcasting.* Academic Press, Dec. 2009. ISBN: 978-0-12-374449-4.

[109] M. Chiang et al. „Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures“. In: *Proceedings of the IEEE* 95.1 (Jan. 2007), pp. 255–312. ISSN: 0018-9219. DOI: 10.1109/JPROC.2006.887322.

[110] T. Arildsen et al. „On Predictive Coding for Erasure Channels Using a Kalman Framework“. In: *IEEE Transactions on Signal Processing* 57.11 (Nov. 2009), pp. 4456–4466. ISSN: 1053-587X. DOI: 10.1109/TSP.2009.2025796.

[111] T. Arildsen et al. „On Predictive Coding for Erasure Channels Using a Kalman Framework“. In: *Proceedings of the 17th European Signal Processing Conference (EUSIPCO-2009)*. Eurasip. EUSIPCO 2009, Glasgow Ltd, Aug. 2009.

[112] T. Arildsen et al. „Fixed-Lag Smoothing for Low-Delay Predictive Coding with Noise Shaping for Lossy Networks“. In: *Proceedings of Data Compression Conference (DCC-2010)*. Los Alamitos, CA: IEEE Computer Society, Mar. 2010, pp. 279–287. DOI: 10.1109/DCC.2010.33.

[113] F. H. P. Fitzek and F. Reichert, eds. *Mobile Phone Programming and its Application to Wireless Networking.* 1st ed. Dordrecht, The Netherlands: Springer, 2007. ISBN: 978-1-4020-5968-1. DOI: 10.1007/978-1-4020-5969-8.

Publication A

On Predictive Coding for Erasure Channels Using a Kalman Framework

Thomas Arildsen, Manohar N. Murthi, Søren Vang Andersen, and Søren Holdt Jensen

This paper was originally published as:
T. Arildsen et al. „On Predictive Coding for Erasure Channels Using a Kalman Framework“. In: *IEEE Transactions on Signal Processing* 57.11 (Nov. 2009), pp. 4456–4466. ISSN: 1053-587X. DOI: 10.1109/TSP.2009.2025796.

This version is the final version submitted to IEEE for publication. Please note that the published version has undergone minor style-editing in the galley proof process. The current layout has been revised compared to the published version.

Abstract

We present a new design method for robust low-delay coding of AR sources for transmission across erasure channels. It is a fundamental rethinking of existing concepts. It considers the encoder a mechanism that produces signal measurements from which the decoder estimates the original signal. The method is based on LPC and Kalman estimation at the decoder. We employ a novel encoder state-space representation with a linear quantization noise model. The encoder is represented by the Kalman measurement at the decoder. The presented method designs the encoder and decoder offline through an iterative algorithm based on closed-form minimization of the trace of the decoder state error covariance. The design method is shown to provide considerable performance gains, when the transmitted quantized prediction errors are subject to loss, in terms of SNR compared to the same coding framework optimized for no loss. The design method applies to stationary AR sources of any order. We demonstrate the method in a framework based on a generalized DPCM encoder. The presented principles can be applied to more complicated coding systems that incorporate predictive coding as well.

1 Introduction

In transmission of real-time signals data losses are typically an unavoidable impairment. The real-time constraint makes it necessary to consider data with a high transmission delay lost. This delay can for example occur as a result of network congestion. On other types of lossy channels such as wireless links, the real-time constraint makes it impractical to retransmit lost data. Transmission can be protected against losses by, e.g., error correcting codes or multiple description coding (MDC), or the effects of losses on the transmitted signal may be mitigated through various loss concealment techniques at the receiver [2–4]. For low-delay coding applications, error-correcting codes are impractical due to the delay they impose. In such cases, another possibility is to modify the source coding itself to increase robustness against losses.

Linear predictive coding (LPC) has been widely used for source coding, especially speech coding, for a long time. It is one of several source coding techniques in standards used in voice over IP (VoIP) and is widely used in several mobile phone standards [5–11]. LPC works well for signals with temporal correlation where it exploits this correlation to compress the source signal. In a typical LPC source coding system, the predictor in the encoder is an all-zero filter that ideally assumes that the source signal is the outcome of an auto-regressive (AR) process, in which case the predictor can perfectly whiten the source signal. In the decoder, the source signal is reconstructed

from the whitened prediction residual.

The predictor in an LPC source coding system is typically determined by modeling the source signal as the outcome of an AR process for which the coefficients are estimated. The predictor can be chosen to match these estimated AR coefficients, i.e., the coefficients of the prediction filter are equal to the coefficients of the source AR process. This is in general not optimal when the prediction residual is affected by noise, e.g., quantization or channel noise, and better performance can be achieved with a mis-matched predictor [12, 13].

Differential pulse code modulation (DPCM) is an example of a predictive source coding scheme which includes feedback of quantization noise in the coding of the source signal [14]. Kalman filtering can be applied in predictive coding to provide minimum mean squared error (MMSE) estimation of the source signal. Previous applications of Kalman filtering to predictive coding employ Kalman filters at both the encoder and decoder and transmit quantized Kalman innovations from encoder to decoder [15–19].

The effect of channel errors on DPCM performance has been investigated for transmission across ATM networks in [20]. The authors investigate optimization of the predictor for channel losses in a first order DPCM system, but provide no optimization results for higher order coding systems.

When considering a Kalman filter-based decoder, the work in [17, 21] applies to optimizing the Kalman filter for given noise statistics by selecting the optimal measurement vector that minimizes some measure on the a posteriori state error covariance. However, this approach does not take channel losses into account.

The handling of lost measurements in a Kalman estimator is investigated thoroughly in [22, 23], but this work does not consider optimization of a coding system for such losses.

An approach for optimization of a predictive quantization scheme employing Kalman-like filters at encoder and decoder is presented in [24, 25] where channel losses are modeled by a Markov model. [25] is contemporaneous work with a different philosophy; it presents an optimization method based on jump linear system (JLS) modeling and linear matrix inequality (LMI)-constrained convex optimization to design fixed gains for the encoder and decoder filters for each channel state. This approach reduces computational complexity by restricting the decoder to account only for present channel loss through a JLS-based decoder that switches between the states of the channel loss model.

In this paper, we present a novel optimization method for the design of low-delay predictive coding systems, demonstrating a method for designing a robust encoder and decoder for given loss statistics. In particular, we examine DPCM, which is a canonical method of predictive coding which captures the basic problems of real-time transmission over channels with packet loss. In contrast to other efforts to design robust DPCM methods (e.g.,

[20, 25]), we consider a generalized DPCM encoder structure with separate prediction and noise feedback filters, an encoding structure commonly employed in speech coding. Moreover, we consider the case where these encoder filters are fixed time-invariant filters, leading to low-complexity quantization of signal samples. This encoder transmits quantization information (related to quantized prediction errors) that is subject to packet loss/erasure. The decoder views the received information from the encoder as noisy signal measurements, and utilizes Kalman filtering principles to perform MMSE estimation of the signal. This approach of viewing the encoder as producing noisy measurements is in contrast to previous approaches in [15–19], in which the encoder's transmitted quantized prediction error is viewed as the innovation, with both the encoder and decoder running synchronized Kalman filters.

Our predictive coding scheme consists of both offline and online stages. In the offline stage, the fixed encoder filters, and the initial Kalman measurement filter at the decoder are jointly designed, taking into account both the quantization noise and packet loss statistics. In the online operation, the decoder's Kalman filter parameters are updated with each received or lost packet, taking into account the particular loss outcome sequence in the MMSE estimation. Since the encoder remains fixed while the decoder is time-varying, synchronization between encoder and decoder is not assumed. Simulation results demonstrate the efficacy of the proposed method. This low-delay predictive coding design approach can be extended beyond DPCM and to the robust transmission of vector data, such as Line Spectral Frequencies. Therefore, this paper presents a re-thinking of fundamental concepts, and presents a new design method that can be employed in different coding application contexts.

The system model used to illustrate the application of our method and the actual design method are presented in Section 2. Section 3 contains descriptions of simulations conducted to evaluate the performance of the method and results of the simulations showing substantial improvements of the presented method over coding without optimization for loss. Finally, Section 4 discusses the implications of the proposed method and the simulation results.

2 Coding Framework and Design Method

This section describes the source encoder and decoder in Sections 2.1 and 2.2. The optimization for sample losses is treated in Section 2.2.4. The coding framework is summarized in Section 2.4. We provide an overview comparison to the method from [25] in Section 2.5.

2.1 Source Encoder

The source encoder chosen to illustrate the application of our design method is based on generalized DPCM coding. We consider an encoder with the

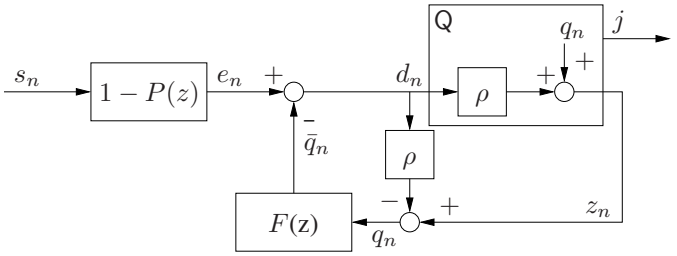


Figure A.1: Generalized DPCM source encoder model with AWN quantizer model and de-correlated quantization noise feedback.

noise shaping structure illustrated in Fig. A.1 as given in [26, 27]. Fig. A.1 includes the quantization noise model described in Section 2.1.1.

The encoder codes the source signal s. The source signal is modeled as outcomes of a stationary AR process

$$s_n = \sum_{i=1}^{N} \alpha_i s_{n-i} + r_n, \tag{A.1}$$

of order N, driven by zero-mean stationary white Gaussian noise r; the α_i are the source AR coefficients, defining the source process together with N.

The encoder has prediction filter P independent from the quantization noise feedback filter F. The structure depicted in Fig. A.1 is equivalent to a classic DPCM encoder, as described in e.g. [14], when $P(z) = F(z)$ and $\rho = 1$.

The input to the quantizer, d_n, is given by

$$d_n = e_n - \bar{q}_n, \tag{A.2}$$

$$e_n = s_n - \sum_{i=1}^{p} a_i s_{n-i}, \tag{A.3}$$

$$\bar{q}_n = \sum_{i=1}^{f} b_i q_{n-i}. \tag{A.4}$$

Note that e_n is the prediction error, $\bar{q}_n$ is the filtered quantization noise feedback, d_n is the input to the quantizer, and $\tilde{q}_n$ is the quantization error; p is the predictor order; $a_i,\ i = 1, \ldots, p$ are the predictor coefficients; f is the noise feedback filter order and $b_i,\ i = 1, \ldots, f$ are the noise feedback filter coefficients. In this work, $p = f = N$.

2.1.1 Linear Quantizer Model

As depicted in Fig. A.1, the output transmitted to the decoder is quantization indices, j, for the quantized prediction error, z_n, in (A.5). As explained in Section 2.2, z_n is seen as the Kalman measurement by the decoder,

$$z_n = Q(d_n) \tag{A.5}$$

We use scalar quantization $Q(\cdot)$ with a gain-plus-additive-noise model [14]. The model accommodates correlation between quantizer input and quantization noise. In fact,

$$z_n = \rho d_n + q_n, \tag{A.6}$$

where $\rho \in [0, 1]$, and q_n is a stationary zero-mean white Gaussian noise, independent of d_n, with variance

$$\sigma_q^2 = k \operatorname{var}\{d_n\}, \tag{A.7}$$

where the quantization noise is modeled with a variance proportional, by a constant k, to the variance of the input to the quantizer, d_n. The assumption of white Gaussian q_n is a simplifying assumption in the sense that quantization noise is generally not Gaussian and only approximately white under high-rate assumptions [28, 29]. In order to be able to model the quantization noise as measurement noise in the Kalman filter in the decoder, this noise must be white Gaussian. There are techniques for handling non-Gaussian measurement noise in a Kalman filter, see [30–32], alternatively, a non-linear measurement model could be accommodated by the extended Kalman filter (EKF) [33]. Nevertheless, we retain the Gaussian assumption in order to keep the Kalman filter of the standard form. This is to avoid non-linear modifications to the covariance updates of the Kalman filter and facilitate the inclusion of the noise model in the optimization approach presented in Section 2.2.4. Note that ρ and k are given by the coding loss, β, of the quantizer:

$$\rho = 1 - \beta \qquad\qquad k = \beta(1 - \beta). \tag{A.8}$$

Note that β is the inverse of the quantizer coding gain [29].

The noise incurred by quantization is

$$\tilde{q}_n = z_n - d_n = (\rho - 1)d_n + q_n. \tag{A.9}$$

In order to simplify the calculation of quantizer input variance in the optimization of the encoder, we wish to feed back a white noise component. Therefore, d_n is scaled by ρ in the quantization noise feedback to de-correlate the noise feedback from d_n. Thus, we only feed back the uncorrelated part of the quantization noise $q_n = z_n - \rho d_n$. This allows us to model the input to the quantizer, d_n, as white Gaussian which simplifies the optimization of the encoder. The design of quantizers for use in the encoder is treated in Section 2.3.

2.2 Kalman Filter-Based Decoder

The decoder is based on Kalman filtering, i.e., MMSE estimation of the source signal s. The Kalman filter at the decoder estimates the source signal based on measurements, z, reconstructed from the received quantization indices, j, which may be subject to losses. In order to derive the Kalman estimator $\hat{s}_n$ of s_n, the source process and encoder equations are modeled by a state space model of the form given in, e.g., [33]. The state transition equation is chosen to represent the evolution of the source signal s_n as well as the states of the encoder filters P and F. The measurement equation represents the filtering and quantization operations of the encoder. So, the measurements become the quantized prediction error outputs from the encoder. Note that where our formulation leads to the quantized prediction error being seen as a Kalman measurement, previous formulations of Kalman predictive coding have mapped this quantity to the Kalman innovation [15–19]. This difference is instrumental for obtaining the robustness to packet loss which we will demonstrate in this paper.

The decoder is derived from the state-space model described below. The process equation is given by (A.10) and the measurement equation by (A.11).

$$\mathbf{x}_{n+1} = \mathbf{F}\mathbf{x}_n + \mathbf{G}\mathbf{w}_n \tag{A.10}$$

$$z_n = \mathbf{h}^{\mathrm{T}}\mathbf{x}_n + q_n \tag{A.11}$$

The state $\mathbf{x}_n$ corresponds to the joint states of the signal, predictor and noise feedback filter.

$$\mathbf{x}_n = \left[\begin{array}{c|ccc|ccc} s_n & s_{n-1} & \cdots & s_{n-p} & q_{n-1} & \cdots & q_{n-f} \end{array}\right]^{\mathrm{T}} \tag{A.12}$$

The state transition matrix $\mathbf{F}$ is defined as follows (the subscripts in (A.13) denote the dimensions of the individual components):

$$\mathbf{F} = \left[\begin{array}{c|c|c} \alpha_{1\times p} & 0 & \mathbf{0}_{(p+1)\times f} \\ \hline \mathbf{I}_p & \mathbf{0}_{p\times 1} & \\ \hline \mathbf{0}_{f\times(p+1)} & & \begin{array}{c|c} \mathbf{0}_{1\times f} & \\ \hline \mathbf{I}_{(f-1)} & \mathbf{0}_{(f-1)\times 1} \end{array} \end{array}\right] \tag{A.13}$$

where $\alpha = [\alpha_1 \cdots \alpha_p]$ are the coefficients of the source AR process, $\mathbf{I}_x$ is an $x \times x$ identity matrix, and $\mathbf{0}$ is an all-zero matrix with the specified dimensions. Thus, the top-left part of $\mathbf{F}$ represents the AR filtering of the process noise, given by (A.1), generating the source signal, and shifts past source signal samples through the state. The bottom-right part of $\mathbf{F}$ delays previous quantization noise samples through the state.

The process noise $\mathbf{w}_n$ is defined as

$$\mathbf{w}_n = \left[\begin{array}{cc} r_n & q_n \end{array}\right]^{\mathrm{T}}, \tag{A.14}$$

which is a stationary zero-mean white Gaussian process noise with

$$\mathbf{Q} = \mathrm{cov}\left\{\mathbf{w}_n, \mathbf{w}_n\right\} = \begin{bmatrix} \mathrm{var}\left\{r_n\right\} & 0 \\ 0 & \mathrm{var}\left\{q_n\right\} \end{bmatrix}. \tag{A.15}$$

Notice that the first component of the process noise, r_n, models the source signal excitation and the second component, q_n, models the quantization noise fed back to the filter F. Clearly, the definition of the process noise (A.14) introduces correlation between the process noise $\mathbf{w}_n$ and the measurement noise in the form of q_n in (A.11). The connection between quantization noise in the state originating from $\mathbf{w}_n$, and q_n added to the measurement is captured by including correlation between process and measurement noise as follows:

$$\mathbf{S} = \mathrm{cov}\left\{\mathbf{w}_n, q_n\right\} = \mathrm{E}\left\{ \begin{bmatrix} r_n \\ q_n \end{bmatrix} q_n \right\} = \begin{bmatrix} 0 \\ \mathbf{R} \end{bmatrix}, \tag{A.16}$$

where $\mathbf{R} = \sigma_q^2$. As shown later, we use a formulation of the Kalman filter which takes the covariance $\mathbf{S}$ into account. Let $\mathbf{G}$ be a transform to allow the process noise $\mathbf{w}_n$ to be defined in a compact form with $\mathbf{G}$ given by

$$\mathbf{G} = \begin{bmatrix} 1 & 0 \\ & \mathbf{0}_{p\times 2} \\ 0 & 1 \\ \mathbf{0}_{(f-1)\times 2} & \end{bmatrix}. \tag{A.17}$$

The measurement vector $\mathbf{h}$ represents the filtering operations of the encoder as well as the scaling in the model of the quantizer

$$\mathbf{h} = \rho\tilde{\mathbf{h}}, \tag{A.18}$$

where $\tilde{\mathbf{h}}$ contains the coefficients of the prediction error and noise feedback filters

$$\tilde{\mathbf{h}} = \begin{bmatrix} 1 & -a_1 & \cdots & -a_p & -b_1 & \cdots & -b_f \end{bmatrix}^{\mathrm{T}} \tag{A.19}$$

such that by (A.2),

$$d_n = \tilde{\mathbf{h}}^{\mathrm{T}}\mathbf{x}_n, \tag{A.20}$$

whereby (A.11) follows from (A.6). To summarize, $\tilde{\mathbf{h}}$ represents the filtering in the encoder before quantization. Due to the quantization noise model presented in Section 2.1.1, $\mathbf{h}$ represents the filtering after quantization and produces the measurements seen by the decoder when these are not lost.

The state-space model (A.10) and (A.11) represents the production of the source signal as well as the encoding of it. This state space model forms the basis of the decoders described in the following sections. First we describe the decoder and the design algorithm for the lossless case. Subsequently, we extend the principles to losses.

2.2.1 Lossless Transmission

The decoder receives information (quantization indices j) to build measurements z_n from the encoder. In the case of lossless transmission, all measurements are received by the decoder. The decoder in this case is given by the Kalman filter with correlated process and measurement noise for the described state space model, (A.10) and (A.11), given in for example [33]

$$\hat{\mathbf{x}}_n = \hat{\mathbf{x}}_n^- + \mathbf{P}_n^-\mathbf{h}\left(\mathbf{h}^{\mathrm{T}}\mathbf{P}_n^-\mathbf{h} + \mathbf{R}\right)^{-1}\left(z_n - \mathbf{h}^{\mathrm{T}}\hat{\mathbf{x}}_n^-\right) \tag{A.21}$$

$$\mathbf{P}_n = \mathbf{P}_n^- - \mathbf{P}_n^-\mathbf{h}\left(\mathbf{h}^{\mathrm{T}}\mathbf{P}_n^-\mathbf{h} + \mathbf{R}\right)^{-1}\mathbf{h}^{\mathrm{T}}\mathbf{P}_n^{-\mathrm{T}} \tag{A.22}$$

$$\hat{\mathbf{x}}_{n+1}^- = \bar{\mathbf{F}}\hat{\mathbf{x}}_n + \mathbf{G}\mathbf{S}\mathbf{R}^{-1}z_n \tag{A.23}$$

$$\mathbf{P}_{n+1}^- = \bar{\mathbf{F}}\mathbf{P}_n\bar{\mathbf{F}}^{\mathrm{T}} + \mathbf{G}\bar{\mathbf{Q}}\mathbf{G}^{\mathrm{T}}, \tag{A.24}$$

in which the following shorthand notation is used:

$$\hat{\mathbf{x}}_n^- = \mathrm{E}\left\{\mathbf{x}_n | z_0, \ldots, z_{n-1}\right\} \qquad \hat{\mathbf{x}}_n = \mathrm{E}\left\{\mathbf{x}_n | z_0, \ldots, z_n\right\}$$

$$\mathbf{P}_n^- = \mathrm{E}\left\{\left(\mathbf{x}_n - \hat{\mathbf{x}}_n^-\right)\left(\mathbf{x}_n - \hat{\mathbf{x}}_n^-\right)^{\mathrm{T}}\right\} \qquad \mathbf{P}_n = \mathrm{E}\left\{\left(\mathbf{x}_n - \hat{\mathbf{x}}_n\right)\left(\mathbf{x}_n - \hat{\mathbf{x}}_n\right)^{\mathrm{T}}\right\}$$

$$\bar{\mathbf{F}} = \left(\mathbf{F} - \mathbf{G}\mathbf{S}\mathbf{R}^{-1}\mathbf{h}^{\mathrm{T}}\right) \qquad \bar{\mathbf{Q}} = \left(\mathbf{Q} - \mathbf{S}\mathbf{R}^{-1}\mathbf{S}^{\mathrm{T}}\right)$$

The decoded source signal $\hat{s}_n$ is given as the first element of $\hat{\mathbf{x}}_n$ according to (A.12).

Since the source signal, $s_,$ is stationary and the encoder fixed (constant $\mathbf{h}$), the Kalman filter statistics will converge to fixed values as $n \to \infty$:

$$\lim_{n\to\infty} \mathbf{P}_n^- = \mathbf{P}^- \qquad\qquad \lim_{n\to\infty} \mathbf{P}_n = \mathbf{P}. \tag{A.25}$$

Correspondingly, we may write the fixed Kalman filter decoder as in (A.26)–(A.28).

$$\hat{\mathbf{x}}_n = \hat{\mathbf{x}}_n^- + \mathbf{P}^-\mathbf{h}\left(\mathbf{h}^{\mathrm{T}}\mathbf{P}^-\mathbf{h} + \mathbf{R}\right)^{-1}\left(z_n - \mathbf{h}^{\mathrm{T}}\hat{\mathbf{x}}_n^-\right) \tag{A.26}$$

$$\hat{\mathbf{x}}_{n+1}^- = \bar{\mathbf{F}}\hat{\mathbf{x}}_n + \mathbf{G}\mathbf{S}\mathbf{R}^{-1}z_n \tag{A.27}$$

where $\mathbf{P}^-$ is the solution to the Riccati equation

$$\mathbf{P}^- = \bar{\mathbf{F}}\mathbf{P}^-\bar{\mathbf{F}}^{\mathrm{T}} - \bar{\mathbf{F}}\mathbf{P}^-\mathbf{h}\left(\mathbf{h}^{\mathrm{T}}\mathbf{P}^-\mathbf{h} + \mathbf{R}\right)^{-1}\mathbf{h}^{\mathrm{T}}\mathbf{P}^{-\mathrm{T}}\bar{\mathbf{F}}^{\mathrm{T}} + \mathbf{G}\bar{\mathbf{Q}}\mathbf{G}^{\mathrm{T}} \tag{A.28}$$

2.2.2 Coder Design for Lossless Transmission

Our optimization of the coding framework is somewhat similar to the approach in [17], but [17] does not consider optimization for sample erasures and the optimization objective has a different structure. The design of the encoder and decoder consists of offline selection of the measurement vector $\mathbf{h}$, as this defines the encoder through (A.18) and (A.19) and defines

the decoder through (A.26)–(A.28). The method is based on choosing the measurement vector $\mathbf{h}^*$ to minimize the mean squared error of the state estimate $\hat{\mathbf{x}}_n$ at the decoder, at time n given the a priori state estimate $\hat{\mathbf{x}}_n^-$ and corresponding state error covariance $\mathbf{P}^-$. If we first look at the situation for the lossless case, described in Section 2.2.1, the objective is

$$\mathbf{h}^* = \arg\min_{\mathbf{h}} \operatorname{Tr}\left[\mathrm{E}\left\{(\mathbf{x}_n - \hat{\mathbf{x}}_n)(\mathbf{x}_n - \hat{\mathbf{x}}_n)^{\mathrm{T}} \mid \hat{\mathbf{x}}_n^-, \mathbf{P}^-\right\}\right], \tag{A.29}$$

which can be written as

$$\mathbf{h}^* = \arg\min_{\mathbf{h}} \operatorname{Tr}\left[\mathbf{P}\right], \tag{A.30}$$

with $\hat{\mathbf{x}}_n$, $\hat{\mathbf{x}}_n^-$, and $\mathbf{P}^-$ given by (A.26)–(A.28), and

$$\mathbf{P} = \mathbf{P}^- - \mathbf{P}^-\mathbf{h}\left(\mathbf{h}^{\mathrm{T}}\mathbf{P}^-\mathbf{h} + \mathbf{R}\right)^{-1}\mathbf{h}^{\mathrm{T}}\mathbf{P}^{-\mathrm{T}}. \tag{A.31}$$

Note that $\mathbf{h}^*$ and $\mathbf{P}^-$, and thereby $\mathbf{P}$, will depend on each other through (A.28) and (A.30): having selected a $\mathbf{h}^*_{(1)}$ according[1] to (A.30) for some $\mathbf{P}^-_{(1)}$, this will yield a new $\mathbf{P}^-_{(2)}$ by (A.28) for $\mathbf{h} = \mathbf{h}^*_{(1)}$, again resulting in a new $\mathbf{h}^*_{(2)}$ by (A.30). Therefore, we use an iterative approach, iterating over (A.28), (A.30) and (A.31) starting from some initial $\mathbf{h}_{(0)}$ and $\mathbf{P}^-_{(1)}$ (to be explained in the following), iterating until convergence. In the following, the index i identifies the iteration number.

Similar to [17], we express the measurement noise covariance–or equivalently, quantization noise variance–$\mathbf{R}_{(i)}$ as a function of $\mathbf{h}_{(i)}$. Because our framework models the encoding as the Kalman measurement and quantization noise as the measurement noise, $\mathbf{R}_{(i)}$ is expressed as follows, cf. Fig. A.1 and (A.2):

$$\begin{aligned}\mathbf{R}_{(i)} &= k \operatorname{var}\{d\} = k\tilde{\mathbf{h}}_{(i)}^{\mathrm{T}}\mathbf{R}_{\mathbf{xx}}\tilde{\mathbf{h}}_{(i)} \\ &= \frac{k}{\rho^2}\mathbf{h}_{(i)}\mathbf{R}_{\mathbf{xx},(i)}\mathbf{h}_{(i)},\end{aligned} \tag{A.32}$$

where $\mathbf{R}_{\mathbf{xx},(i)}$ is the state correlation matrix, which has the following structure

$$\mathbf{R}_{\mathbf{xx},(i)} = \begin{bmatrix} \mathbf{R}_{ss} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_f\mathbf{R}_{(i-1)} \end{bmatrix}. \tag{A.33}$$

[1] We use the subscripts $\cdot_{(0)}, \cdot_{(1)} \ldots$ to label successive iterative calculations of a quantity.

Equations (A.32) and (A.33) allow reformulation of (A.31) as

$$\mathbf{P}_{(i)} = \mathbf{P}^{-}_{(i)} - \frac{\mathbf{P}^{-}_{(i)}\mathbf{h}_{(i)}\mathbf{h}^{\mathrm{T}}_{(i)}{\mathbf{P}^{-}_{(i)}}^{\mathrm{T}}}{\mathbf{h}^{\mathrm{T}}_{(i)}\left(\mathbf{P}^{-}_{(i)} + \frac{k}{\rho^2}\mathbf{R}_{\mathbf{xx},(i)}\right)\mathbf{h}_{(i)}}. \tag{A.34}$$

The minimization stated in (A.30) of the trace of (A.34) can be attained by maximizing the trace of its right-most term.

$$\begin{aligned}\mathbf{h}^{*}_{(i)} &= \arg\min_{\mathbf{h}} \mathrm{Tr}\left[\mathbf{P}^{-}_{(i)}\right] \\ &= \arg\max_{\mathbf{h}} \frac{\mathbf{h}^{\mathrm{T}}{\mathbf{P}^{-}_{(i)}}^{2}\mathbf{h}}{\mathbf{h}^{\mathrm{T}}\left(\mathbf{P}^{-}_{(i)} + \frac{k}{\rho^2}\mathbf{R}_{\mathbf{xx},(i)}\right)\mathbf{h}},\end{aligned} \tag{A.35}$$

where ${\mathbf{P}^{-}_{(i)}}^{2} = {\mathbf{P}^{-}_{(i)}}^{\mathrm{T}}\mathbf{P}^{-}_{(i)}$ since $\mathbf{P}^{-}_{(i)}$ is symmetric. Quantization noise is now taken into account in (A.35), through (A.32). Equation (A.35) may be rewritten as a Rayleigh quotient through a Cholesky factorization of the matrix in the denominator $\mathbf{L}\mathbf{L}^{\mathrm{T}} = \left(\mathbf{P}^{-}_{(i)} + \frac{k}{\rho^2}\mathbf{R}_{\mathbf{xx},(i)}\right)$ where $\mathbf{L}$ is a lower triangular matrix. We define $\mathbf{y} = \mathbf{L}^{\mathrm{T}}\mathbf{x}$ such that

$$\frac{\mathbf{h}^{\mathrm{T}}{\mathbf{P}^{-}_{(i)}}^{2}\mathbf{h}}{\mathbf{h}^{\mathrm{T}}\left(\mathbf{P}^{-}_{(i)} + \frac{k}{\rho^2}\mathbf{R}_{\mathbf{xx},(i)}\right)\mathbf{h}} = \frac{\mathbf{y}^{\mathrm{T}}\mathbf{L}^{-1}{\mathbf{P}^{-}_{(i)}}^{2}\mathbf{L}^{-\mathrm{T}}\mathbf{y}}{\mathbf{y}^{\mathrm{T}}\mathbf{y}}. \tag{A.36}$$

The vector $\mathbf{y}^{*}_{(i)}$ maximizing the right-hand side of (A.36), i.e.,

$$\mathbf{y}^{*}_{(i)} = \arg\max_{\mathbf{y}} \frac{\mathbf{y}^{\mathrm{T}}\mathbf{L}^{-1}{\mathbf{P}^{-}_{(i)}}^{2}\mathbf{L}^{-\mathrm{T}}\mathbf{y}}{\mathbf{y}^{\mathrm{T}}\mathbf{y}}, \tag{A.37}$$

is given as the eigenvector of $\mathbf{L}^{-1}{\mathbf{P}^{-}_{(i)}}^{2}\mathbf{L}^{-\mathrm{T}}$ corresponding to its largest eigenvalue [17]. Clearly, the fractions in (A.35) and (A.37) are invariant to scaling of $\mathbf{x}$ or $\mathbf{y}$, equivalently. As a result, we may take the measurement vector as given by (A.37) with a normalization by the first element of the vector in order to keep $\mathbf{h}_{(i)}$ as formulated in (A.19), with its first element equal to 1. Then

$$\tilde{\mathbf{h}}^{*}_{(i)} = \frac{\mathbf{L}^{-\mathrm{T}}\mathbf{y}^{*}_{(i)}}{c} \qquad\qquad \mathbf{h}^{*}_{(i)} = \rho\tilde{\mathbf{h}}^{*}_{(i)}, \tag{A.38}$$

where c is the first element of the vector $\mathbf{L}^{-\mathrm{T}}\mathbf{y}^{*}_{(i)}$. Having selected $\mathbf{h}^{*}_{(i)}$, $\mathbf{P}^{-}$

is updated according to (A.28):

$$\mathbf{P}^{-}_{(i+1)} = \bar{\mathbf{F}}\mathbf{P}^{-}_{(i)}\bar{\mathbf{F}}^{\mathrm{T}} - \bar{\mathbf{F}}\mathbf{P}^{-}_{(i)}\mathbf{h}^{*}_{(i)}\left(\mathbf{h}^{\mathrm{T}}_{(i)[*]}\mathbf{P}^{-}_{(i)}\mathbf{h}^{*}_{(i)} + \mathbf{R}_{(i)}\right)^{-1}{\mathbf{h}^{*}_{(i)}}^{\mathrm{T}}{\mathbf{P}^{-}_{(i)}}^{\mathrm{T}}\bar{\mathbf{F}}^{\mathrm{T}} + \mathbf{G}\bar{\mathbf{Q}}\mathbf{G}^{\mathrm{T}}, \quad \text{(A.39)}$$

where iteration indices have been omitted on the quantities $\bar{\mathbf{F}}$ and $\bar{\mathbf{Q}}$ to simplify the equation. These quantities are however dependent on $\mathbf{h}^{*}_{(i)}$.

The encoder and decoder are designed by iteratively performing the steps given by (A.37)–(A.39). The algorithm is initiated with initial measurement vector $\mathbf{h}_{(0)}$ set to match the source and $\mathbf{P}_{(0)}$ set to the unique stabilizing solution to (A.28) for $\mathbf{h} = \mathbf{h}_{(0)}$. The algorithm is outlined in Table A.1.

Table A.1: Design Algorithm for Lossless Transmission

Initialize $\mathbf{h}_{(0)}$: $a_l = b_l = \alpha_l,\ \forall l$
Initialize $\mathbf{P}^{-}_{(1)}$ to unique stabilizing solution to (A.28) for $\mathbf{h} = \mathbf{h}_0$
Set ϵ to desired precision and $i = 0$
Set stop difference $= \infty$
while stop difference $> \epsilon$ **do**
  Set $i = i + 1$
  Minimize $\mathbf{P}_{(i)}$ by (A.37)
  Calculate $\mathbf{h}^{*}_{(i)}$ by (A.38)
  Calculate $\mathbf{P}^{-}_{(i+1)}$ by (A.39)
  Set stop difference $= \mathrm{Tr}\left[\mathbf{P}^{-}_{(i)} - \mathbf{P}^{-}_{(i+1)}\right]$
end while
Select $\mathbf{h}^{*}$ as $\mathbf{h}^{*}_{(i)}$

2.2.3 Lossy Transmission

Considering the situation where measurements z_n may be lost, we have a time-varying Kalman filter. The measurement vector $\mathbf{h}_n$ and measurement noise covariance $\mathbf{R}_n$ are time-varying. This models the possible loss of measurements at the decoder. Specifically, we substitute $\mathbf{h}$ and $\mathbf{R}$ in (A.21)–(A.24) by

$$\mathbf{h}_n = \gamma_n \mathbf{h} \quad \text{(A.40)}$$

$$\mathbf{R}_n = \gamma_n \mathbf{R} + (1 - \gamma_n)\,\sigma^2 \mathbf{I}, \quad \text{(A.41)}$$

where γ_n are outcomes of a stationary Bernoulli random process modeling measurement arrival with arrival probability $\Pr\{\gamma_n = 1\} = \bar{\gamma}$ and loss

probability $\Pr\{\gamma_n = 0\} = 1 - \bar{\gamma}$. Note that $\mathbf{R}$ is the measurement noise covariance in the case of no loss and $\sigma^2\mathbf{I}$ is the measurement noise covariance in the case of loss. We let $\sigma^2 \to \infty$ in (A.41), representing infinite uncertainty about the measurement z_n at the decoder when it is lost in transmission. See [22, 34] for other examples of this approach. Replacing $\mathbf{R}$ and $\mathbf{h}$ by (A.40) and (A.41) in (A.21)–(A.24) and taking $\lim_{\sigma^2\to\infty}$, we obtain the equations defining the online filtering operation

$$\hat{\mathbf{x}}_n = \hat{\mathbf{x}}_n^- + \gamma_n \mathbf{P}_n^- \mathbf{h} \left(\mathbf{h}^{\mathrm{T}} \mathbf{P}_n^- \mathbf{h} + \mathbf{R}\right)^{-1} \left(z_n - \mathbf{h}^{\mathrm{T}} \hat{\mathbf{x}}_n^-\right) \tag{A.42}$$

$$\mathbf{P}_n = \mathbf{P}_n^- - \gamma_n \mathbf{P}_n^- \mathbf{h} \left(\mathbf{h}^{\mathrm{T}} \mathbf{P}_n^- \mathbf{h} + \mathbf{R}\right)^{-1} \mathbf{h}^{\mathrm{T}} \mathbf{P}_n^{-\mathrm{T}} \tag{A.43}$$

$$\hat{\mathbf{x}}_{n+1}^- = \bar{\mathbf{F}}\left(\gamma_n\right) \hat{\mathbf{x}}_n + \gamma_n \mathbf{G}\mathbf{S}\mathbf{R}^{-1} z_n \tag{A.44}$$

$$\mathbf{P}_{n+1}^- = \bar{\mathbf{F}}\left(\gamma_n\right) \mathbf{P}_n \bar{\mathbf{F}}\left(\gamma_n\right)^{\mathrm{T}} + \mathbf{G}\bar{\mathbf{Q}}\left(\gamma_n\right) \mathbf{G}^{\mathrm{T}}, \tag{A.45}$$

where

$$\bar{\mathbf{F}}\left(\gamma_n\right) = \left(\mathbf{F} - \gamma_n \mathbf{G}\mathbf{S}\mathbf{R}^{-1}\mathbf{h}^{\mathrm{T}}\right)$$
$$\bar{\mathbf{Q}}\left(\gamma_n\right) = \left(\mathbf{Q} - \gamma_n \mathbf{S}\mathbf{R}^{-1}\mathbf{S}^{\mathrm{T}}\right).$$

As in the lossless case, the decoded source signal $\hat{s}_n$ is given as the first element of $\hat{\mathbf{x}}_n$ according to (A.12).

The important difference between the lossless case and the lossy case is that the decoder equations now depend on sample arrival γ_n. Furthermore, one cannot rely on fixed $\mathbf{P}^-$ and $\mathbf{P}$ in the lossy case since these become stochastic through their dependence on γ_n.

2.2.4 Coder Design for Lossy Transmission

Extending the design method from Section 2.2.2 to the decoder for the lossy case described in Section 2.2.3, we could consider the objective

$$\mathbf{h}_n^* = \arg\min_{\mathbf{h}} \operatorname{Tr}\left[\mathbf{P}_n | \gamma_0 \ldots \gamma_n\right], \tag{A.46}$$

to obtain a $\mathbf{h}_n^*$ at each time step n optimized for all arrivals $\gamma_0 \ldots \gamma_n$. Hereby we would minimize the trace of (A.43) rewritten via (A.32) and (A.33) as

$$\mathbf{P}_n = \mathbf{P}_n^- - \gamma_n \frac{\mathbf{P}_n^- \mathbf{h}_n \mathbf{h}_n^{\mathrm{T}} \mathbf{P}_n^{-\mathrm{T}}}{\mathbf{h}_n^{\mathrm{T}} \left(\mathbf{P}_n^- + \frac{k}{\rho^2}\mathbf{R}_{\mathbf{xx}}\right) \mathbf{h}_n}. \tag{A.47}$$

Since this optimization would minimize the trace of (A.47), we can see that the optimization is not defined at loss events, i.e., at n for which $\gamma_n = 0$. $\mathbf{P}_n$ is independent of $\mathbf{h}$ in the event of a loss, since $\mathbf{P}_n = \mathbf{P}_n^-$ in this case. Furthermore, since $\mathbf{h}$ defines *both* the encoder and decoder, (A.46) would

require the encoder to know γ_n which in turn requires instantaneous loss-less feedback of this information from decoder to encoder.

Instead, we seek a method that allows offline calculation *of a constant* $\mathbf{h}^*$, *given the statistics of loss.* So, the goal is a method that improves decoding performance under average loss conditions rather than the specific loss outcomes. In contrast to the usual Kalman filter, $\mathbf{P}_n$ is stochastic due to measurement losses γ_n. We propose the following offline method for designing measurement vectors for improved performance under sample losses. Ideally, it would be desirable to obtain a $\mathbf{h}^*$ that at each n minimizes the expectation of $\mathbf{P}_n$ with respect to all $\gamma_k,\ k = 0, \dots n$, i.e., $\mathrm{E}_{\gamma_0 \dots \gamma_n}\{\mathbf{P}_n\}$. However, it is not possible to directly calculate this expectation, a fact which is also pointed out in [22]. We use a simplified approach where the philosophy is to obtain a $\mathbf{h}^*$ that minimizes the ensemble average of $\mathbf{P}_n$ over γ.

The method is a modification of the design for lossless transmission presented in Section 2.2.2. At each iteration i, $\mathbf{h}_{(i)}$ is selected to minimize the trace of $\mathrm{E}_\gamma\left\{\mathbf{P}_{(i)}\right\}$, the ensemble average of $\mathbf{P}_{(i)}$, (A.43), with respect to γ (the measurement loss process). This requires the arrival probability $\bar{\gamma}$ to be known in order to design the encoder and decoder. $\mathbf{P}^-_{(i)}$ is updated according to the discrete-time Riccati equation, [33, p. 108], of the decoder Kalman filter, adapted for measurement losses

$$\mathbf{P}^-_{(i+1)} = \mathbf{F}\mathbf{P}^-_{(i)}\mathbf{F}^{\mathrm{T}} + \mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}} - \gamma \frac{\left(\mathbf{F}\mathbf{P}^-_{(i)}\mathbf{h}^*_{(i)} + \mathbf{G}\mathbf{S}\right)\left(\mathbf{F}\mathbf{P}^-_{(i)}\mathbf{h}^*_{(i)} + \mathbf{G}\mathbf{S}\right)^{\mathrm{T}}}{\mathbf{h}^{*\,\mathrm{T}}_{(i)}\left(\mathbf{P}^-_{(i)} + \frac{k}{\rho^2}\mathbf{R}_{\mathbf{xx},(i)}\right)\mathbf{h}^*_{(i)}}. \quad \text{(A.48)}$$

We take the ensemble average of (A.48) with respect to γ

$$\mathrm{E}_\gamma\left\{\mathbf{P}^-_{(i+1)}\right\} = \mathbf{F}\,\mathrm{E}_\gamma\left\{\mathbf{P}^-_{(i)}\right\}\mathbf{F}^{\mathrm{T}} + \mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}} - \bar{\gamma}\frac{\left(\mathbf{F}\,\mathrm{E}_\gamma\left\{\mathbf{P}^-_{(i)}\right\}\mathbf{h}^*_{(i)} + \mathbf{G}\mathbf{S}\right)\left(\mathbf{F}\,\mathrm{E}_\gamma\left\{\mathbf{P}^-_{(i)}\right\}\mathbf{h}^*_{(i)} + \mathbf{G}\mathbf{S}\right)^{\mathrm{T}}}{\mathbf{h}^{*\,\mathrm{T}}_{(i)}\left(\mathrm{E}_\gamma\left\{\mathbf{P}^-_{(i)}\right\} + \frac{k}{\rho^2}\mathbf{R}_{\mathbf{xx},(i)}\right)\mathbf{h}^*_{(i)}}. \quad \text{(A.49)}$$

The measurement vector is selected at each iteration according to (A.50), i.e.,

$$\mathbf{h}^*_{(i)} = \arg\max_{\mathbf{h}} \frac{\mathbf{h}^{\mathrm{T}}\,\mathrm{E}_\gamma\left\{\mathbf{P}^{-\,2}_{(i)}\right\}\mathbf{h}}{\mathbf{h}^{\mathrm{T}}\left(\mathrm{E}_\gamma\left\{\mathbf{P}^-_{(i)}\right\} + \frac{k}{\rho^2}\mathbf{R}_{\mathbf{xx},(i)}\right)\mathbf{h}}, \quad \text{(A.50)}$$

which now takes both quantization noise and loss of measurements into account.

Equation (A.50) may be rewritten as a Rayleigh quotient by the same approach as in Section 2.2.2, cf. (A.36). We define $\mathbf{y} = \mathbf{L}^{\mathrm{T}}\mathbf{x}$ similar to Section 2.2.2, replacing $\mathbf{P}^{-}_{(i)}$ by $\mathrm{E}_{\gamma}\left\{\mathbf{P}^{-}_{(i)}\right\}$ such that

$$\mathbf{y}^{*}_{(i)} = \arg\max_{\mathbf{y}} \frac{\mathbf{y}^{\mathrm{T}}\mathbf{L}^{-1}\,\mathrm{E}_{\gamma}\left\{{\mathbf{P}^{-}_{(i)}}^{2}\right\}\mathbf{L}^{-\mathrm{T}}\mathbf{y}}{\mathbf{y}^{\mathrm{T}}\mathbf{y}}, \tag{A.51}$$

is given as the eigenvector of $\mathbf{L}^{-1}\,\mathrm{E}_{\gamma}\left\{{\mathbf{P}^{-}_{(i)}}^{2}\right\}\mathbf{L}^{-\mathrm{T}}$ corresponding to its largest eigenvalue. As in Section 2.2.2, the measurement vector is calculated as

$$\tilde{\mathbf{h}}^{*}_{(i)} = \frac{\mathbf{L}^{-\mathrm{T}}\mathbf{y}^{*}_{(i)}}{c} \qquad\qquad \mathbf{h}^{*}_{(i)} = \rho\tilde{\mathbf{h}}^{*}_{(i)}, \tag{A.52}$$

where c is the first element of the vector $\mathbf{L}^{-\mathrm{T}}\mathbf{y}^{*}_{(i)}$.

Equations (A.49), (A.51) and (A.52) are iterated until convergence of (A.49), upon which the resulting $\mathbf{h}^{*}_{(i)}$ is chosen as fixed measurement vector $\mathbf{h}^{*}$ for the decoder given by (A.42)–(A.45) and the corresponding $\tilde{\mathbf{h}}^{*}$ for the encoder. The optimization method is summarized in Table A.2.

Table A.2: Design Algorithm for Lossy Transmission

Initialize $\mathbf{h}_{(0)}$: $a_l = b_l = \alpha_l,\ \forall l$
Initialize $\mathrm{E}_{\gamma}\left\{\mathbf{P}^{-}_{(1)}\right\}$ to unique stabilizing solution to (A.28) for $\mathbf{h} = \mathbf{h}_0$
Set ϵ to desired precision and $i = 0$
Set stop difference $= \infty$
while stop difference $> \epsilon$ **do**
  Set $i = i + 1$
  Minimize $\mathrm{E}_{\gamma}\left\{\mathbf{P}_{(i)}\right\}$ by (A.51)
  Calculate $\mathbf{h}^{*}_{(i)}$ by (A.52)
  Calculate $\mathrm{E}_{\gamma}\left\{\mathbf{P}^{-}_{(i+1)}\right\}$ by (A.49)
  Set stop difference $= \mathrm{Tr}\left[\mathrm{E}_{\gamma}\left\{\mathbf{P}^{-}_{(i)}\right\} - \mathrm{E}_{\gamma}\left\{\mathbf{P}^{-}_{(i+1)}\right\}\right]$
end while
Select $\mathbf{h}^{*}$ as $\mathbf{h}^{*}_{(i)}$

2.3 Quantizer Design

In general, it is not a trivial matter to design a quantizer for a predictive quantization system. The optimal quantizer depends on the encoder filters, and the encoder filters depend on the quantization noise. Therefore,

existing approaches proceed by iteratively optimizing the filters and the quantizer in turns. The optimum design of quantizers for predictive quantization schemes has been treated in the literature, e.g., in [35]. In this paper, we concentrate on the optimization of the encoder and decoder filters to improve decoding performance with respect to sample losses. The impact of quantization in the decoder plays a secondary role compared to the loss of transmitted data. Therefore, we choose a simpler suboptimal approach to the design of quantizers.

As the encoder (and decoder) is designed for a specific loss probability by changing the encoder filters, P and F, accordingly, the statistics of the input to the quantizer, d_n, generally vary with the loss probability. Therefore, the quantizer should also be adapted for the specific loss probability in order to be appropriately loaded.

The source process is an AR process driven by zero-mean white Gaussian noise. As seen from (A.1), the source signal is a sum of Gaussian random variables and so, is Gaussian. The prediction filter output e_n is zero-mean Gaussian by the same argument. Since the prediction filter in general does not match the source (generally $a_i \neq \alpha_i,\ \forall i$), the prediction residual is not white. According to the quantization noise model presented in Section 2.1.1, the noise, $\bar{q}_n$, fed back to the quantizer input is also zero-mean Gaussian. Under the model assumptions, the input, d_n, to the quantizer is thus zero-mean Gaussian.

The quantizer in the encoder is designed based on the statistics of the input in this case, the Gaussian p.d.f. with zero mean and variance calculated as follows. Equation (A.32) in Section 2.2.2 states the quantization noise variance for the time-varying case used in the optimization algorithm. In the coding framework, for a fixed measurement vector $\mathbf{h}$, all signals in the encoder are stationary and so, (A.32) reduces to (A.53), i.e.,

$$\mathbf{R} = k \operatorname{var}\{d_n\} = \frac{k}{\rho^2} \mathbf{h}^{\mathrm{T}} (\mathbf{A} + \mathbf{B}) \mathbf{h}, \tag{A.53}$$

where

$$\begin{aligned} \mathbf{A} &= \left[\begin{array}{c|c} \mathbf{R}_{ss} & \mathbf{0}_{(p+1)\times f} \\ \hline \mathbf{0}_{f\times(p+1)} & \mathbf{0}_{(f\times f)} \end{array}\right] \\ \mathbf{B} &= \mathbf{R} \left[\begin{array}{c|c} \mathbf{0}_{(p+1)\times(p+1)} & \mathbf{0}_{(p+1)\times f} \\ \hline \mathbf{0}_{f\times(p+1)} & \mathbf{I}_{(f\times f)} \end{array}\right]. \end{aligned} \tag{A.54}$$

From (A.18), (A.19), (A.53) and (A.54) we can calculate the quantizer input variance as follows:

$$\operatorname{var}\{d_n\} = \frac{\mathbf{h}^{\mathrm{T}} \mathbf{A} \mathbf{h}}{\rho^2 - k \sum_{i=1}^{f} {b_i}^2}. \tag{A.55}$$

We have considered both Lloyd-Max and uniform quantization for the coding framework presented in this paper.

Lloyd-Max quantizers can be designed to match a specific input p.d.f. using the "Lloyd II" algorithm [14]. Lloyd's and Max's original quantizers for Gaussian input can be found in [36, 37] and scaled according to input variance.

Uniform quantizers can be designed to match a Gaussian input p.d.f. using the expression for the step size in [38].

The coding loss, β, and corresponding parameters, ρ and k, are estimated empirically for the quantizer. These parameters are independent of quantizer scaling, provided that the quantizer is optimally loaded for the given input, and only depend on the quantizer type, uniform or Lloyd-Max, and resolution. So β is estimated as follows:

1. Design a quantizer, $Q(x)$, (Lloyd-Max or uniform) with given precision for a unit-variance zero-mean Gaussian distribution, $f_X(x)$.
2. Generate a random sequence of data, x, according to the distribution $f_X(x)$.
3. Quantize x: $y = Q(x)$.
4. Estimate β as shown in (A.56), cf. definition of coding gain in [29].

$$\hat{\beta} = \frac{\mathrm{E}\{x-y\}^2}{\sigma_x^2} \tag{A.56}$$

The estimate $\hat{\beta}$ for a particular quantizer (type and resolution) is used as β in the calculation of quantization model parameters k and ρ in the encoder and decoder presented in Sections 2.1 and 2.2.

2.4 Summary of Coding Framework

For transmission across erasure channels, the framework presented in Sections 2.1 and 2.2 operates as follows:

- It is assumed that both the encoder and the decoder know the source signal model $\{\alpha_i,\ i = 1, \ldots, N\}$, $\mathrm{var}\{r\}$, and channel arrival probability $\bar{\gamma}$.
- The encoder and decoder parameters in the form of $\mathbf{h}$ are designed according to the method in Sections 2.2.2 and 2.2.4 in case of lossless transmission. This is done offline in both encoder and decoder, respectively.
- The encoder filter parameters a_i and b_i, $i = 1, \ldots, N$ are obtained from the designed $\mathbf{h}^*$ by (A.18) and (A.19).

- The quantizer $Q(\cdot)$ is designed as outlined in Section 2.3.
- The encoder codes the source signal according to (A.2)–(A.4) and transmits quantization indices, j, for reconstruction of z_n at the decoder, which requires the decoder to know the quantizer codebook.
- The decoder receives quantization indices from the encoder with a probability of $\bar{\gamma}$ (equal to 1 in case of lossless transmission) and decodes the source signal depending on whether the current index was lost or not ($\gamma_n = 0$ / $\gamma_n = 1$), using $\mathbf{h}^*$ in (A.42)–(A.45) ((A.26)–(A.28) in case of lossless transmission.)

2.5 Comparison to Related Method

As mentioned in the introduction, [25] presents a method for robust predictive quantization. This section presents an overview comparison illustrating important differences between [25] and our proposed method. We shall refer to our method as Iterative Measurement Vector Improvement (IMVI) and the method in [25] as Gain Vector Search (GVS).

The encoders are linear time-invariant (LTI) systems, both in the case of GVS and IMVI, whereas the decoders are generally time-varying. The decoder in GVS varies as a JLS according to the state of the Markov loss model, with a fixed set of encoder/decoder gains for each state. Thus, the decoding in GVS only depends on the current state of loss. Our decoder in IMVI varies both according to sample loss, γ_n, as well as the time-varying Kalman filter statistics, $\mathbf{P}_n^-$ and $\mathbf{P}_n$. The Kalman filter statistics encompass the effects of all previous losses, so the decoding in IMVI at time n depends on loss at time n as well as all previous losses.

The GVS method works by optimizing two different Kalman-like filters, at the encoder and decoder respectively. Both the encoder and decoder filters are identical in structure to a Kalman filter, but the filter gains are not calculated in the same manner as in Kalman filters. Our IMVI method employs an actual Kalman filter, but only at the decoder. The encoder relies on fixed finite impulse response (FIR) filters. The IMVI method applied to the coding framework in this paper is based on an encoder structure in which the filtering of quantized prediction errors has been split into separate prediction error and quantization noise feedback parts. This offers a higher degree of freedom in encoder design.

The GVS framework accommodates auto-regressive moving average (ARMA) source models, while our model is restricted to accommodate AR source models in its current form.

The GVS framework uses a general Markov loss model, whereas our model only explicitly accommodates i.i.d. losses. Other types of (non-i.i.d.) losses such as Gilbert-Elliot loss models can be handled in our IMVI framework in terms of overall loss probability.

3 Results

3.1 Simulations

Simulations have been conducted to evaluate the performance of the optimization method proposed in Section 2.2.4.

This paper has supplementary downloadable material available at the web page [39], provided by the authors. This includes all Matlab code necessary to fully reproduce the simulation results in this section.

- For testing IMVI, stationary random source signals were generated from AR processes of different orders. Sample arrivals γ_n were simulated as outcomes of a Bernoulli random process over a series of loss probabilities $\bar{\gamma} \in [0, 1]$ and applied to the transmitted encoder quantization indices j. The generated source signals were encoded with encoder and decoder designed for each specified loss probability ($\bar{\gamma}$ in (A.49)). The quantization indices with losses were decoded using the Kalman decoder given by (A.42)–(A.45).

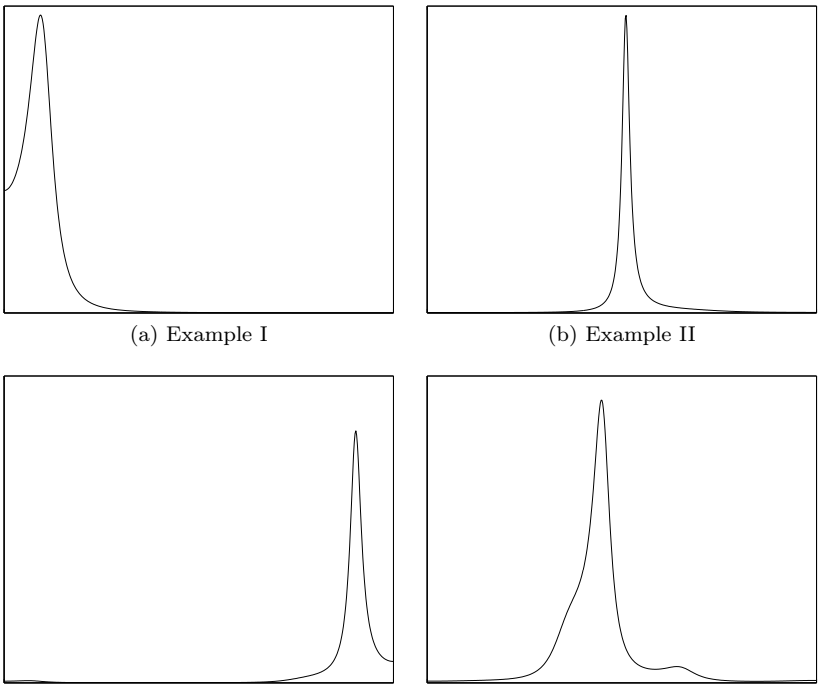
- As a baseline for comparison for IMVI, source signals were generated in the same manner as for IMVI above. The generated source signals were encoded with encoder and decoder designed for no loss ($\bar{\gamma} = 0$). The quantization indices, j, subject to the same losses as above for IMVI were decoded using the Kalman decoder given by (A.42)–(A.45). We shall denote this baseline method "Baseline".

The simulations have been conducted for both uniform and Lloyd-Max quantizers at 2, 3, and 4 bits/sample, respectively.

Test data were generated from statistical models estimated from signals encountered in speech: AR coefficients were estimated from 20ms subsequences selected from voice-active regions of speech found in [40]. The coefficients represent sequences with both low-pass, band-pass, and high-pass spectral shapes. For each of the examples we have plotted the power spectrum of the source AR process in Fig. A.2.

Decoded signal signal-to-noise ratio (SNR) is compared for IMVI and Baseline. In the following, we present results of the simulations described in Section 3.1. We present data from four different examples of source data produced from AR source processes.

The decoded signal SNRs are plotted in Fig. A.3a, A.3b, A.4a and A.4b for the simulated range of loss probabilities at quantization rates of 2, 3, and 4 bits/sample. Baseline-{2,3,4} and IMVI-{2,3,4} respectively. Examples I and II have been produced with Lloyd-Max quantization. Examples III and IV have been produced with uniform quantization. The source processes of the examples are of third, fifth, ninth, and tenth order, respectively.



(a) Example I (b) Example II

(c) Example III (d) Example IV

Figure A.2: Power spectra of the source signals used in examples I-IV.

3.2 Numerical Examples

The source AR process for Example I is a third-order process with the following parameters: $\alpha_1 = 1.6898$, $\alpha_2 = -0.7865$, $\alpha_3 = -0.0035$. The parameters correspond to the matched prediction parameters shown in Table A.3 (with $F = P$). The filter parameters for Baseline-2 are shown

Table A.3: Matched prediction parameters for Example I.

a_1	a_2	a_3	b_1	b_2	b_3
1.6898	−0.7865	−0.0035	1.6898	−0.7865	−0.0035

in Table A.4. The filter parameters designed for the specific loss rates at 2 bits/sample quantization (IMVI-2) are shown in Table A.5. The accompanying quantizer parameters are listed in Table A.6; the quantizer parameters for Baseline-2 are the parameters for $\bar{\gamma} = 0$ at all loss probabilities. Parameters for the remaining cases of Example I (Baseline-3, -4 and IMVI-3, -4) as well as for Examples II-IV have been omitted to save space. For

Table A.4: Filter parameters for Baseline-2, Example I.

a_1	a_2	a_3	b_1	b_2	b_3
1.1190	0.1758	−0.4515	1.7468	−0.5987	−0.2181

Table A.5: Filter parameters designed for the specific loss rates in Example I (IMVI-2).

$\bar{\gamma}$ [%]	a_1	a_2	a_3	b_1	b_2	b_3
0	1.1194	0.1752	−0.4513	1.7468	−0.5990	−0.2178
0.10	1.1079	0.1755	−0.4474	1.7349	−0.5947	−0.2163
0.14	1.1030	0.1760	−0.4458	1.7300	−0.5928	−0.2158
0.21	1.0960	0.1768	−0.4435	1.7233	−0.5902	−0.2151
0.30	1.0860	0.1785	−0.4405	1.7141	−0.5866	−0.2142
0.43	1.0721	0.1817	−0.4364	1.7018	−0.5813	−0.2132
0.62	1.0526	0.1873	−0.4312	1.6855	−0.5738	−0.2124
0.89	1.0255	0.1971	−0.4247	1.6641	−0.5628	−0.2120
1.27	0.9883	0.2129	−0.4168	1.6363	−0.5467	−0.2127
1.83	0.9376	0.2375	−0.4072	1.6003	−0.5232	−0.2153
2.64	0.8690	0.2732	−0.3952	1.5538	−0.4891	−0.2206
3.79	0.7771	0.3212	−0.3783	1.4940	−0.4409	−0.2287
5.46	0.6559	0.3794	−0.3512	1.4178	−0.3751	−0.2379
7.85	0.5002	0.4378	−0.3037	1.3224	−0.2896	−0.2436
11.29	0.3086	0.4744	−0.2218	1.2071	−0.1859	−0.2382
16.24	0.0888	0.4528	−0.0939	1.0767	−0.0737	−0.2139
23.36	−0.1442	0.3305	0.0762	0.9409	0.0301	−0.1689
33.60	−0.3799	0.0695	0.2519	0.8055	0.1107	−0.1102
48.33	−0.6172	−0.3422	0.3224	0.6565	0.1633	−0.0437
69.52	−0.8542	−0.8011	0.0186	0.4091	0.1811	0.0418
100	−1.0983	−1.0983	−1.0000	0	0	0

the remaining examples, we show the decoded signal SNRs in Fig. A.3a, A.3b, A.4a and A.4b.

3.3 Summary

The examples show substantial improvements in decoded signal SNR under sample erasure conditions. For all examples, the improvement is rather modest at low loss probability, especially at the lowest quantization rate (2 bits/sample), improving for higher quantization rates (3 and 4 bits/sample). For higher loss rates, the improvement in decoded signal SNR is substantial. The maximum decoded signal SNR observed in examples I-IV are shown in Table A.7.

At 2 bits/sample, IMVI demonstrates a maximum improvement in decoded signal SNR in the range 1.7 to 3.7 dB, at 3 bits/sample, we see improvements in the range 3.3 to 7.1 dB, and at 4 bits/sample, the method shows improvements in the range 4.2 to 9.7 dB.

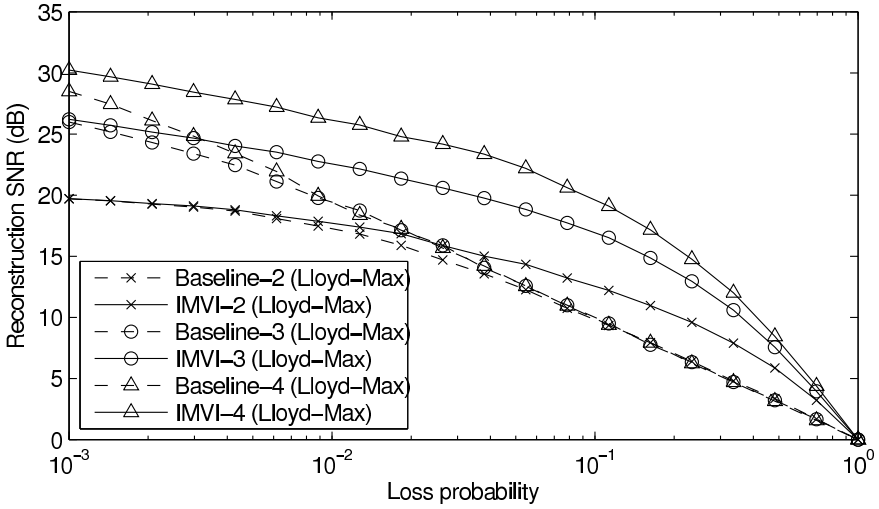
The examples I-IV demonstrate that IMVI is capable of substantially improving the coding performance of the coding framework presented in Sec-

Table A.6: Quantizer parameters used in Example I, IMVI-2.

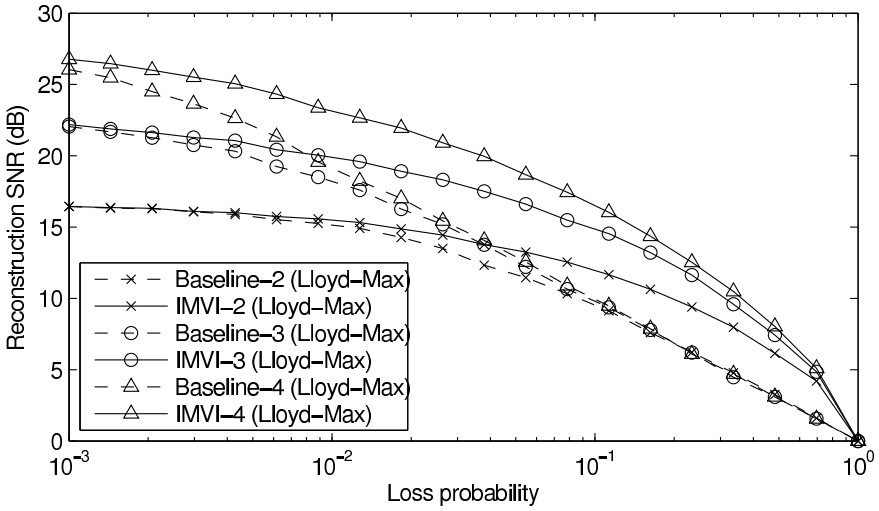
$$\beta = 0.1174$$

$\bar{\gamma}$ [%]	Partition		
0	-1.4108	-1.9106×10^{-7}	1.4108
0.10	-1.4134	-1.9141×10^{-7}	1.4134
0.14	-1.4151	-1.9164×10^{-7}	1.4151
0.21	-1.4181	-1.9205×10^{-7}	1.4181
0.30	-1.4232	-1.8600×10^{-7}	1.4232
0.43	-1.4318	-1.8713×10^{-7}	1.4318
0.62	-1.4462	-1.8901×10^{-7}	1.4462
0.89	-1.4694	-1.9204×10^{-7}	1.4694
1.27	-1.5061	-1.8996×10^{-7}	1.5061
1.83	-1.5631	-1.9026×10^{-7}	1.5631
2.64	-1.6501	-1.8706×10^{-7}	1.6501
3.79	-1.7822	-1.8815×10^{-7}	1.7822
5.46	-1.9819	-1.8806×10^{-7}	1.9819
7.85	-2.2836	-1.8795×10^{-7}	2.2836
11.29	-2.7348	-1.8840×10^{-7}	2.7348
16.24	-3.3968	-1.8903×10^{-7}	3.3968
23.36	-4.3657	-1.8940×10^{-7}	4.3657
33.60	-5.8621	-1.9133×10^{-7}	5.8621
48.33	-8.3731	-1.9147×10^{-7}	8.3731
69.52	-12.5918	-1.8789×10^{-7}	12.5918
100	-19.3211	-1.8813×10^{-7}	19.3211

$\bar{\gamma}$ [%]	Codebook			
0	-2.1708	-0.6507	0.6507	2.1708
0.10	-2.1748	-0.6519	0.6519	2.1748
0.14	-2.1774	-0.6527	0.6527	2.1774
0.21	-2.1820	-0.6541	0.6541	2.1820
0.30	-2.1899	-0.6565	0.6565	2.1899
0.43	-2.2032	-0.6605	0.6605	2.2032
0.62	-2.2253	-0.6671	0.6671	2.2253
0.89	-2.2610	-0.6778	0.6778	2.2610
1.27	-2.3175	-0.6947	0.6947	2.3175
1.83	-2.4052	-0.7210	0.7210	2.4052
2.64	-2.5391	-0.7612	0.7612	2.5391
3.79	-2.7423	-0.8221	0.8221	2.7423
5.46	-3.0496	-0.9142	0.9142	3.0496
7.85	-3.5139	-1.0534	1.0534	3.5139
11.29	-4.2081	-1.2615	1.2615	4.2081
16.24	-5.2267	-1.5668	1.5668	5.2267
23.36	-6.7176	-2.0137	2.0137	6.7176
33.60	-9.0202	-2.7040	2.7040	9.0202
48.33	-12.8839	-3.8622	3.8622	12.8839
69.52	-19.3753	-5.8082	5.8082	19.3753
100	-29.7300	-8.9122	8.9122	29.7300

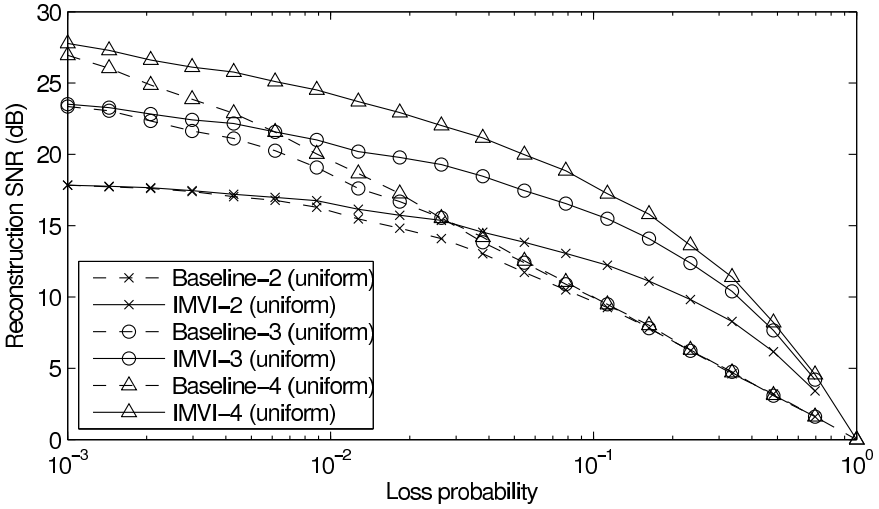


(a) Example I

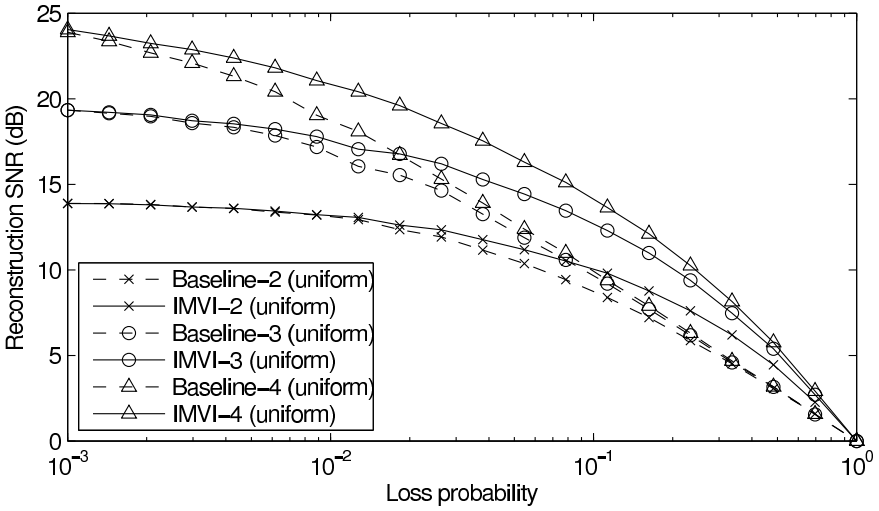


(b) Example II

Figure A.3: Examples of coding: (a) a third-order AR source, (b) a fifth-order AR source. Decoded signal SNRs are plotted against i.i.d. channel loss probability for Baseline and IMVI with Lloyd-Max quantization at 2-4 bits/sample.



(a) Example III



(b) Example IV

Figure A.4: Examples of coding: (a) a ninth-order AR source, (b) a tenth-order AR source. Decoded signal SNRs are plotted against i.i.d. channel loss probability for Baseline and IMVI with uniform quantization at 2-4 bits/sample.

Table A.7: Maximum observed SNR improvement in Examples I-IV.

Quantization rate [bits/sample]	2	3	4
Example I			
Loss prob.	23.4%	16.2%	11.3%
SNR improvement [dB]	3.2	7.1	9.7
Example II			
Loss prob.	33.6%	16.2%	11.3%
SNR improvement [dB]	3.2	5.4	6.6
Example III			
Loss prob.	33.6%	16.2%	16.2%
SNR improvement [dB]	3.7	6.3	7.8
Example IV			
Loss prob.	23.3%	16.2%	11.3%
SNR improvement [dB]	1.7	3.3	4.2

tions 2.1 and 2.2. The method consistently improves performance for different AR source signal models and quantizers.

4 Concluding Remarks

We have presented a novel method for optimization of predictive quantization of AR signals for transmission over channels with sample erasures. An important contribution of the presented method is a coding framework "design philosophy" that considers the encoding a process that produces noisy measurements of the source signal, in Kalman estimation's understanding of the term. The decoding is viewed as optimal estimation of the source signal based on these measurements.

The proposed method, IMVI, provides offline design of the encoder and decoder for optimal estimation by a Kalman filter at the decoder. By taking channel erasures into account in minimizing the trace of the Kalman state error covariance, we have obtained a design method that allows selection of encoder and decoder parameters which improve robustness to losses and provides MMSE estimation given the actual channel losses at the decoder.

As mentioned in the introduction, earlier applications of Kalman filtering in source coding, [15–19], have employed Kalman filtering at both the encoder and decoder and have not specifically considered transmission loss. Our method employs a Kalman filter at the decoder. The encoder relies on fixed FIR filters. This has the advantage of keeping the encoder simple, which could be a simple sensor node limited in power consumption and/or computation power, while providing MMSE estimation at the de-

coder, which could be a centralized controller or monitoring node without such restrictions in power consumption or lacking computational power.

In this paper, IMVI is demonstrated by application to a generalized DPCM encoder structure in which the filtering of quantized prediction errors has been split into separate prediction error and quantization noise feedback parts. This offers a higher degree of freedom in encoder design than an encoder more along the lines of classic DPCM with only a single filter. This higher degree of freedom may provide additional gains over single-filter encoders.

IMVI is limited to AR source signal models in the current framework. We believe it is feasible to extend the current model to more general ARMA source signal models, making the framework more versatile. This is a topic of future investigation.

IMVI has been demonstrated to improve decoded signal SNR substantially under sample erasure conditions for a diverse selection of source signal models. Furthermore, the improvements are demonstrated consistently for several different model orders and quantization parameters.

References

[1] T. Arildsen et al. „On Predictive Coding for Erasure Channels Using a Kalman Framework“. In: *IEEE Transactions on Signal Processing* 57.11 (Nov. 2009), pp. 4456–4466. ISSN: 1053-587X. DOI: 10.1109/TSP.2009.2025796.

[2] M. Y. Kim and W. B. Kleijn. „Comparative rate-distortion performance of multiple description coding for real-time audiovisual communication over the Internet“. In: *IEEE Transactions on Communications* 54.4 (Apr. 2006), pp. 625–636. ISSN: 0090-6778. DOI: 10.1109/TCOMM.2006.873071.

[3] C. A. Rødbro et al. „Hidden Markov model-based packet loss concealment for voice over IP“. In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.5 (Sept. 2006), pp. 1609–1623. ISSN: 1558-7916. DOI: 10.1109/TSA.2005.858561.

[4] D. Persson, T. Eriksson, and P. Hedelin. „Packet Video Error Concealment With Gaussian Mixture Models“. In: *IEEE Transactions on Image Processing* 17.2 (Feb. 2008), pp. 145–154. ISSN: 1057-7149. DOI: 10.1109/TIP.2007.914151.

[5] *7 kHz Audio-Coding within 64 kbit/s*. Recommendation G.722. International Telecommunication Union, Nov. 1988.

[6] *Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP)*. Recommendation G.729. International Telecommunication Union, Jan. 2007.

[7] *AMR Speech Codec*. Technical Specification TS 126 071. Version 7.0.1. European Telecommunications Standards Institute, July 2007.

[8] *Adaptive Multi-Rate - Wideband (AMR-WB) Speech Codec*. Technical Specification TS 126 171. Version 7.0.0. European Telecommunications Standards Institute, June 2007.

[9] *Enhanced Variable Rate Codec (EVRC)*. Technical Requirements C.S0014-0. Version 1.0. 3rd Generation Partnership Project 2, Dec. 1999.

[10] *Selectable Mode Vocoder (SMV) Service Option for Wideband Spread Spectrum Communication Systems*. Technical Requirements C.S0030-0. Version 3.0. 3rd Generation Partnership Project 2, Jan. 2004.

[11] *Source-Controlled Variable-Rate Multimode Wideband Speech Codec (VMR-WB)*. Technical Requirements C.S0052-A. Version 1.0. 3rd Generation Partnership Project 2, Apr. 2005.

[12] M. Naraghi-Pour and D. L. Neuhoff. „Mismatched DPCM encoding of autoregressive processes“. In: *IEEE Transactions on Information Theory* 36.2 (1990), pp. 296–304. ISSN: 0018-9448. DOI: 10.1109/18.52476.

[13] O. G. Guleryuz and M. T. Orchard. „On the DPCM compression of Gaussian autoregressive sequences“. In: *IEEE Transactions on Information Theory* 47.3 (2001), pp. 945–956. ISSN: 0018-9448. DOI: 10.1109/18.915650.

[14] N. S. Jayant and P. Noll. *Digital Coding of Waveforms - Principles and Applications to Speech and Video*. Prentice Hall, 1984. ISBN: 0-13-211913-7.

[15] J. Gibson, S. Jones, and J. Melsa. „Sequentially Adaptive Prediction and Coding of Speech Signals“. In: *IEEE Transactions on Communications* 22.11 (1974), pp. 1789–1797. ISSN: 0096-2244.

[16] S. Crisafulli, J. D. Mills, and R. R. Bitmead. „Kalman filtering techniques in speech coding“. In: *1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. 1992, 77–80 vol.1. DOI: 10.1109/ICASSP.1992.225968.

[17] T. V. Ramabadran and D. Sinha. „Speech data compression through sparse coding of innovations“. In: *IEEE Transactions on Speech and Audio Processing* 2.2 (1994), pp. 274–284. ISSN: 1063-6676. DOI: 10.1109/89.279276.

[18] S. V. Andersen, S. H. Jensen, and E. Hansen. „Quantization noise modeling in low-delay speech coding“. In: *Proc. IEEE Workshop on Speech Coding For Telecommunications*. 1997, pp. 65–66. DOI: 10.1109/SCFT.1997.623898.

[19] S. V. Andersen. „Quantization Noise Modeling in Predictive Speech Coding“. PhD thesis. Aalborg University, 1998.

[20] M. H. Chan. „The performance of DPCM operating on lossy channels with memory“. In: *IEEE Transactions on Communication* 43.234 (Feb-Mar-Apr 1995), pp. 1686–1696. DOI: 10.1109/26.380240.

[21] T. V. Ramabadran and D. Sinha. „On the selection of measurements in least-squares estimation“. In: *IEEE International Conference on Systems Engineering*. 1989, pp. 221–226. DOI: 10.1109/ICSYSE.1989.48659.

[22] B. Sinopoli et al. „Kalman filtering with intermittent observations“. In: *IEEE Transactions on Automatic Control* 49.9 (2004), pp. 1453–1464. ISSN: 0018-9286. DOI: 10.1109/TAC.2004.834121.

[23] L. Schenato et al. „Foundations of Control and Estimation Over Lossy Networks“. In: *Proceedings of the IEEE* 95.1 (2007), pp. 163–187. ISSN: 0018-9219. DOI: 10.1109/JPROC.2006.887306.

[24] A. K. Fletcher, S. Rangan, and V. K. Goyal. „Estimation from lossy sensor data: jump linear modeling and Kalman filtering“. In: *Proceedings of the third international symposium on information processing in sensor networks*. Berkeley, California, USA: ACM Press, 2004, pp. 251–258. DOI: 10.1145/984622.984659.

[25] A. K. Fletcher et al. „Robust Predictive Quantization: Analysis and Design Via Convex Optimization“. In: *IEEE Journal of Selected Topics in Signal Processing* 1.4 (Dec. 2007), pp. 618–632. ISSN: 1932-4553. DOI: 10.1109/JSTSP.2007.910622.

[26] S. Tewksbury and R. Hallock. „Oversampled, linear predictive and noise-shaping coders of order N $>$ 1“. In: *IEEE Transactions on Circuits and Systems* 25.7 (July 1978), pp. 436–447. ISSN: 0098-4094.

[27] B. Atal. „Predictive Coding of Speech at Low Bit Rates“. In: *IEEE Transactions on Communications* 30.4 (1982), pp. 600–614. ISSN: 0096-2244.

[28] W. R. Bennett. „Spectra of Quantized Signals“. In: *Bell Systems Technical Journal* 27 (1948), pp. 446–472.

[29] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[30] W.-R. Wu and A. Kundu. „Recursive filtering with non-Gaussian noises“. In: *IEEE Transactions on Signal Processing* 44.6 (1996), pp. 1454–1468. ISSN: 1053-587X. DOI: 10.1109/78.506611.

[31] C. Masreliez. „Approximate non-Gaussian filtering with linear state and observation relations“. In: *IEEE Transactions on Automatic Control* 20.1 (1975), pp. 107–110. ISSN: 0018-9286.

[32] W. Niehsen. „Robust Kalman filtering with generalized Gaussian measurement noise“. In: *IEEE Transactions on Aerospace and Electronic Systems* 38.4 (Oct. 2002), pp. 1409–1412. ISSN: 0. DOI: 10.1109/TAES.2002.1145765.

[33] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1979, repr. Minneola, New York: Dover Publications Inc., 2005. ISBN: 0-486-43938-0.

[34] X. Liu and A. Goldsmith. „Kalman filtering with partial observation losses“. In: *43rd IEEE Conference on Decision and Control*. Vol. 4. 2004, 4180–4186 Vol.4.

[35] N. Farvardin and J. Modestino. „Rate-distortion performance of DPCM schemes for autoregressive sources“. In: *IEEE Transactions on Information Theory* 31.3 (1985), pp. 402–418. ISSN: 0018-9448.

[36] S. Lloyd. „Least squares quantization in PCM“. In: *IEEE Transactions on Information Theory* 28.2 (1982). The material in this paper was presented in part at the Institute of Mathematical Statistics Meeting, Atlantic City, NJ, September 10-13, 1957., pp. 129–137. ISSN: 0018-9448.

[37] J. Max. „Quantizing for minimum distortion“. In: *IEEE Transactions on Information Theory* 6.1 (1960), pp. 7–12. ISSN: 0018-9448.

[38] J. Bucklew and J. Gallagher N. „Some properties of uniform step size quantizers (Corresp.)“ In: *IEEE Transactions on Information Theory* 26.5 (1980), pp. 610–613. ISSN: 0018-9448.

[39] T. Arildsen. http://es.aau.dk/staff/tha.

[40] J. S. Garofolo. *DARPA TIMIT Acoustic-Phonetic Speech Database Training Set, CD-ROM Prototype Distribution.*

Publication B

On Predictive Coding for Erasure Channels Using a Kalman Framework

Thomas Arildsen, Manohar N. Murthi, Søren Vang Andersen, and Søren Holdt Jensen

This paper was originally published as:
T. Arildsen et al. „On Predictive Coding for Erasure Channels Using a Kalman Framework“. In: *Proceedings of the 17th European Signal Processing Conference (EUSIPCO-2009)*. Eurasip. EUSIPCO 2009, Glasgow Ltd, Aug. 2009.

First published in the Proceedings of the 17th European Signal Processing Conference (EUSIPCO-2009) in 2009, published by EURASIP.

The current layout and citation numbering has been revised compared to the published version.

Abstract

We present a new design method for robust low-delay coding of AR sources for transmission across erasure channels. The method is based on LPC with Kalman estimation at the decoder. The method designs the encoder and decoder offline through an iterative algorithm based on minimization of the trace of the decoder state error covariance. The design method applies to stationary AR sources of any order. Simulation results show considerable performance gains, when the transmitted quantized prediction errors are subject to loss, in terms of SNR compared to the same coding framework optimized for no loss. We furthermore investigate the impact on decoding performance when channel losses are correlated. We find that the method still provides substantial improvements in this case despite being designed for i.i.d. losses.

1 Introduction

In transmission of real-time signals data losses are typically an unavoidable impairment. Transmission can be protected against losses by, e.g., error correcting codes or multiple description coding (MDC), or the effects of losses on the transmitted signal may be mitigated through various loss concealment techniques at the receiver. For low-delay coding applications, error-correcting codes are impractical due to the delay they impose. In such cases, another possibility is to modify the source coding itself to increase robustness against losses.

Linear predictive coding (LPC) is a principle commonly employed in speech applications. Differential pulse code modulation (DPCM) is an example of a predictive source coding scheme [1]. Kalman filtering can be applied in predictive coding to provide minimum mean squared error (MMSE) estimation of the source signal. Previous applications of Kalman filtering to predictive coding employ Kalman filters at both encoder and decoder and transmit quantized Kalman innovations from encoder to decoder, requiring synchronized encoders and decoders [2–5].

When considering a Kalman filter-based decoder, the work in [4] applies to optimizing the Kalman filter for given noise statistics by selecting the optimal measurement vector that minimizes some measure on the a posteriori state error covariance. However, this approach does not take channel losses into account. Handling of lost measurements in a Kalman estimator is investigated thoroughly in [6, 7], but this work does not consider optimization of the coding system for such losses.

An approach for optimization of a predictive quantization scheme employing Kalman-like filters at encoder and decoder is presented in [8] where channel losses are modeled by a Markov model. [8] is contemporaneous

work with a different philosophy than what we present here; it presents an optimization method based on jump linear system (JLS) modeling and linear matrix inequality (LMI)-constrained convex optimization to design fixed gains for the encoder and decoder filters for each channel state.

In [9], we present a novel optimization method for the design of low-delay predictive coding systems, demonstrating a method for designing a robust encoder and decoder for given channel loss statistics. In particular, we examine DPCM, which is a canonical method of predictive coding which captures the basic problems of real-time transmission over channels with packet loss. In contrast to other efforts to design robust DPCM methods (e.g., [8, 10]), we consider a generalized DPCM encoder structure with separate prediction and noise feedback filters, an encoding structure commonly employed in speech coding. Moreover, we consider the case where these encoder filters are fixed time-invariant filters, leading to low-complexity quantization of signal samples. This encoder transmits quantization information (related to quantized prediction errors) that is subject to channel loss/erasure. The decoder views the received information from the encoder as noisy signal measurements, and utilizes Kalman filtering principles to perform linear minimum mean-squared error (LMMSE) estimation of the signal. This approach of viewing the encoder as producing noisy measurements is in contrast to previous approaches in [2–5], in which the encoder's transmitted quantized prediction error is viewed as the innovation, with both the encoder and decoder running synchronized Kalman filters.

Our predictive coding scheme consists of both offline and online stages. In the offline stage, the fixed encoder filters, and the corresponding Kalman measurement vector at the decoder are jointly designed, taking into account both the quantization noise and channel loss statistics. In the online operation, the decoder's time-varying Kalman filter estimates each signal sample, taking into account the individual loss events in the estimation. Since the encoder remains fixed while the decoder is time-varying, synchronization between encoder and decoder is not assumed. Simulation results in [9] demonstrate the efficacy of the proposed method.

In this paper we present further results investigating the performance of the proposed design method. In addition to the i.i.d. losses considered in [9], we investigate performance under Gilbert-Elliot correlated losses to assess how the method handles under more demanding loss conditions than it was intended for. We show that although overall performance of the coding framework is degraded by correlated erasures, compared to i.i.d erasures, our design method still provides significant improvements compared to the same coding framework optimized for no loss.

To illustrate the application of our design method, we employ a coding framework based on generalized DPCM coding. We chose DPCM as this captures the essence of low delay predictive coding. The principle can be extended to more complex coding schemes as well, for example based on

vector quantization of LSFs (line spectral frequencies) or other methods.

2 Coding Framework and Design Method

This section describes the source encoder and decoder in Sections 2.1 and 2.2. The optimization for sample losses is treated in Section 2.3.

2.1 Source Encoder

The encoder chosen to illustrate the application of our design method is based on generalized DPCM coding. We consider an encoder with a noise

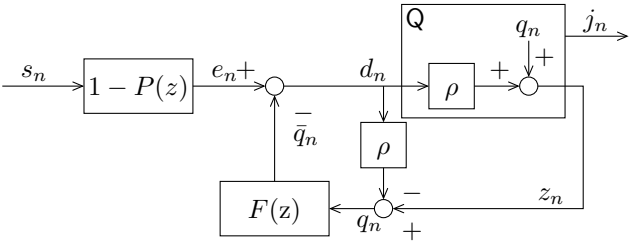


Figure B.1: Generalized DPCM source encoder with additive white noise (AWN) quantizer model and de-correlated quantization noise feedback.

shaping structure illustrated in Figure B.1, [11]. The source signal and the encoder are defined by (B.1)–(B.4).

$$s_n = \sum_{i=1}^{N} \alpha_i s_{n-i} + r_n, \tag{B.1}$$

$$d_n = e_n - \bar{q}_n, \tag{B.2}$$

$$e_n = s_n - \sum_{i=1}^{p} a_i s_{n-i}, \tag{B.3}$$

$$\bar{q}_n = \sum_{i=1}^{f} b_i q_{n-i} \tag{B.4}$$

The source process of order N is driven by zero-mean stationary white Gaussian noise r; α_i are the source auto-regressive (AR) coefficients, defining the source process together with N; e_n is the prediction error; $\bar{q}_n$ is the filtered quantization noise feedback; d_n is the input to the quantizer; q_n is the additive quantization error; p is the predictor order and $a_i,\ i = 1, \ldots, p$ are its coefficients; f is the noise feedback filter order and $b_i,\ i = 1, \ldots, f$ are

its coefficients. Please note that the encoder definition allows $p \neq f \neq N$ in general.

As depicted in Figure B.1, the output transmitted to the decoder is quantization indices, j_n, for the quantized prediction error, z_n, in (B.5), seen as the Kalman measurement by the decoder, cf. Section 2.2.

$$z_n = Q(d_n) \tag{B.5}$$

We use scalar quantization $Q(\cdot)$ with a gain-plus-additive-noise model [1]. The model accommodates correlation between quantizer input and quantization noise.

$$z_n = \rho d_n + q_n, \tag{B.6}$$

where $\rho \in [0, 1]$, and q_n is a stationary zero-mean white Gaussian noise, independent of d_n, with variance

$$\sigma_q^2 = k \operatorname{var}\{d_n\}, \tag{B.7}$$

where the quantization noise is modeled with a variance proportional, by a constant k, to the variance of the input to the quantizer. The assumption of Gaussian q_n is a simplifying assumption in the sense that quantization noise is generally not Gaussian and only approximately white under high-rate assumptions [12]. In order to be able to model the quantization noise as measurement noise in the Kalman filter in the decoder and to facilitate quantizer design, this noise must be white Gaussian. Since q_n is not truly Gaussian, the decoder Kalman filter will not provide MMSE estimation of the source signal. The Kalman filter will however be the best linear estimator of the source signal–LMMSE-optimal. ρ and k are given by the coding loss, β, of the quantizer:

$$\rho = 1 - \beta \qquad\qquad k = \beta(1 - \beta), \tag{B.8}$$

where β is the inverse of the quantizer coding gain [12].

The noise incurred by quantization is

$$\tilde{q}_n = z_n - d_n = (\rho - 1)d_n + q_n. \tag{B.9}$$

In order to simplify the calculation of quantizer input variance in the optimization of the encoder, we wish to feed back a white noise component. Therefore, d_n is scaled by ρ in the quantization noise feedback to de-correlate the noise feedback from d_n. Thus, we only feed back the uncorrelated part of the quantization noise $q_n = z_n - \rho d_n$. This allows us to model the input to the quantizer, d_n, as Gaussian which simplifies the optimization of the encoder and the quantizer design. See [9] for design of quantizers used in the encoder.

2.2 Kalman Filter-Based Decoder

The decoder is based on Kalman filtering, i.e., LMMSE estimation of the source signal s. The Kalman filter at the decoder estimates the source signal based on measurements, z, reconstructed from the received quantization indices, j_n. In order to derive the Kalman estimator $\hat{s}_n$ of s_n, the source process and encoder equations are modeled by a state space model of the form given in, e.g., [13]. The state transition equation is chosen to represent the evolution of the source signal s_n as well as the states of the encoder filters P and F. The measurement equation represents the filtering and quantization operations of the encoder.

The decoder is based on the state-space model in (B.10) and (B.11).

$$\mathbf{x}_{n+1} = \mathbf{F}\mathbf{x}_n + \mathbf{G}\mathbf{w}_n \tag{B.10}$$

$$z_n = \mathbf{h}^{\mathrm{T}}\mathbf{x}_n + q_n, \tag{B.11}$$

where the state $\mathbf{x}_n$ is defined as

$$\mathbf{x}_n = \left[\begin{array}{c|ccc|ccc} s_n & s_{n-1} & \cdots & s_{n-p} & q_{n-1} & \cdots & q_{n-f} \end{array}\right]^{\mathrm{T}}, \tag{B.12}$$

and the state transition matrix $\mathbf{F}$ is defined as

$$\mathbf{F} = \left[\begin{array}{c|c|c} \alpha_{1\times N} & \mathbf{0}_{1\times(p+1-N)} & \mathbf{0}_{(p+1)\times f} \\ \hline \mathbf{I}_p & \mathbf{0}_{p\times 1} & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{f\times(p+1)}} & \begin{array}{c|c} \multicolumn{2}{c}{\mathbf{0}_{1\times f}} \\ \hline \mathbf{I}_{(f-1)} & \mathbf{0}_{(f-1)\times 1} \end{array} \end{array}\right], \tag{B.13}$$

where $\alpha = [\alpha_1 \cdots \alpha_N]$ are the coefficients of the source AR process, $\mathbf{I}_x$ are $x \times x$ identity matrices, and $\mathbf{0}$ are all-zero matrices (the subscripts in (B.13) denote the dimensions of the individual components). Thus, the top-left part of $\mathbf{F}$ represents the AR filtering of the process noise, given by (B.1), generating the source signal, and shifts past source signal samples through the state. The bottom-right part of $\mathbf{F}$ simply shifts previous quantization noise samples through the state. Note that this formulation allows $p \geq N-1$ and $f \geq 0$.

The process noise $\mathbf{w}_n$ is defined as

$$\mathbf{w}_n = \left[\begin{array}{cc} r_n & q_n \end{array}\right]^{\mathrm{T}}, \tag{B.14}$$

which is stationary zero-mean white Gaussian with

$$\mathbf{Q} = \mathrm{cov}\left\{\mathbf{w}_n, \mathbf{w}_n\right\} = \left[\begin{array}{cc} \mathrm{var}\left\{r_n\right\} & 0 \\ 0 & \mathrm{var}\left\{q_n\right\} \end{array}\right] \tag{B.15}$$

The first (scalar) component of the process noise, r_n, models the source signal excitation and the second (scalar) component, q_n, models the quantization noise fed back to the filter F. The definition (B.14) introduces

correlation between the process noise $\mathbf{w}_n$ and the measurement noise q_n in (B.11) as follows

$$\mathbf{S} = \mathrm{cov}\left\{\mathbf{w}_n, q_n\right\} = \mathrm{E}\left\{\begin{bmatrix} r_n \\ q_n \end{bmatrix} q_n\right\} = \begin{bmatrix} 0 \\ \mathbf{R} \end{bmatrix}, \tag{B.16}$$

where $\mathbf{R} = \sigma_q^2$ is a scalar since we consider scalar-valued measurements z_n. We use a formulation of the Kalman filter which takes the covariance $\mathbf{S}$ into account.

The matrix $\mathbf{G}$ is a transform to allow the process noise $\mathbf{w}_n$ to be defined in a compact form. $\mathbf{G}$ is given by (B.17).

$$\mathbf{G} = \begin{bmatrix} 1 & \mathbf{0}_{2\times p} & 0 & \mathbf{0}_{2\times(f-1)} \\ 0 & & 1 & \end{bmatrix}^{\mathrm{T}} \tag{B.17}$$

The measurement vector $\mathbf{h}$ represents the filtering operations of the encoder as well as the scaling in the model of the quantizer

$$\mathbf{h} = \rho\tilde{\mathbf{h}}, \tag{B.18}$$

where $\tilde{\mathbf{h}}$ contains the coefficients of the prediction error and noise feedback filters

$$\tilde{\mathbf{h}} = \begin{bmatrix} 1 & -a_1 & \cdots & -a_p & -b_1 & \cdots & -b_f \end{bmatrix}^{\mathrm{T}}, \tag{B.19}$$

such that by (B.2),

$$d_n = \tilde{\mathbf{h}}^{\mathrm{T}}\mathbf{x}_n, \tag{B.20}$$

whereby (B.11) follows from (B.6). To summarize, $\tilde{\mathbf{h}}$ represents the filtering in the encoder before quantization. Due to the quantization noise model presented in Section 2.1, $\mathbf{h}$ represents the filtering after quantization and produces the measurement seen by the decoder.

The decoder receives information (quantization indices j_n) to build measurements z_n from the encoder.

Considering the situation where z_n may be lost, we have a time-varying Kalman filter. The measurement vector $\mathbf{h}_n$ and measurement noise covariance $\mathbf{R}_n$ are time-varying. This models the possible loss of measurements at the decoder.

$$\mathbf{h}_n = \gamma_n\mathbf{h} \tag{B.21}$$

$$\mathbf{R}_n = \gamma_n\mathbf{R} + (1-\gamma_n)\,\sigma^2\mathbf{I}, \tag{B.22}$$

where γ_n are outcomes of a random stationary Bernoulli process modeling measurement arrival with arrival probability $\Pr\{\gamma_n = 1\} = \bar{\gamma}$ and loss probability $\Pr\{\gamma_n = 0\} = 1 - \bar{\gamma}$. $\mathbf{R}$ is the measurement noise covariance in the case of no loss and $\sigma^2\mathbf{I}$ is the measurement noise covariance in the case of loss. We let $\sigma^2 \to \infty$ in (B.22), representing infinite uncertainty about

the measurement z_n when it is lost in transmission. See [6] for an example of this approach. Replacing $\mathbf{R}_n$ and $\mathbf{h}_n$ by (B.21) and (B.22) in the Kalman filter, as found in, e.g., [13], and taking $\lim_{\sigma\to\infty}$, we obtain

$$\hat{\mathbf{x}}_n = \hat{\mathbf{x}}_n^- + \gamma_n \mathbf{P}_n^- \mathbf{h} \left(\mathbf{h}^{\mathrm{T}} \mathbf{P}_n^- \mathbf{h} + \mathbf{R}\right)^{-1} \left(z_n - \mathbf{h}^{\mathrm{T}} \hat{\mathbf{x}}_n^-\right) \tag{B.23}$$

$$\mathbf{P}_n = \mathbf{P}_n^- - \gamma_n \mathbf{P}_n^- \mathbf{h} \left(\mathbf{h}^{\mathrm{T}} \mathbf{P}_n^- \mathbf{h} + \mathbf{R}\right)^{-1} \mathbf{h}^{\mathrm{T}} \mathbf{P}_n^- \tag{B.24}$$

$$\hat{\mathbf{x}}_{n+1}^- = \left(\mathbf{F} - \gamma_n \mathbf{G}\mathbf{S}\mathbf{R}^{-1}\mathbf{h}^{\mathrm{T}}\right) \hat{\mathbf{x}}_n + \gamma_n \mathbf{G}\mathbf{S}\mathbf{R}^{-1} z_n \tag{B.25}$$

$$\begin{aligned}\mathbf{P}_{n+1}^- = &\left(\mathbf{F} - \gamma_n \mathbf{G}\mathbf{S}\mathbf{R}^{-1}\mathbf{h}^{\mathrm{T}}\right) \mathbf{P}_n \left(\mathbf{F} - \gamma_n \mathbf{G}\mathbf{S}\mathbf{R}^{-1}\mathbf{h}^{\mathrm{T}}\right)^{\mathrm{T}} \\ &+ \mathbf{G} \left(\mathbf{Q} - \gamma_n \mathbf{S}\mathbf{R}^{-1}\mathbf{S}^{\mathrm{T}}\right) \mathbf{G}^{\mathrm{T}},\end{aligned} \tag{B.26}$$

The decoded signal $\hat{s}_n$ is the first element of $\hat{\mathbf{x}}_n$ according to (B.12). We see that the encoder takes the individual channel erasures/losses into account, represented by $\gamma_n = 0$. This effectively reduces the current estimate of the state $\hat{\mathbf{x}}_n$ to the a priori estimate $\hat{\mathbf{x}}_n^-$ in case of erasure of j_n, i.e., a prediction from past observations.

2.3 Encoder and Decoder Design

One could choose to calculate filter parameters/measurement vectors $\mathbf{h}_n^*$ at each time step n to improve decoding performance at the next time step $n+1$. However, this approach would require loss-less feedback of observed arrivals γ_n from decoder to encoder before time $n+1$ in order to be able to calculate identical parameters at encoder and decoder. In order not to impose this loss-less near-instantaneous feedback requirement on the system, we seek a method that allows offline calculation of *a constant* $\mathbf{h}^*$, *given the statistics of loss*. So, the goal is a method that improves decoding performance under average loss conditions rather than the specific loss outcomes. The offline calculation of $\mathbf{h}^*$ furthermore decreases the computational complexity of the framework since the calculation can be performed once in stead of being performed at each time step n. In contrast to Kalman filtering without loss, $\mathbf{P}_n$ is stochastic here due to measurement losses (the random variable γ_n). We propose the following offline method for designing measurement vectors for improved performance under sample losses. We use a simplified approach where the philosophy is to obtain a $\mathbf{h}^*$ minimizing the expectation of $\mathbf{P}_n$ over current loss, γ_n, at time n.

$$\mathbf{h}^* = \arg\min_{\mathbf{h}} \operatorname{Tr}\left[\mathrm{E}_{\gamma_n}\left\{\mathbf{P}_n\right\}\right], \tag{B.27}$$

where $\mathbf{P}_n$ is given in (B.24)

Similar to [4], we express the measurement noise covariance–or equivalently, quantization noise variance–$\mathbf{R}$ as a function of $\mathbf{h}$ as given in Proposition 2.1.

Proposition 2.1.

$$\mathbf{R}_{(i)} = \frac{k}{\rho^2}\mathbf{h}_{(i)}^{\mathrm{T}}\mathbf{R}_{\mathbf{xx},(i)}\mathbf{h}_{(i)}, \tag{B.28}$$

where $\mathbf{R}_{\mathbf{xx},(i)}$ is the state correlation matrix, with the structure

$$\mathbf{R}_{\mathbf{xx},(i)} = \begin{bmatrix} \mathbf{R}_{ss} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_f\mathbf{R}_{(i-1)} \end{bmatrix}. \tag{B.29}$$

Proof. We refer to [9] for further details. □

The index (i) in Proposition 2.1 and following equations denotes the quantity calculated at iteration i. Using Proposition 2.1 and taking the expectation of (B.24) gives

$$\mathrm{E}_\gamma\left\{\mathbf{P}_{(i)}\right\} = \mathrm{E}_\gamma\left\{\mathbf{P}_{(i)}^-\right\} - \bar{\gamma}\frac{\mathrm{E}_\gamma\left\{\mathbf{P}_{(i)}^-\right\}\mathbf{h}\mathbf{h}^{\mathrm{T}}\,\mathrm{E}_\gamma\left\{\mathbf{P}_{(i)}^-\right\}}{\mathbf{h}^{\mathrm{T}}\left(\mathrm{E}_\gamma\left\{\mathbf{P}_{(i)}^-\right\} + \frac{k}{\rho^2}\mathbf{R}_{\mathbf{xx},(i)}\right)\mathbf{h}}, \tag{B.30}$$

where $\bar{\gamma}$ is the arrival probability. The inverse in (B.24) corresponds to scalar division, because we have scalar measurements. Time index n has been omitted, the index (i) indicates iteration number.

$\mathbf{P}_n^-$ is updated according to the discrete-time Riccati equation, [13, p. 108], of the decoder Kalman filter, adapted for measurement losses, the expectation of which is given in Proposition 2.2.

Proposition 2.2.

$$\begin{aligned}\mathrm{E}_\gamma\left\{\mathbf{P}_{(i)}^-\right\} = {}& \mathbf{F}\,\mathrm{E}_\gamma\left\{\mathbf{P}_{(i-1)}^-\right\}\mathbf{F}^{\mathrm{T}} \\ & - \bar{\gamma}\frac{\left(\mathbf{F}\,\mathrm{E}_\gamma\left\{\mathbf{P}_{(i-1)}^-\right\}\mathbf{h} + \mathbf{G}\mathbf{S}\right)\left(\mathbf{F}\,\mathrm{E}_\gamma\left\{\mathbf{P}_{(i-1)}^-\right\}\mathbf{h} + \mathbf{G}\mathbf{S}\right)^{\mathrm{T}}}{\mathbf{h}^{\mathrm{T}}\left(\mathrm{E}_\gamma\left\{\mathbf{P}_{(i-1)}^-\right\} + \frac{k}{\rho^2}\mathbf{R}_{\mathbf{xx},(i)}\right)\mathbf{h}} \\ & + \mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}}.\end{aligned} \tag{B.31}$$

Proof. We refer to [9] for further details. □

This requires the arrival probability $\bar{\gamma}$ to be known in order to design the encoder and decoder.

At each iteration i, $\mathbf{h}_{(i)}$ is selected to minimize the trace of (B.30) according to (B.32).

$$\mathbf{h}_{(i)} = \arg\max_{\mathbf{h}} \frac{\mathbf{h}^{\mathrm{T}}\left(\mathrm{E}_\gamma\left\{\mathbf{P}_{(i)}^-\right\}\right)^2\mathbf{h}}{\mathbf{h}^{\mathrm{T}}\left(\mathrm{E}_\gamma\left\{\mathbf{P}_{(i)}^-\right\} + \frac{k}{\rho^2}\mathbf{R}_{\mathbf{xx},(i)}\right)\mathbf{h}}, \tag{B.32}$$

Algorithm 1 Design algorithm for lossy transmission.

$\mathbf{h}_{(0)}$: $a_l = b_l = \alpha_l,\ l \leq N, \qquad a_l = 0,\ N < l \leq p, \qquad b_l = 0,\ N < l \leq f$
Initialize $\mathrm{E}_\gamma\{\mathbf{P}^-_{(1)}\}$ to unique stabilizing solution to (B.26) for $\mathbf{h} = \mathbf{h}_0$ and $\gamma_n = 1$.
Set ϵ to desired precision; $i = 0$; stop difference $= \infty$
while stop difference $> \epsilon$ **do**
  Set $i = i + 1$
  Minimize $\mathrm{E}_\gamma\{\mathbf{P}_{(i)}\}$ by (B.33)
  Calculate $\mathbf{h}^*_{(i)}$ by (B.34)
  Calculate $\mathrm{E}_\gamma\{\mathbf{P}^-_{(i+1)}\}$ by (B.31)
  Set stop difference $= \left|\mathrm{Tr}\left[\mathrm{E}_\gamma\{\mathbf{P}^-_{(i+1)}\} - \mathrm{E}_\gamma\{\mathbf{P}^-_{(i)}\}\right]\right|$
end while
Select $\mathbf{h}^*$ as $\mathbf{h}^*_{(i)}$

in which $(\mathrm{E}_\gamma\{\mathbf{P}^-_{(i)}\})^2$ is short for $\mathrm{E}_\gamma\{\mathbf{P}^-_{(i)}\}^{\mathrm{T}}\,\mathrm{E}_\gamma\{\mathbf{P}^-_{(i)}\}$ since $\mathbf{P}^-_{(i)}$ is symmetric.

Equation (B.32) may be rewritten as a Rayleigh quotient through a Cholesky factorization $\mathbf{L}\mathbf{L}^{\mathrm{T}} = \mathrm{E}_\gamma\{\mathbf{P}^-_{(i)}\} + \frac{k}{\rho^2}\mathbf{R}_{\mathbf{xx},(i)}$ where $\mathbf{L}$ is a lower triangular matrix. This allows us to obtain $\mathbf{h}_{(i)}$ as outlined in Proposition 2.3.

Proposition 2.3.

$$\mathbf{y}^*_{(i)} = \arg\max_{\mathbf{y}} \frac{\mathbf{y}^{\mathrm{T}}\mathbf{L}^{-1}\left(\mathrm{E}_\gamma\left\{\mathbf{P}^-_{(i)}\right\}\right)^2\mathbf{L}^{-\mathrm{T}}\mathbf{y}}{\mathbf{y}^{\mathrm{T}}\mathbf{y}}, \tag{B.33}$$

is given as the eigenvector of $\mathbf{L}^{-1}(\mathrm{E}_\gamma\{\mathbf{P}^-_{(i)}\})^2\mathbf{L}^{-\mathrm{T}}$ corresponding to its largest eigenvalue [4]. We obtain the measurement vector from $\mathbf{y}^*_{(i)}$ given by (B.33) with a normalization by the first element of the vector in order to keep $\mathbf{h}_{(i)}$ as formulated in (B.19), with its first element equal to 1.

$$\tilde{\mathbf{h}}_{(i)} = \frac{\mathbf{L}^{-\mathrm{T}}\mathbf{y}^*_{(i)}}{c} \qquad\qquad \mathbf{h}_{(i)} = \rho\tilde{\mathbf{h}}_{(i)}, \tag{B.34}$$

where c is the first element of the vector $\mathbf{L}^{-\mathrm{T}}\mathbf{y}^*_{(i)}$.

Proof. We refer to [9] for further details. □

Equations (B.31), (B.33) and (B.34) are iterated until convergence of (B.31), upon which the resulting $\mathbf{h}_{(i)}$ is chosen as fixed measurement vector $\mathbf{h}^*$ for the decoder given by (B.23)–(B.26) and the corresponding $\tilde{\mathbf{h}}^*$ for the encoder given by the relation (B.18). The design method is summarized in Algorithm 1.

3 Simulations

In [9] we present examples of decoded signal-to-noise ratio (SNR) performance gains achieved by the design presented in Section 2.3. In this paper we present additional simulation results for a source process with a somewhat different power spectral shape than the examples in [9].

- A stationary random source signal was generated from an AR(10) process. The source signal was encoded with encoder and decoder designed for each simulated arrival probability $\bar{\gamma}$. The quantization indices with losses were decoded using the Kalman decoder given by (B.23)–(B.26). This setup is referred to as "Iterative Measurement Vector Improvement (IMVI)".

- As a baseline for comparison, the same source signal was encoded and decoded with encoder and decoder designed for no loss ($\bar{\gamma} = 0$). We shall denote this baseline method "Baseline".

Sample arrivals γ_n were simulated as:

- Independent identically distributed arrivals, modeled as outcomes of a Bernoulli random process over a series of arrival probabilities $\bar{\gamma} \in [0, 1]$ and applied to the transmitted encoder quantization indices j_n. This loss process is referred to as "i.i.d" in the following.

- Outcomes of a Gilbert-Elliot loss process simulated as a two-state Markov process where state 1 corresponds to loss ($\gamma_n = 0$) and state 2 to correct arrival ($\gamma_n = 1$). The state transition probabilities are

$$p_{1,1} = 1 - p_{1,2} = \lambda_1 \tag{B.35}$$
$$p_{2,1} = 1 - p_{2,2} = \lambda_2. \tag{B.36}$$

  The loss model will remain in the loss state for exponentially distributed times with mean error burst length $1/(1-\lambda_1)$. The stationary distribution of the Markov chain is

$$q_1 = 1 - q_2 = \frac{\lambda_2}{1 - \lambda_1 + \lambda_2}, \tag{B.37}$$

  where q_2, the probability of being in the no-loss state, corresponds to the overall arrival probability $\bar{\gamma}$ in the filter design algorithm of Section 2.3. We refer to loss processes of this type as "GE-2" and "GE-3" for mean error burst lengths of 2 and 3, respectively.

Algorithm 1 is designed for i.i.d. losses only, but we include correlated losses in the simulations to assess the algorithm's performance under more

difficult loss conditions. The simulations have been conducted for a Lloyd-Max quantizer at 2 bits/sample.

Test data were generated from an AR model estimated from a 20ms sub-sequence selected from a voice-active region of speech found in [14]. The coefficients used were: $\alpha_1 = 0.8694$, $\alpha_2 = -0.4616$, $\alpha_3 = 0.0186$, $\alpha_4 = 0.4603$, $\alpha_5 = -0.2236$, $\alpha_6 = 0.0302$, $\alpha_7 = 0.1843$, $\alpha_8 = -0.1721$, $\alpha_9 = 0.0239$, $\alpha_{10} = -0.2115$. The power spectrum of the source AR process is plotted in Figure B.2. Decoded signal SNR is compared for IMVI and Baseline in Figure B.3 for a range of loss probabilities.

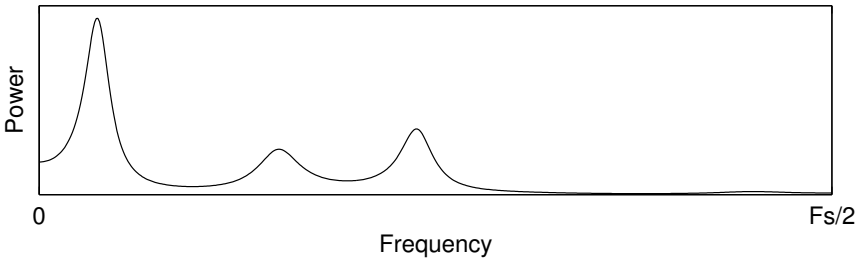


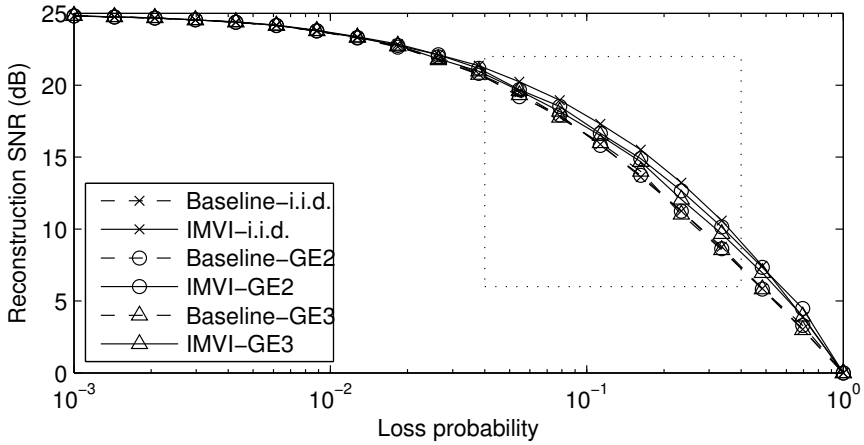
Figure B.2: Power spectral density of 10th order AR source signal.

Figure B.3 first and foremost shows how IMVI is capable of improving decoded signal SNR over the coding framework, Baseline, which has not been optimized for loss. In all three loss scenarios, IMVI improves SNR by, at best, 1.1 dB to 2.0 dB with most notable improvements above 5 % loss rate. The plots show that the improvement obtained by our method is degraded as the losses become correlated; in the i.i.d. loss scenario, we observe improvements of up to 2.0 dB (appr. 23 % loss rate); in the GE-2 scenario, we observe improvements of up to 1.5 dB (appr. 34 % loss rate); in the GE-3 scenario, we observe improvements of up to 1.1 dB (appr. 34 % to 48 % loss rate). This indicates that the design algorithm is still effective at correlated loss scenarios, although not intended for such losses, but performance improvements are more modest in these cases.

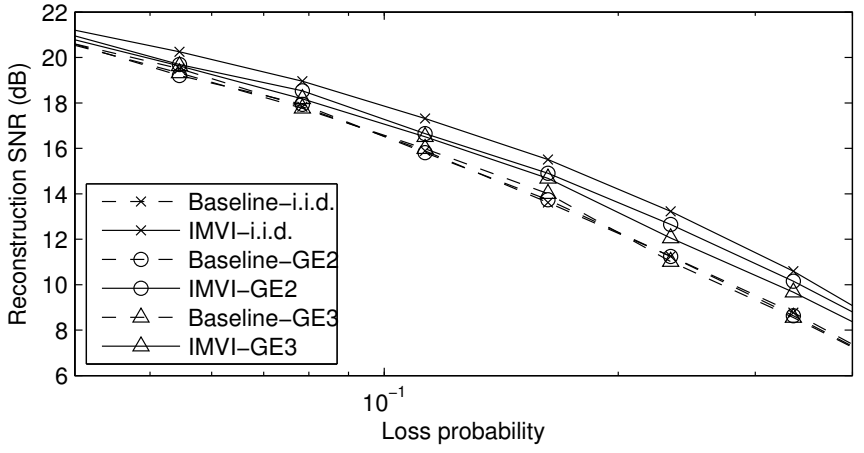
3.1 Parameter Validation

We evaluate the correspondence between the expected state error covariance $\mathrm{E}_{\gamma_n}\{\mathbf{P}_n\}$ used in (B.27) and the empirical average over actual observed $\mathbf{P}_n$ in the running decoder, denoted $\mathrm{avg}(\mathbf{P}_n)$. This is done to validate that the optimization objective used in Algorithm 1 is reasonable.

We investigate $\mathrm{Tr}[\mathbf{P}_n]$ since the diagonal elements are the error variances of the estimated state elements. These relate directly to the SNR of the decoded signal. The resulting $\mathrm{Tr}[\mathrm{E}_{\gamma_n}\{\mathbf{P}_n\}]$ from Algorithm 1 is compared



(a)



(b)

Figure B.3: Decoded signal SNR for 10th order AR source signal. Plot (b) shows the marked segment of plot (a) enlarged.

to $\mathrm{Tr}\left[\mathrm{avg}\left(\mathbf{P}_n\right)\right]$ observed at the decoder and averaged over 10^5 samples. We compared the measured quantities at a loss rate of $1-\bar{\gamma}=0.2336$:

$$\mathrm{Tr}\left[\mathrm{E}_{\gamma_n}\left\{\mathbf{P}_n\right\}\right] = 7.3266 \tag{B.38}$$

$$\mathrm{Tr}\left[\mathrm{avg}\left(\mathbf{P}_n\right)\right]_{\text{i.i.d.}} = 6.7400 \tag{B.39}$$

$$\mathrm{Tr}\left[\mathrm{avg}\left(\mathbf{P}_n\right)\right]_{\text{GE-2}} = 7.5081 \tag{B.40}$$

$$\mathrm{Tr}\left[\mathrm{avg}\left(\mathbf{P}_n\right)\right]_{\text{GE-3}} = 7.9330 \tag{B.41}$$

From (B.38) and (B.39) we observe that Algorithm 1 seems to over-estimate the state error covariance in this example. From (B.38), (B.40) and (B.41) it is observed that Algorithm 1 apparently under-estimates the state error covariance in the case of correlated losses in this example. This could explain the smaller improvement obtained by the algorithm under correlated losses. At the same time it seems that over-estimating the state error covariance (the i.i.d. case) does not have as large an impact on the resulting improvement.

4 Conclusion

We have presented a novel method for optimization of predictive quantization of AR signals for transmission across erasure channels. An important contribution of the presented method is a coding framework "design philosophy" that considers the encoding a process that produces noisy measurements of the source signal. The decoding is viewed as optimal linear estimation of the source signal based on these measurements.

The proposed method provides offline design of the encoder and decoder. By taking channel erasures into account in minimizing the trace of the Kalman state error covariance, we have obtained a design method that allows selection of encoder and decoder parameters which improve robustness to losses in a framework that provides LMMSE estimation given the received measurements at the decoder.

We point out that the presented design method can be applied to predictive coding systems in general and is not limited to the particular framework presented in this paper.

The presented method has been demonstrated to improve decoded signal SNR substantially under sample erasure conditions for a diverse selection of source signal models [9]. In this paper, we present results for another signal model in addition to the examples in [9] and for correlated channel erasures. These results show that the presented method is capable of improving decoded signal quality substantially under loss conditions when losses become correlated, despite the fact that the method is designed for i.i.d. losses.

In the presented example, the design method calculates a larger expected state error covariance than what is observed in the actual decoder; i.e. the

encoder and decoder are designed somewhat conservatively, corresponding to a slightly higher loss rate than the actual rate. This indicates room for additional improvements if this behaviour generalizes to other signal models.

References

[1] N. S. Jayant and P. Noll. *Digital Coding of Waveforms - Principles and Applications to Speech and Video*. Prentice Hall, 1984. ISBN: 0-13-211913-7.

[2] J. Gibson, S. Jones, and J. Melsa. „Sequentially Adaptive Prediction and Coding of Speech Signals“. In: *IEEE Transactions on Communications* 22.11 (1974), pp. 1789–1797. ISSN: 0096-2244.

[3] S. Crisafulli, J. D. Mills, and R. R. Bitmead. „Kalman filtering techniques in speech coding“. In: *1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. 1992, 77–80 vol.1. DOI: 10.1109/ICASSP.1992.225968.

[4] T. V. Ramabadran and D. Sinha. „Speech data compression through sparse coding of innovations“. In: *IEEE Transactions on Speech and Audio Processing* 2.2 (1994), pp. 274–284. ISSN: 1063-6676. DOI: 10.1109/89.279276.

[5] S. V. Andersen, S. H. Jensen, and E. Hansen. „Quantization noise modeling in low-delay speech coding“. In: *Proc. IEEE Workshop on Speech Coding For Telecommunications*. 1997, pp. 65–66. DOI: 10.1109/SCFT.1997.623898.

[6] B. Sinopoli et al. „Kalman filtering with intermittent observations“. In: *IEEE Transactions on Automatic Control* 49.9 (2004), pp. 1453–1464. ISSN: 0018-9286. DOI: 10.1109/TAC.2004.834121.

[7] L. Schenato et al. „Foundations of Control and Estimation Over Lossy Networks“. In: *Proceedings of the IEEE* 95.1 (2007), pp. 163–187. ISSN: 0018-9219. DOI: 10.1109/JPROC.2006.887306.

[8] A. K. Fletcher et al. „Robust Predictive Quantization: Analysis and Design Via Convex Optimization“. In: *IEEE Journal of Selected Topics in Signal Processing* 1.4 (Dec. 2007), pp. 618–632. ISSN: 1932-4553. DOI: 10.1109/JSTSP.2007.910622.

[9] T. Arildsen et al. „On Predictive Coding for Erasure Channels Using a Kalman Framework“. In: *IEEE Transactions on Signal Processing* 57.11 (Nov. 2009), pp. 4456–4466. ISSN: 1053-587X. DOI: 10.1109/TSP.2009.2025796.

[10] M. H. Chan. „The performance of DPCM operating on lossy channels with memory“. In: *IEEE Transactions on Communication* 43.234 (Feb-Mar-Apr 1995), pp. 1686–1696. DOI: 10.1109/26.380240.

[11] B. Atal and M. Schroeder. „Predictive coding of speech signals and subjective error criteria“. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27.3 (1979), pp. 247–254. ISSN: 0096-3518.

[12] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[13] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1979, repr. Minneola, New York: Dover Publications Inc., 2005. ISBN: 0-486-43938-0.

[14] J. S. Garofolo. *DARPA TIMIT Acoustic-Phonetic Speech Database Training Set, CD-ROM Prototype Distribution*.

Publication C

Fixed-Lag Smoothing for Low-Delay Predictive Coding with Noise Shaping for Lossy Networks

Thomas Arildsen, Jan Østergaard, Manohar N. Murthi, Søren Vang Andersen, Søren Holdt Jensen

This paper was originally published as:
T. Arildsen et al. „Fixed-Lag Smoothing for Low-Delay Predictive Coding with Noise Shaping for Lossy Networks“. In: *Proceedings of Data Compression Conference (DCC-2010)*. Los Alamitos, CA: IEEE Computer Society, Mar. 2010, pp. 279–287. DOI: 10.1109/DCC.2010.33.

The current layout has been revised compared to the published version.

Abstract

We consider linear predictive coding and noise shaping for coding and transmission of AR sources over lossy networks. We generalize an existing framework to arbitrary filter orders and propose use of fixed-lag smoothing at the decoder, in order to further reduce the impact of transmission failures. We show that fixed-lag smoothing up to a certain delay can be obtained without additional computational complexity by exploiting the state-space structure. We prove that the proposed smoothing strategy strictly improves performance under quite general conditions. Finally, we provide simulations on AR sources, and channels with correlated losses, and show that substantial improvements are possible.

1 Introduction

In coding of source signals for transmission across lossy networks, it is traditionally the job of a subsequent error eorrecting code (ECC) to ensure robustness against transmission losses. In [1] we have presented a method for design of linear predictive coding (LPC) with noise shaping optimized for transmission losses, which does not rely on ECCs in order to achieve the desired degree of robustness towards losses. The coding problem is formulated and solved as a state estimation problem and the coding performance is improved through optimization for the known transmission loss statistics over part of a state-space model representing the source and the encoder.

Kalman estimation is employed for state estimation at the decoder. The technique for handling lost measurements has been described in [2, 3]. However, [2, 3] do not consider the optimization of a coding system for such losses. Optimization of a LPC system for transmission losses has been considered in [4, 5]. The approach in [4, 5] is, however, quite different from the approach presented in this paper and the decoding is based on fixed filters alternating between loss and non-loss states and are as such not linear minimum mean-squared error (LMMSE)-optimal estimators. [4, 5] work with a flexible Markov loss model. An optimal estimator for such Markov jump linear systems is found in [6]. The estimator type employed in [4, 5] is more akin to the approach described in [7].

In this paper, we propose to use fixed-lag smoothing at the decoder, in order to reduce the impact of transmission failures, at the expense of a small decoder delay. We show that if the desired smoothing lag is less than or equal to the order of the predictor, fixed-lag smoothing can be obtained without additional computational complexity. Moreover, under quite general conditions, we prove that our proposed smoothing strategy strictly improves performance. Finally, we provide simulations on higher order auto-regressive (AR) sources, and channels with correlated losses,

and show that substantial improvements are possible. In particular, at loss rates around 10 %, a reduction of about 2 dB in distortion is observed, with only three samples delay at the decoder.

2 Coding Framework

This section describes the coding framework. The framework and design method have been presented in [1]. In the following, we summarize and generalize important results of [1], which will be needed in order to establish our main results in Section 3.

2.1 Source Encoder

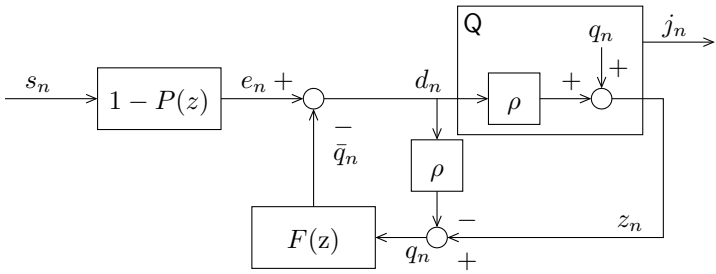


Figure C.1: Source encoder with linear additive noise quantizer model.

The source signal s is modeled as outcomes of an AR process [1]

$$s_n = \sum_{i=1}^{N} \alpha_i s_{n-i} + r_n, \tag{C.1}$$

of order N, driven by zero-mean stationary white Gaussian noise r with known variance σ_r^2. The source process is assumed stable.

The encoder (shown in Figure C.1) is defined by the following equations [1]

$$d_n = e_n - \bar{q}_n, \qquad e_n = s_n - \sum_{i=1}^{p} a_i s_{n-i}, \qquad \bar{q}_n = \sum_{i=1}^{f} b_i q_{n-i}, \tag{C.2}$$

where e_n is the prediction error, $\bar{q}_n$ is the filtered quantization noise feedback, d_n is the input to the quantizer, and q_n is the additive component of the quantization error; p is the predictor order; $a_i,\ i = 1, \ldots, p$ are the predictor coefficients; f is the noise feedback filter order and $b_i,\ i = 1, \ldots, f$ are the noise feedback filter coefficients.

The output transmitted to the decoder is quantization indices, j, for the quantized prediction error, $z_n = Q(d_n)$. The encoder's scalar quantization $Q(\cdot)$ is approximated by the following linear model with additive noise

$$z_n = \rho d_n + q_n, \tag{C.3}$$

where $\rho \in [0, 1]$, and q_n is a stationary zero-mean white Gaussian noise, independent of d_n, with variance $\sigma_q^2 = k \operatorname{var}\{d_n\}$. The quantization noise variance is proportional, by a constant k, to the variance of the input to the quantizer, d_n. ρ and k are given by the coding loss, β, of the quantizer:

$$\rho = 1 - \beta \qquad\qquad k = \beta(1 - \beta),$$

where β is the inverse of the quantizer coding gain [8]. See [1] for a discussion of this linear model of the quantizer.

The quantization noise feecback is de-correlated from d_n by scaling d_n by ρ in the path subtracting the quantizer input from the quantizer output. This ensures that only the additive part of the quantization noise $q_n = z_n - \rho d_n$ is fed back. This allows us to model the fed back noise as white Gaussian which simplifies the optimization of the encoder. The design of Lloyd-Max quantizers for use in the encoder is treated in [1].

2.2 Kalman Filter-Based Decoder

The decoder uses LMMSE (Kalman) estimation of the source signal s based on the measurements z obtained from the partially received sequence of quantization indices j. The Kalman estimator $\hat{s}_n$ of s_n is derived as in, e.g., [9] from a state-space model of the source and encoder. The process equation encompasses both the production of the source signal as well as the evolution of the states of the encoder filters P and F. The measurement equation represents the encoder's filters and the linear model of the quantizer such that the measurements are the quantized prediction errors from the encoder. The following descriptions of the state-space model and decoder equations are a generalization from the state-space model in [1].

The process equation, (C.4), and the measurement equation, (C.5), constitute the state-space model modeling the source and the encoder.

$$\mathbf{x}_{n+1} = \mathbf{F}\mathbf{x}_n + \mathbf{G}\mathbf{w}_n \tag{C.4}$$

$$z_n = \mathbf{h}^{\mathrm{T}}\mathbf{x}_n + q_n \tag{C.5}$$

The state $\mathbf{x}_n$ is composed of the current signal sample and the states of the predictor and noise feedback filter.

$$\mathbf{x}_n = \left[\; s_n \;\middle|\; s_{n-1} \quad \cdots \quad s_{n-(M-1)} \;\middle|\; q_{n-1} \quad \cdots \quad q_{n-f} \;\right]^{\mathrm{T}}, \tag{C.6}$$

where $M = \max\{N, p+1\}$. The state transition matrix $\mathbf{F}$ is defined as

$$\mathbf{F} = \left[\begin{array}{cc|c} \alpha_{1\times N} & \mathbf{0}_{1\times(M-N)} & \mathbf{0}_{M\times f} \\ \mathbf{I}_{M-1} & \mathbf{0}_{(M-1)\times 1} & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{f\times M}} & \begin{array}{cc} \mathbf{0}_{1\times f} & \\ \mathbf{I}_{(f-1)} & \mathbf{0}_{(f-1)\times 1} \end{array} \end{array}\right], \tag{C.7}$$

where $\alpha = [\alpha_1 \cdots \alpha_N]$ are the coefficients of the source AR process, $\mathbf{I}_x$ is an $x \times x$ identity matrix, and $\mathbf{0}$ is an all-zero matrix. The subscripts in (C.7) denote the dimensions of the individual components.

The process noise $\mathbf{w}_n$ is stationary zero-mean white Gaussian with covariance $\mathbf{Q}$

$$\mathbf{w}_n = \begin{bmatrix} r_n \\ q_n \end{bmatrix} \qquad \mathbf{Q} = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_q^2 \end{bmatrix}. \tag{C.8}$$

The measurement noise is the additive part of the quantization noise, q_n, so the measurement noise covariance in typical Kalman filter notation is $\mathbf{R} = \sigma_q^2$. The connection between quantization noise in the state originating from $\mathbf{w}_n$, and q_n added to the measurement is captured by defining the covariance, $\mathbf{S}$, between process and measurement noise as follows:

$$\mathbf{S} = \mathrm{E}\left\{ \begin{bmatrix} r_n \\ q_n \end{bmatrix} q_n \right\} = \begin{bmatrix} 0 \\ \sigma_q^2 \end{bmatrix}. \tag{C.9}$$

We use a formulation of the Kalman filter taking the covariance $\mathbf{S}$ into account. $\mathbf{G}$ is a transform enabling the process noise $\mathbf{w}_n$ to be defined in a compact form

$$\mathbf{G} = \begin{bmatrix} 1 & 0 \\ \mathbf{0}_{(M-1)\times 2} \\ 0 & 1 \\ \mathbf{0}_{(f-1)\times 2} \end{bmatrix}, \text{ or } \mathbf{G} = \begin{bmatrix} 1 & 0 \\ \mathbf{0}_{(M-1)\times 2} \end{bmatrix}, \text{ if } f = 0. \tag{C.10}$$

The measurement vector $\mathbf{h}$ represents the filters P and F as well as the scaling ρ from (C.3).

$$\mathbf{h} = \rho\tilde{\mathbf{h}} \quad \tilde{\mathbf{h}} = \begin{bmatrix} 1 & -a_1 \cdots - a_p & \mathbf{0}_{1\times(M-p-1)} & -b_1 \cdots - b_f \end{bmatrix}^{\mathrm{T}}, \tag{C.11}$$

where $\tilde{\mathbf{h}}$ contains the coefficients of the prediction error and noise feedback filters, cf. (C.2). The zeros fill the measurement vector to match the necessary size of the state in case P is chosen with order $p < N - 1$.

Following the approach described in [1], (C.12)–(C.15) are obtained for the Kalman estimator $\hat{\mathbf{x}}_n$ of $\mathbf{x}_n$ in the case of channel erasures.

$$\hat{\mathbf{x}}_n = \hat{\mathbf{x}}_n^- + \gamma_n \mathbf{P}_n^- \mathbf{h} \left(\mathbf{h}^{\mathrm{T}} \mathbf{P}_n^- \mathbf{h} + \mathbf{R} \right)^{-1} \left(z_n - \mathbf{h}^{\mathrm{T}} \hat{\mathbf{x}}_n^- \right) \tag{C.12}$$

$$\mathbf{P}_n = \mathbf{P}_n^- - \gamma_n \mathbf{P}_n^- \mathbf{h} \left(\mathbf{h}^{\mathrm{T}} \mathbf{P}_n^- \mathbf{h} + \mathbf{R} \right)^{-1} \mathbf{h}^{\mathrm{T}} {\mathbf{P}_n^-}^{\mathrm{T}} \tag{C.13}$$

$$\hat{\mathbf{x}}_{n+1}^- = \mathbf{F} \hat{\mathbf{x}}_n + \gamma_n \mathbf{G} \mathbf{S} \mathbf{R}^{-1} \left(z_n - \mathbf{h}^{\mathrm{T}} \hat{\mathbf{x}}_n \right) \tag{C.14}$$

$$\begin{aligned} \mathbf{P}_{n+1}^- = {} & \left(\mathbf{F} - \gamma_n \mathbf{G} \mathbf{S} \mathbf{R}^{-1} \mathbf{h}^{\mathrm{T}} \right) \mathbf{P}_n \left(\mathbf{F} - \gamma_n \mathbf{G} \mathbf{S} \mathbf{R}^{-1} \mathbf{h}^{\mathrm{T}} \right)^{\mathrm{T}} \\ & + \mathbf{G} \left(\mathbf{Q} - \gamma_n \mathbf{S} \mathbf{R}^{-1} \mathbf{S}^{\mathrm{T}} \right) \mathbf{G}^{\mathrm{T}}, \end{aligned} \tag{C.15}$$

where

$$\hat{\mathbf{x}}_n = \mathrm{E}\left\{ \mathbf{x}_n | z_0 \dots z_n \right\} \quad \hat{\mathbf{x}}_{n+1}^- = \mathrm{E}\left\{ \mathbf{x}_{n+1} | z_0 \dots z_n \right\}, \ \tilde{\mathbf{x}}_n = \mathbf{x}_n - \hat{\mathbf{x}}_n$$

$$\mathbf{P}_n = \mathrm{E}\left\{ \tilde{\mathbf{x}}_n \tilde{\mathbf{x}}_n^{\mathrm{T}} \right\} \qquad \mathbf{P}_{n+1}^- = \mathrm{E}\left\{ \left(\mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1}^- \right) \left(\mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1}^- \right)^{\mathrm{T}} \right\}.$$

3 Fixed-Lag Smoothing

As seen in (C.6), the state $\mathbf{x}_n$ contains both the source sample s_n as well as $M-1$ previous source samples $s_{n-1} \dots s_{n-(M-1)}$. This means that in addition to providing an estimate of the current source sample, $\hat{s}_{n|n}$, the state estimate also provides the delayed source signal estimates $\hat{s}_{n-1|n} \dots \hat{s}_{n-(M-1)|n}$. In this section we show how these estimates provide fixed-lag smoothing.

In [10], the authors describe a fixed-lag smoothing approach with an example of a state-space-model somewhat similar (models an autoregressive source) to the one considered in this paper in Section 2.2. This smoothing approach does not apply to the signal model presented here due to the requirements of [10, eq. (5)].

A different fixed-lag smoothing approach using Kalman filters is found in, e.g., [9, 11]. This smoothing approach is based on an augmentation of the state-space model and requires extensions to the Kalman estimator.

Fixed-lag smoothing with a state-space model similar to the one in Section 2.2 is discussed in [12]. The model in [12] is simpler, e.g., does not contain a separate noise feedback part and the paper does not present mathematical details of estimation improvement by smoothing.

The fixed-lag smoothing approach made possible with our presented state-space model comes at no additional computational cost as it is already an inherent part of the state-space model, i.e., the smoothed (delayed) estimates are readily available from the state estimate.[1] In the following,

[1] We have not been able to prove that our proposed smoothing technique is optimal. However, we have implemented existing smoothing techniques based on augmented state-space models [9, 11] and through simulations observed that the results are indeed identical.

we show under which conditions the delayed estimates $\hat{s}_{n-k|n}$ provide an improvement over $\hat{s}_{n-k|n-k}$ for $k \in [1, M-1]$.

From (C.14) we see that

$$\left[\begin{array}{c|c|c} \hat{s}^-_{n|n-1} & \hat{s}^-_{n-1|n-1} \dots \hat{s}^-_{n-(M-1)|n-1} & \hat{q}^-_{n-1|n-1} \dots \hat{q}^-_{n-f|n-1} \end{array} \right]^{\mathrm{T}} = \mathbf{F} \left[\begin{array}{c|c|c} \hat{s}_{n-1|n-1} & \hat{s}_{n-2|n-1} \dots \hat{s}_{n-M|n-1} & \hat{q}_{n-2|n-1} \dots \hat{q}_{n-f-1|n-1} \end{array} \right]^{\mathrm{T}} + \gamma_n \mathbf{G}\mathbf{S}\mathbf{R}^{-1} \left(z_n - \mathbf{h}^{\mathrm{T}} \hat{\mathbf{x}}_n \right). \quad \text{(C.16)}$$

By the structure of $\mathbf{G}$, $\mathbf{S}$, and $\mathbf{R}$ we see that

$$\mathbf{G}\mathbf{S}\mathbf{R}^{-1} = \begin{cases} \left[\mathbf{0}_{1\times M} \; 1 \; \mathbf{0}_{1\times(f-1)} \right]^{\mathrm{T}} & \text{for } f > 0 \\ \left[\mathbf{0}_{1\times M} \right]^{\mathrm{T}} & \text{for } f = 0, \end{cases} \quad \text{(C.17)}$$

meaning that the right summand in (C.16) affects only $\hat{q}^-_{n-1}$. From the structure of $\mathbf{F}$, see (C.7), we thus conclude that

$$\hat{s}^-_{n-k|n-1} = \hat{s}_{n-k|n-1}, \text{ for } k \in [1, M-1]. \quad \text{(C.18)}$$

It follows from (C.13) and (C.18) that in order for the estimate $\hat{s}_{n-k|n}$ (in $\hat{\mathbf{x}}_n$) to be better than $\hat{s}_{n-k|n-1}$ (in $\hat{\mathbf{x}}_{n-1}$), we require elements $2 \dots M$ on the diagonal of $\mathbf{P}_n$ to be smaller than the corresponding elements on the diagonal of $\mathbf{P}^-_n$. Examining (C.13) it is trivial to see that this is not the case in the event of loss ($\gamma_n = 0$). In the event of correct arrival ($\gamma_n = 1$) we see that elements $2 \dots M$ on the diagonal $\mathbf{d} = \operatorname{diag}\left(\mathbf{P}^-_n \mathbf{h}\mathbf{h}^{\mathrm{T}} \mathbf{P}^{-\mathrm{T}}_n \right)$ in (C.13) must be positive in order to fulfill the requirement.

$$\mathbf{d} = [d_1 \dots d_{2N+1}]^{\mathrm{T}} = \operatorname{diag}\left(\mathbf{P}^-_n \mathbf{h}\mathbf{h}^{\mathrm{T}} \mathbf{P}^{-\mathrm{T}}_n \right). \quad \text{(C.19)}$$

By construction, $\mathbf{h}\mathbf{h}^{\mathrm{T}}$ is a positive semidefinite matrix. By definition of a positive semidefinite matrix we see that the elements d_i on the diagonal $\mathbf{d}$, are $d_i \geq 0$, $\forall i$. Thus it is proved that the estimates $\hat{s}_{n-k|n}$ are never worse than $\hat{s}_{n-k|n-1}$. In fact, improvement is generally guaranteed under the conditions stated in Theorem 1.

Theorem 1. Given at least one of the following conditions:

1. The encoder prediction error filter is $1 - P(z) \neq H^{-1}_{\mathrm{src}}(z)$, i.e., $1 - P(z)$ does not whiten the source completely.
2. There is noise feedback in the encoder: $F(z) \neq 0$.

Then

$$\mathrm{E}\left\{ \left(s_{n-k} - \hat{s}_{n-k|n} \right)^2 \right\} < \mathrm{E}\left\{ \left(s_{n-k} - \hat{s}_{n-k|n-1} \right)^2 \right\}, \text{ for } k \in [1, M-1].$$

Proof of Theorem 1. Consider the diagonal elements d_i from (C.19). Observe from (C.13) that

$$\mathrm{E}\left\{\left(s_{n-k}-\hat{s}_{n-k|n}\right)^2\right\} = \mathrm{E}\left\{\left(s_{n-k}-\hat{s}_{n-k|n-1}\right)^2\right\} - \frac{d_{1+k}}{\mathbf{h}^{\mathrm{T}}\mathbf{P}_n^{-}\mathbf{h}+\mathbf{R}}, \text{ for } k\in[1,M-1].$$

We prove the theorem by showing that $d_i>0$ for $i=2\dots M$ for conditions 1 and 2. Considering $\gamma_n=1$, (C.15) reduces to

$$\mathbf{P}_n^{-}\mathbf{h} = \left(\mathbf{F}-\mathbf{G}\mathbf{S}\mathbf{R}^{-1}\mathbf{h}^{\mathrm{T}}\right)\mathbf{P}_n\left(\mathbf{F}-\mathbf{G}\mathbf{S}\mathbf{R}^{-1}\mathbf{h}^{\mathrm{T}}\right)^{\mathrm{T}}\mathbf{h} + \mathbf{G}\left(\mathbf{Q}-\mathbf{S}\mathbf{R}^{-1}\mathbf{S}^{\mathrm{T}}\right)\mathbf{G}^{\mathrm{T}}\mathbf{h}. \quad \text{(C.20)}$$

Referring to (C.7) and (C.17) ($-\mathbf{G}\mathbf{S}\mathbf{R}^{-1}\mathbf{h}^{\mathrm{T}}$ subtracts $\mathbf{h}^{\mathrm{T}}$ from the $(M+1)th$ row of $\mathbf{F}$) we see that

$$\left(\mathbf{F}-\mathbf{G}\mathbf{S}\mathbf{R}^{-1}\mathbf{h}^{\mathrm{T}}\right)_{(i,j)} = \mathbf{F}_{(i,j)}, \text{ for } i=2\dots M,\ j=1\dots M+f. \quad \text{(C.21)}$$

From (C.8)–(C.10) we see that

$$\left\{\mathbf{G}\left(\mathbf{Q}-\mathbf{S}\mathbf{R}^{-1}\mathbf{S}^{\mathrm{T}}\right)\mathbf{G}^{\mathrm{T}}\right\}_{i,j} = \begin{cases}\sigma_r^2 & \text{for } i=j=1\\ 0 & \text{for } i\neq 1,\ j\neq 1.\end{cases} \quad \text{(C.22)}$$

Let $p_{i,j}$ denote the $(i,j)th$ element of $\mathbf{P}_n^{-}$, $p_{i,j}=p_{j,i}$. Then

$$\begin{aligned}\left\{\mathbf{P}_n^{-}\mathbf{h}\right\}_{(i)} &= \left\{\left(\mathbf{F}-\mathbf{G}\mathbf{S}\mathbf{R}^{-1}\mathbf{h}^{\mathrm{T}}\right)\mathbf{P}_n\left(\mathbf{F}-\mathbf{G}\mathbf{S}\mathbf{R}^{-1}\mathbf{h}^{\mathrm{T}}\right)^{\mathrm{T}}\right\}_{(i,1\dots(M+f))}\mathbf{h} \\ &\quad + \underbrace{\left\{\mathbf{G}\left(\mathbf{Q}-\mathbf{S}\mathbf{R}^{-1}\mathbf{S}^{\mathrm{T}}\right)\mathbf{G}^{\mathrm{T}}\right\}_{(i,1\dots(M+f))}}_{\mathbf{0},\text{ cf. (C.22)}}\mathbf{h},\ \text{for } i=2\dots M \\ &= \Bigg[\underbrace{\left(\sum_{j=1}^{N}\alpha_j p_{i,j}\right)\ p_{i,1}\ \cdots\ p_{i,M-1}}_{\text{(a)}} \\ &\quad \cdot \underbrace{\left(\sum_{j=1}^{M+f}\mathbf{h}_{(j)}p_{i,j}\right)\ p_{i,(M+1)}\ \cdots\ p_{i,(M+f)}}_{\text{(b)}}\Bigg]\mathbf{h},\ \text{for } i=2\dots N+1.\end{aligned} \quad \text{(C.23)}$$

It follows from (C.19) that $d_i=\left(\left\{\mathbf{P}_n^{-}\mathbf{h}\right\}_{(i)}\right)^2$. Observe that under the converse of condition 1, $a_i=\alpha_i$, for $i=0\dots N$ and $a_i=0$, for $i=N+$

$1 \dots M$. Thus under condition 1,

$$\exists i \text{ such that } a_i \neq \begin{cases} \alpha_i & i = 0 \dots N \\ 0 & i = N+1 \dots M. \end{cases} \tag{C.24}$$

By the structure of $\mathbf{h}$, cf. (C.11), and (C.23), part (a), (C.24) guarentees that $d_i > 0$.

Under condition 2,

$$\exists i \text{ such that } b_i \neq 0. \tag{C.25}$$

By the structure of $\mathbf{h}$ and (C.23), part (b), (C.25) guarentees that $d_i > 0$. □

Remark 1. From the state-space model it may seem that it is not possible to obtain smoothed estimates $\hat{s}_{n-l|n}$ for lags $l > M-1$, i.e. that one cannot obtain smoothed estimates beyond the chosen order of the predictor, p. It is however possible to obtain smoothed estimates for arbitrary lags l without increasing the predictor order p, by defining

$$\bar{a}_i = \begin{cases} a_i & i = 0 \dots p \\ 0 & i = p+1 \dots l, \end{cases}$$

where $\{\bar{a}_i\}$ and l are used in place of $\{a_i\}$ and p in the encoder and decoder equations in Sections 2.1 and 2.2.

Smoothed estimates at lags $l > M-1$ come at additional computational cost since they require extension of the signal model beyond what is required to model the encoder and source.

4 Simulations

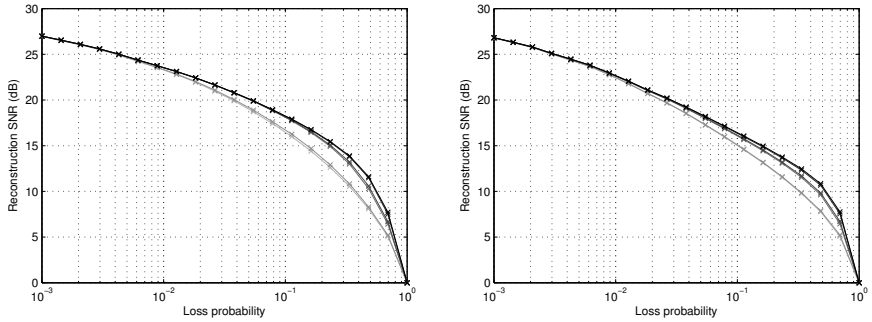
We present results for coding of a stationary source with transmission across an erasure channel. Section 4.1 presents the decoding performance results and Section 4.2 presents the encoder filters designed for the loss statistics under consideration (using the filter design algortihm in [1]) and exemplifies how these filters generally satisfy the conditions in Theorem 1.

4.1 Performance of Fixed-Lag Smooting

A set of Monte Carlo simulations were performed for the order $N = 5$ stationary source defined by the parameters $\alpha_1 = -0.2948$, $\alpha_2 = -0.9527$, $\alpha_3 = -0.0032$, $\alpha_4 = 0.0040$, $\alpha_5 = 0.1995$.

We consider channel erasure probabilities between 1×10^{-3} and 1 and simulate i.i.d. losses as well as Gilbert-Elliot (GE) losses with a mean error burst length 3, denoted "GE-3". Simulations have been performed

for identical source sequences of length 1×10^6 at the simulated erasure probabilities. The quantizer is a Lloyd-Max quantizer at 4 bits/sample. In this example, we set the encoder filter orders to $p = f = N$.



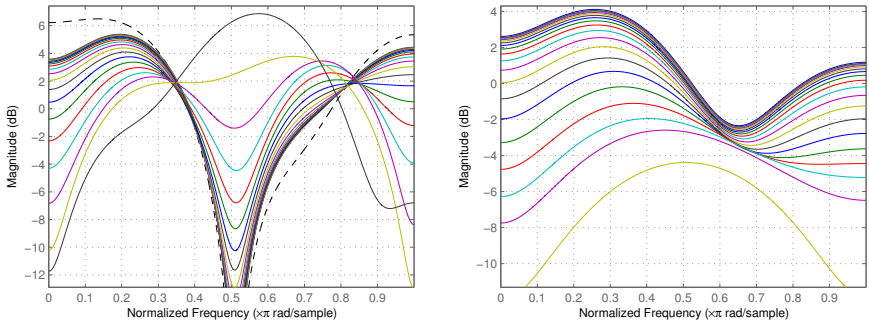
(a) (b)

Figure C.2: Examples of coding: (a) i.i.d. channel erasures, (b) GE-3 channel erasures. Lloyd-Max quantization at 4 bits/sample. Results for increasing smoothing lags (0-5 samples) are plotted (from bottom to top) in increasingly darker shades of grey.

Signal-to-noise ratios (SNRs) of the decoded signals are shown in Figure C.2a for i.i.d. channel erasures and in Figure C.2b for GE-3. The figures clearly show substantial improvement in SNR when using smoothing. In the i.i.d. loss case, the improvement by smoothing is up to 3.5 dB (at 48.3 % loss prob.) and approximately 2 dB at 10 %. The improvement is most pronounced at high loss rates at which the decoding error continues to decrease significantly up to lag 4 out of 5 shown in the figure. In the GE-3 loss case, the decoding error is evidently worse due to the correlated losses. However an improvement in SNR of up to 3.0 dB is still achievable in this case (at 48.3 % loss prob.) and approximately 1.5 dB at 10 %.

4.2 Examples of Encoder Filters

Figure C.3a depicts the magnitude spectra of the encoder prediction error filters $1 - P(z)$ designed for each of the channel erasure probabilities considered in Section 4.1. The encoder noise feedback filter spectra are depicted in Figure C.3b. The inverse of the source magnitude spectrum is plotted as a dashed line. Figure C.3 was included to emphasize that the filter design algorithm generally produces encoder filters that fullfil both of the requirements in Theorem 1. Figure C.3a shows that the prediction error filters do not equal the inverse of the source model and thus fulfill Theorem 1,



(a) (b)

Figure C.3: Encoder filters designed for the example in Section 4.1: (a) $1 - P(z)$ magnitude spectra for the erasure probabilities simulated in the example, the dashed line depicts the magnitude spectrum of the inverse of the source model; (b) accompanying $F(z)$ magnitude spectra. The spectra are designed for increasing loss probability from top to bottom.

condition 1. Figure C.3b shows that the noise feedback filters are generally non-zero and thus fulfill Theorem 1, condition 2.

5 Conclusions

We have presented a generalization of the source coding and filter design framework previously introduced in [1], providing fixed-lag smoothing up to arbitrary lengths as well as encoder prediction error and noise feedback filters of arbitrary orders. These properties were constrained to the source model order in our previous work.

We have examined the fixed-lag smoothing properties of the coding framework and pointed out that smoothed (delayed) estimates at the decoder for lags up to at least $\max\{N-1, p\}$ (source model and predictor orders N and p, resp.) are readily available at no additional computational cost which is not the case for other more general smoothing approaches associated with Kalman filtering. We have provided proof that the estimation error of these delayed estimates is guaranteed to decrease under simple conditions fulfilled by the filter design approach of the coding framework.

We have provided simulation results that demonstrate how the described smoothing approach can provide substantial improvements in estimation accuracy. Furthermore we have shown accompanying filter design examples which support the theoretical foundations for the observed improvements in estimation accuracy by smoothing.

References

[1] T. Arildsen et al. „On Predictive Coding for Erasure Channels Using a Kalman Framework". In: *IEEE Transactions on Signal Processing* 57.11 (Nov. 2009), pp. 4456–4466. ISSN: 1053-587X. DOI: 10.1109/TSP.2009.2025796.

[2] B. Sinopoli et al. „Kalman filtering with intermittent observations". In: *IEEE Transactions on Automatic Control* 49.9 (2004), pp. 1453–1464. ISSN: 0018-9286. DOI: 10.1109/TAC.2004.834121.

[3] L. Schenato et al. „Foundations of Control and Estimation Over Lossy Networks". In: *Proceedings of the IEEE* 95.1 (2007), pp. 163–187. ISSN: 0018-9219. DOI: 10.1109/JPROC.2006.887306.

[4] A. K. Fletcher, S. Rangan, and V. K. Goyal. „Estimation from lossy sensor data: jump linear modeling and Kalman filtering". In: *Proceedings of the third international symposium on information processing in sensor networks*. Berkeley, California, USA: ACM Press, 2004, pp. 251–258. DOI: 10.1145/984622.984659.

[5] A. K. Fletcher et al. „Robust Predictive Quantization: Analysis and Design Via Convex Optimization". In: *IEEE Journal of Selected Topics in Signal Processing* 1.4 (Dec. 2007), pp. 618–632. ISSN: 1932-4553. DOI: 10.1109/JSTSP.2007.910622.

[6] O. L. V. Costa. „Linear minimum mean square error estimation for discrete-time Markovian jump linear systems". In: *IEEE Transactions on Automatic Control* 39.8 (1994), pp. 1685–1689. ISSN: 0018-9286. DOI: 10.1109/9.310052.

[7] S. C. Smith and P. Seiler. „Estimation with lossy measurements: jump estimators for jump systems". In: *IEEE Transactions on Automatic Control* 48.12 (Dec. 2003), pp. 2163–2171. ISSN: 0018-9286. DOI: 10.1109/TAC.2003.820140.

[8] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[9] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1979, repr. Minneola, New York: Dover Publications Inc., 2005. ISBN: 0-486-43938-0.

[10] P. Hedelin and I. Jönsson. „Applying a smoothing criterion to the Kalman filter". In: *IEEE Transactions on Automatic Control* 23.5 (1978), pp. 916–921. ISSN: 0018-9286.

[11] G. C. Goodwin and K. S. Sin. *Adaptive Filtering: Prediction and Control (Prentice-Hall Information and System Sciences Series)*. Prentice Hall, 1984. ISBN: 013004069X.

[12] S. Crisafulli, J. D. Mills, and R. R. Bitmead. „Kalman filtering techniques in speech coding“. In: *1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. 1992, 77–80 vol.1. DOI: 10.1109/ICASSP.1992.225968.

Publication D

Cross Layer Protocol Design for Wireless Communication

Thomas Arildsen and Frank H.P. Fitzek

This book chapter was originally published as:
T. Arildsen and F. H. P. Fitzek. „Cross Layer Protocol Design for Wireless Communication“. In: *Mobile Phone Programming and its Application to Wireless Networking*. Ed. by F. H. P. Fitzek and F. Reichert. 1st ed. Dordrecht, The Netherlands: Springer, 2007. Chap. 17, pp. 343–362. ISBN: 978-1-4020-5968-1. DOI: 10.1007/978-1-4020-5969-8_17.

The current layout and citation numbering has been revised compared to the published version.

Abstract

This chapter provides an introduction to cross-layer protocol design. Cross-layer design is needed in order to enhance data transmission across especially wireless network connections. This enhancement is achievable by being able to optimize certain parameters jointly across multiple layers in stead of considering each layer separately. Cross-layer protocol design is a principle that provides the possibility of enhancing the architecture and operation of the layered protocol stack by allowing communication between non-adjacent layers as an extension of what is possible in the OSI model. We briefly summarize the layered network protocol stack concept of the ISO OSI model and explain cross-layer communication in this context. We provide an overview of different categories of exchanging data across layers of the network protocol stack and point out advantages and drawbacks of different approaches. We provide examples from the literature of such approaches. Similarly, we review a number of recent examples from research literature of how cross-layer protocol design is being put to use. The examples provide an overview of which layers are involved in different cross-layer optimization approaches. The examples also provide an overview of which technological areas are currently considered in – and which optimization techniques employ – cross-layer design.

1 Introduction

Recent years' development within wireless communication means that wireless data transmission has found its way into a wide range of mobile or portable consumer electronics. The use of mobile phones is gradually shifting its focus from voice-only applications to multimedia streaming, Internet browsing, file downloading, etc. Network access from computers is now commonly wireless, e.g. 802.11, UMTS, WiMAXX.

The various network protocols generally use the layered architecture known from the ISO open systems interconnection (OSI) model [1]. The OSI model is a descriptive network scheme designed to facilitate inter-operation between various network technologies. The model describes how data is transferred from one application on a computer system to another application on another computer system through a network medium. The OSI model deals with this data transfer by dividing the involved tasks into seven separate layers as shown in Figure D.1.

The division into separate layers reduces the complexity of the protocol stack, because it makes it possible to restrict one's attention to one specific layer at a time while other layers can be abstracted from that layer by standardized interfaces. Each layer communicates with the corresponding layer at the other end of the network through the layers below it. Each layer

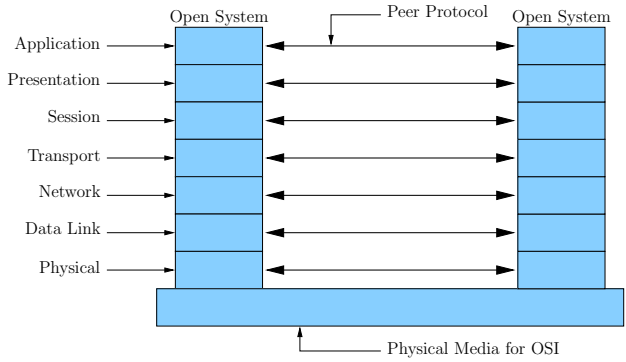


Figure D.1: The layers of the OSI model.

uses functionality of the layer below it and provides its services to the layer above it. Thus network layers in the OSI model only communicate directly with their immediate neighbors. The seven layers are the following:

Application Layer provides network functionality to applications outside the protocol stack (e.g. user applications such as office applications, games, etc.). The application layer establishes the availability of intended communication partners, synchronies and establishes agreement on procedures for error recovery and control of data integrity.

Presentation Layer ensures that information sent by the application layer of one system is readable by the application layer of another system. If necessary, the presentation layer translates between multiple data formats by using a common format. It provides encryption and compression of data.

Session Layer defines how to start, control and end conversations (called sessions) between applications. It also synchronies dialogue between two hosts' presentation layers and manages their data exchange. The session layer provides efficient data transfer.

Transport Layer regulates information flow to ensure end-to-end connectivity between host applications reliably and accurately. The transport layer segments data from the sending host's system and reassembles the data into a data stream on the receiving host's system.

The boundary between the session layer and the transport layer can be thought of as the boundary between application protocols and dataflow protocols. The application, presentation, and session layers are concerned with application issues, whereas the lower four layers are concerned with data transport issues.

Network Layer provides end-to-end delivery of packets across the network. Defines logical addressing so that any endpoint can be identified. Defines how routing works and how routes are learned so that the packets can be delivered. The network layer also defines how to fragment a packet into smaller packets to accommodate different media.

Data Link Layer provides access to the networking medium and physical transmission across the medium and this enables the data to locate its intended destination on a network. The data link layer provides reliable transmission of data across a physical link by using the Medium Access Control (MAC) addresses. The data link layer uses the MAC address to define a hardware or data link address in order for multiple stations to share the same medium and still uniquely identify each other. The data link layer is concerned with network topology, network access, error notification, ordered delivery of frames, and flow control.

Physical Layer deals with the physical characteristics of the transmission medium. It defines the electrical, mechanical, procedural, and functional specifications for activating, maintaining, and deactivating the physical link between end systems.

Layer-N entities in a network communicate with layer-N peer entities at the other end(s) of the link/network using a specific layer-N protocol. They communicate through the facilities offered to them by layer $N-1$. Thus information transmitted from a network node propagates from the top to the bottom of its protocol stack before finally being transmitted across the physical medium. Each layer encapsulates the data from its higher adjacent layer and adds its own control information. Each layer may partition the data from the higher layer into several parts or collect multiple parts into one before handing it to the its lower adjacent layer. The corresponding inverse operations take place in reverse order at the receiving network node(s). An example of data propagating through the protocol of a transmitting network node is shown in Figure D.2.

The layered architecture facilitates development of protocol components by abstraction such that a particular layer only has to concern itself with the interfaces to the layer above it and to the layer below it. This modularity facilitates development of protocols, because individual layers can be tested separately. It also allows developers of network protocols to contribute with a particular layer in stead of an entire protocol encompassing functionality corresponding to all layers. The layered approach facilitates standardization as well, again through division into smaller separate parts.

The layered architecture allows different applications to transport data in several different ways across the same networks and it allows networks

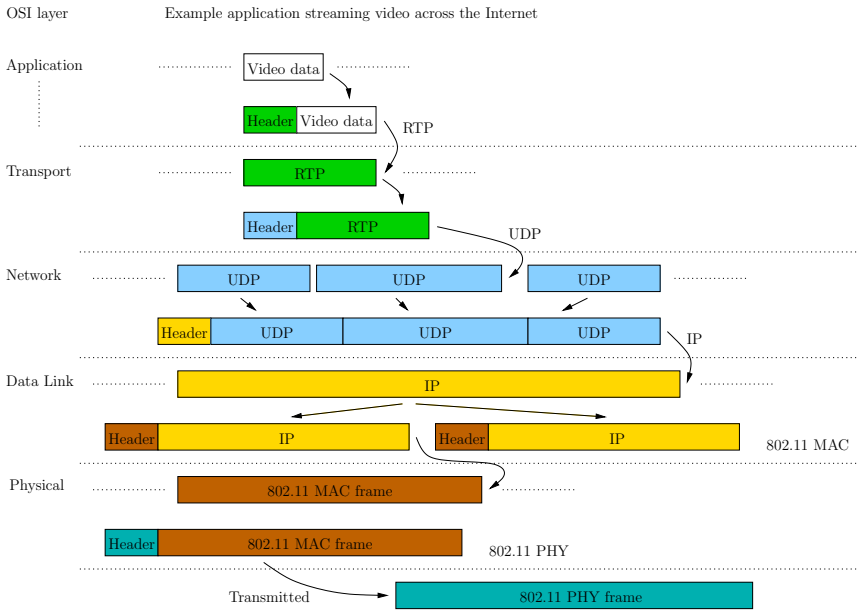


Figure D.2: Data transmission in the OSI model with an example showing a scenario in which an application transmits video across a wireless Internet connection.

to be connected across a heterogeneous variety of physical media - e.g. the Internet.

The layered network protocol architecture has enabled the evolution of networks into what they are today. However, many protocols were mainly developed for cabled networks and work very well with these. The fast growth of wireless network technologies means that wireless network access is becoming more and more common, especially on the last hop to the users. The wireless medium has very different properties in terms of for example channel fading and interference. Due to these differences, the existing layered protocols have several drawbacks and there is room for improvement in the way protocols at different layers cooperate.

2 Cross-Layer Protocol Design

2.1 The Principle

The principle of cross-layer protocol design is to extend the architecture of the layered protocol stack to allow communication between non-neighboring layers in addition to what is already possible as described in Section 1 and to allow reading and controlling parameters of one layer from other layers. By this definition, cross-layer protocol design is a very broad subject. Therefore, we will in the following describe what we need it for, how to organize the actual communication across the protocol stack, and how cross-layer design is currently being employed in the research literature.

At each layer in the protocol stack, various choices exist for transmission of the data units passing through them. This could be different speech codecs for a VoIP application, different transport protocols such as TCP or UDP at the transport layer, or for example, at the physical layer, different modulation types for transmission on a wireless channel. Such possibilities at the different layers constitute a flexibility in the overall protocol stack. Cross-layer design plays an important role in relation to this flexibility. Cross-layer design is a means by which one can get specific knowledge across the protocol stack between separate layers and thus exploit the flexibility through making the protocol stack adaptive. Cross-layer design makes it possible to control features of different protocol layers jointly across the network protocol stack.

Protocols at different layers of the protocol stack may implement similar functionality. This could introduce redundant operations in the protocol stack, e.g. forward error-correction (FEC) being applied at two different layers etc. In stead, such redundant operations could for example be undertaken at only one of the layers by appropriate adjustments or jointly controlled for overall optimal operation of the functionality in question at the involved layers. Similarly, different protocols could implement complementary operations at different layers which could be jointly optimized in order to exploit the resulting collective operations more efficiently. An example could be to coordinate the operation of automatic repeat request (ARQ) at one layer with a FEC mechanism at another layer such that in case a very strong FEC mechanism is currently applied at one layer, it might not be necessary to actually use ARQ at the other layer because FEC will be able to compensate most of the data losses – or vice-versa.

Considering how to approach cross-layer design, it should be a "non-destructive" approach. The layered protocol architecture is fundamentally a good idea and should be kept intact - it still provides us with for example the flexibility of being able to adapt the protocol stack to different radio technologies by replacing some of the lower layers or to use different applications at the highest layers. Cross-layer design should be used to enhance

the existing architecture by exploiting opportunities of jointly optimizing parameters/behavior of the protocol layers that would not otherwise be possible.

Cross-layer design enables performance gains in a multitude of different aspects of wireless networking. However, it also potentially goes against some important benefits of the original layered architecture. A given cross-layer design implementation may inter-weave two or more layers such that these layers cannot be separated. It may introduce dependencies in the protocol such that one protocol cannot simply replace another one at a certain layer. This means that it potentially degrades the modularity and freedom to 'compose' the protocol stack. Another thing to consider is that some cross-layer optimizations may drastically increase the computational demands of running the protocol stack due to the additional degrees of freedom that are introduced in optimizing the protocol stack's performance.

Cross-layer designs need to be considered carefully. In any design implemented, one should consider its possible impact on the existing protocol stack. This is for example pointed out in a somewhat pessimistic way in [2].

In the following sections we first describe different types of communication across network protocol layers in Section 2.2 and then review some of the existing ideas for cross-layer optimizations in Section 2.3.

2.2 Communication Across Protocol Layers

One can choose to see cross-layer design from different viewpoints. One viewpoint that needs to be considered for practical deployment of cross-layer optimizations is how to integrate the cross-layer design into the protocol stack – how is the communication between different layers going to be realized?

We consider the following two categories of cross-layer communication: Using existing protocols, with the sub-categories implicit and explicit, or using dedicated signaling mechanisms, with the sub-categories of signaling pipes, direct communication, and external cross-layer management. These categories are explained in the following:

1. **Communication using existing protocols** We divide this category into implicit/inherent and explicit communication using existing protocols:

    a) **Implicit/Inherent** Cross-layer communication here simply consists of lower layers reading and/or perhaps altering data within the data units passing through them, belonging to higher layers. In this type of cross-layer communication, no additional data is transferred between layers compared to what is already the case within the traditional OSI architecture. The only difference lies in the fact that lower layers snoop into higher layers' packets to

gain knowledge of what is taking place here and to exploit this. We call this kind of communication inherent or implicit because the data exploited across layers is already available.

One clear advantage to this type of communication is that nothing needs to be changed except at the layer(s) that will be snooping into the data of other layers. The obvious drawback is that the data that can be exploited is limited to whatever data is transmitted by the higher layers. Another drawback is that data can only be exploited at lower layers relative to the layer to which the particular data belongs. This is due to the fact that the data needs to pass through the layer interested in that data, so for example, the transport layer cannot gain knowledge of the link layer using this type of communication. However, lower layers can possibly alter data passing through them to higher layers and thus manipulate the operation of these higher layers. The higher layers in question will generally be unaware of this interaction.

b) **Explicit** Cross-layer communication in this category is an extension of the above-mentioned category. The communication here is explicit since involved layers are aware of it and actively participating in the communication. However, the data is transferred by means of already existing protocols through the interfaces between layers defined by the OSI model.

Using this method, additional information can be transferred between layers compared to what is transferred between layers in the traditional approach. This is accomplished because the interacting layers are aware of the communication and can be designed to exchange specific additional information between them. However, there are still limitations as to where data can be sent to and from due to the use of existing protocols.

In addition to the above-mentioned drawbacks and benefits of using existing protocol formats, one could also mention the following, common to both implicit and explicit cross-layer communication:

- The information from higher layers may be difficult to access by lower layers due to for example segmentation, blocking, and concatenation of the higher layers' data units as illustrated in Figure D.2. However, information flowing downward is easier to convey than information flowing upward since a particular layer's data units will pass through lower layers.
- Information from lower layers to higher layers is difficult to convey by these mechanisms since for example inserting extra information in packets to higher layers passing through a par-

ticular lower layer requires alteration of the packets including check sums and other content-dependent parameters, and the concerned packets may be segmented, concatenated etc. across multiple of the lower layer's data units, as illustrated in Figure D.2. In addition, lower layers would have to rely on data units, which they do not initiate, passing through them to upper layers.

- There is also a clear advantage of exchanging information through the mechanisms of existing protocols. The information exchange is transparent to intermediate layers so it does not require changing the protocols of intermediate layers in order get information through them.

2. **Explicit communication using dedicated mechanisms** Introducing dedicated mechanisms for communication across the layers of the protocol stack gives the ultimate freedom since the communication is not bound by restrictions of existing protocols that were not designed for this kind of mechanisms. Within these dedicated mechanisms we consider three different kinds:

   a) The first mechanism arranges exchange of data across the layers as a 'signaling pipe' traversing all layers through which any layer can send data to or receive data from any other layer. This

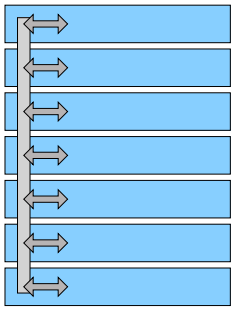


Figure D.3: Dedicated signaling mechanism: a signaling pipe across the entire protocol stack.

mechanism is illustrated in Figure D.3. This provides a general framework under which cross-layer optimizations can be introduced at any layer, taking advantage of the available communication mechanism. However, all layers of the protocol stack must be modified to implement the signaling mechanism. Using this type of cross-layer communication also implies that any cross-

layer optimization must be implemented inside one or more of the layers.

b) The second mechanism is a more specialized approach where signaling interfaces are introduced specifically for direct communication between interacting layers. It is illustrated in Figure D.4.

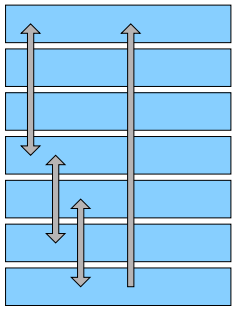


Figure D.4: Dedicated signaling mechanism: specific interfaces between interacting layers.

This allows introduction of dedicated signaling mechanisms only where needed. This could for example also provide benefits related to timing considerations where the above-mentioned signaling pipe as a general framework would be a too slow mechanism for very time-critical signaling. Cross-layer optimizations also still need to be implemented inside one or more of the interacting layers.

c) The third mechanism is a general mechanism as the first one, but in stead of data being exchanged through the layers of the protocol stack, an external management mechanism is introduced as illustrated in Figure D.5. The layers of the protocol stack communicate individually with the external cross-layer manager. In this way, each layer only needs to consider cross-layer communication with one other party. All cross-layer optimization operations can be collected in the cross-layer manager which can take data from several layers into account and control these jointly. However, this type of mechanism may be considerably more computationally demanding and difficult to implement since this mechanism, among the three mentioned here, will require the most extra functionality compared to a traditional OSI architecture.

The approach of letting layers communicate directly with each other or with some management middle-ware external to the protocol layers allows the designer to tailor the framework exactly to the cross-layer

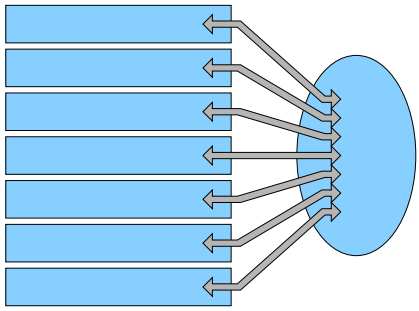


Figure D.5: Dedicated signaling mechanism: communication through an external cross-layer manager.

optimizations in question. Especially if the optimization is of a kind that incorporates information from and/or control of more than two layers, the centralized management approach may be beneficial. However, it should be noted that there are also considerable drawbacks of this approach. It requires much more customization of the involved protocols in order to enable the communication mechanisms since the traditional protocols' mechanisms can not be used in this context. Furthermore, the added communication mechanisms between the layers will be likely to increase the computational requirements of the system which may have very limited resources, especially in the case of mobile phones and similar devices.

In the following sections, existing work is covered that deals with cross-layer design concerning how to realize communication across the different layers. Since cross-layer design is a relatively young research area, there is not yet any standardization in the area concerning actual frameworks for realizing cross-layer designs. There have however been some attempts at defining such frameworks.

2.2.1 Explicit Communication Based on Existing Protocols

An example of the approach described as method 1a is [3]. In this example, a lower layer (MAC) simply reads priority information set in a header from a higher layer (application – the video coder's network adaptation layer). This is data already set at the application layer and all that is needed in order to introduce the cross-layer data exchange is to modify the link (MAC) layer to be able to read it.

Several examples exist of the approach described in the preceding section as method 1b. An early example of such an approach is Wu et. al.'s article from 1999 where they suggest Interlayer Signaling Pipes (ISP) which con-

sists of exchanging the relevant information in the IPv6 header field Wireless Extension Header (WEH) [4]. This approach can actually be said to be an implementation of the category 2a approach by category 1b mechanisms. In [5] from 2001, Sudame & Badrinath suggest a framework, complete with API for Linux, that exchanges cross-layer information based on ICMP packets. Examples are provided with information from the driver (layers below network layer) exploited in application and transport layers. Mérigeault & Lamy's article from 2003 suggests a concept for signaling information between application and network access[1] layers [6]. They do so by piping the extra information through RTP packets. It is accomplished by means of network adaptation layers (NAL) in the form a a Source Adaptation Layer (SAL) between application and transport layers and a Channel Adaptation Layer (CAL) between network and network access layers. The SAL and CAL handle the cross-layer information in the extra packets in relation to their protocol layers and filter out RTCP packets generated by the transport layer in response to the "artificial" extra packets. A similar concept is pursued much more extensively by the PHOENIX IST project that attempts to establish a framework for adapting video transmission in a cross-layer manner across an IPv6 network [7].

2.2.2 Dedicated Signaling Mechanisms

Some suggestions for cross-layer signaling frameworks approach the problem by introducing mechanisms to let any layers communicate directly with each other through new interfaces that circumvent the standardized interfaces of the OSI protocol stack (category 2b). One such example is [8]. The example is not very detailed, but rather describes the general idea and points out its suggested framework, CLASS, as light-weight with high flexibility and fast signaling but with high complexity.

The bulk of this family of solutions however suggest some sort of management entity external to all the layers with which all of the involved layers will communicate individually and which will handle the considered cross-layer optimizations in a centralized manner, such as described in category 2c. One of the earliest examples is by Inouye et. al. from 1997 who suggest a framework for mobile computers for adaptation to the availability of different network interfaces based on several parameters that define this availability [9]. Recent examples of cross-layer frameworks include [10] which arranges cross-layer optimizations in so-called coordination planes and handles them through a cross-layer manager. There is the quite extensive ECLAIR framework [11] which introduces so-called tuning layers for access to the individual layers' parameters and a collection of "protocol

[1]Network access layer is used as a common name for DLC/MAC + PHY in this paper.

optimizers" for handling the individual optimizations. The approach in [12] has a so-called local view for storing parameters from individual layers for access from all layers as well as a corresponding global view that serves the same purpose in a network-wide scope for cross-layer optimizations that span multiple network devices.

2.3 State of the Art

In this section, we go through some of the most recent work in the area to characterize which parts of the protocol stack are involved, what sort of information is exploited, and what purposes the optimizations serve. The overview is organized by most significant involved layer from the bottom up.

Considering what to optimize at the different layers, Raisinghani & Iyer's 2004 article provides a brief overview of what topics could be considered [13]. They go through the layers approximately corresponding to the OSI model and suggest parameters that could be interesting at the particular layer and what sort of interaction could be relevant with lower and higher layers, respectively. This is a starting point to getting an overview of cross-layer protocol design.

This section provides a more extensive overview of existing research in the field and attempts to bring it up to date.

2.3.1 Physical/Link Layers

While quite a lot of research work includes the physical layer in cross-layer optimizations, there is little focus mainly on this layer. Much of the work involving the physical layer revolves around the data link layer, especially MAC. For example, Alonso & Agustí focus on MAC-PHY interaction in [14, 15]. In the former they present a method using distributed queueing random access protocol (DQRAP) in a CDMA system to increase overall throughput and minimize power consumption and thus inter-cell interference. Cross-layer information consists of channel state information (CSI) and target spreading factor to reach a desired bit-error rate transmitted from receiver PHY to transmitter MAC[2]. In the latter they consider scheduling/prioritization in a CDMA base station MAC based on CSI from mobile nodes. They also cover somewhat the same as in the former paper and extend the principle to WLAN. In [14] they do not clearly specify how the information is exchanged between layers and in [15] it is stated that the layers interchange explicit control information by means of specific control channels – what we define as dedicated signaling mechanisms. In [16], which Alonso among others co-authored, they explore prioritization between

[2]In this case, information is exchanged not only across layers within the protocol stack of one mobile device, but between separate mobile devices as well.

real-time (VoIP) and non-real-time traffic in a distributed-queueing MAC. Channel quality obtained from the physical layer is used to dynamically prioritize users with good conditions in order to improve overall throughput. It is merely stated that the MAC (link) layer acquires information through a "cross-layer dialogue" with the physical layer which could be implemented either through existing protocols or through a dedicated signaling mechanism.

Toufik & Knopp in [17] consider sub-carrier allocation and antenna selection in a MIMO OFDMA system taking CSI from PHY into account in the allocation at MAC level. Song & Li's work is related in the sense that they consider sub-carrier and power allocation in an OFDMA system. In [18] they optimize the system by maximizing utility functions based on transmission rates of the users, taking CSI into account. In [19] they extend it to base the utility functions on waiting time in addition to rates. They take it further in [20] where they mix the different utility maximizations from the two former papers in their simulations of a scenario consisting of users with different delay requirements, categorized as voice, streaming, and best-effort traffic users. Delay is very important to voice users, somewhat less important to streaming users, and has much lower importance to best-effort users. Filin et. al.'s paper [21] also considers resource allocation in an OFDMA system, but they focus on minimizing the time-frequency resource usage through a more heuristic approach. Their work includes MAC through scheduling of data flow segments as well as the physical layer through the control of transmission power, modulation, and coding as well as the use of SNR estimates. Compared to the previously mentioned approaches in this section, which all use information from the physical layer to change parameters in the link layer, [21] is interesting because they also control parameters in the physical layer based on information from the link layer. The above-mentioned five papers concentrate particularly on the optimizations, using the necessary parameters without considering how to exchange these between layers.

Zhang et. al.'s paper from 2006, [22], integrates physical and link layers in a model of the impact of physical layer Adaptive Modulation and Coding (AMC) and MIMO on link layer QoS provisioning. They model the physical layer service process as a finite-state Markov chain. As such, neither does this paper cover how to exchange the information between layers.

Fawal et. al. in [23] also look at MAC-PHY interaction. Their work concentrates on impulse radio UWB (IR-UWB) where they describe a selection of aspects and how to address them with a PHY-aware MAC implementation. Their objective is mainly to achieve energy-efficiency. Energy efficiency is also an issue in [24] where Mišić et. al. attempt to increase the lifetime of sensor networks by managing sensor activity taking MAC-layer congestion and interference as well as noise from PHY into consideration. None of these two articles seem to be concerned with how to exchange the

actual information between layers.

2.3.2 Physical/Application Layers

[25] provides an example of an optimization involving the physical layer and the application layer. It is an extensive framework developed under the PHOENIX project [26]. It incorporates the mentioned layers at both transmitter and receiver side in optimizing transmission of a video stream. Thus the information exchange takes place both across layers in the individual mobile devices as well as across the network between individual devices. Parameters involved in the optimization are source significance information (SSI), source a-priori information, source a-posteriori information, and a video quality measure, all from the application layer, network state information from intermediate layers, as well as CSI and decision reliability information (DRI) from the physical layer. This is a very extensive and advanced piece of work. As mentioned earlier, an important aspect of the PHOENIX project, e.g. in [25, 27], is the framework for exchanging cross-layer information through already present protocols, e.g. via IPv6 extension headers and ICMPv6. Thus, the information exchange considered here is explicit, utilizing the existing protocols' capabilities.

2.3.3 Link/Network Layers

Tseng et. al. address optimization of hand-off in a Mobile IP/802.11 scenario [28]. In stead of reading/adjusting parameters across layers, as most other examples do, the authors here signal events across layers. As such, this work is interesting because it is not a matter of reading parameters from other layers. Rather, it is a matter of the timing of specific signals introduced between the layers which constitutes the difference compared to the traditional OSI approach. Hand-off-related events are signaled between the link layer of the 802.11 protocol and the Mobile IP network layer in order to speed up the hand-off process. It is not stated how the information is exchanged between layers, but due to the timing-critical nature of the signaling it is likely that one cannot rely on the existing protocols and that dedicated signaling mechanisms between the two layers must be employed. The IEEE 802.21 working group is, among other things, dealing specifically with this type of signaling [29].

2.3.4 Link/Transport Layers

The examples presented for this combination of protocol layers are focused on taking action in the transport layer based on information from the link layer, i.e. an upward information flow.

One example of link/transport layer interaction is Wu et. al.'s [4]. They investigate a cross-layer scheme involving TCP at the transport layer and

Radio Link Protocol (RLP) at the link layer in the IS-707[3] standard. What they do is to enable TCP to exploit radio link parameters such as data rate, radio link round-trip delay, and fading conditions. This is done in order to mitigate unfortunate effects of TCP in a wireless environment. Wu et. al. exchange cross-layer information through existing protocol mechanisms.

Sudame & Badrinath's [5] is another example of link/transport layer interaction although the details of which lower layer is involved in the setup is not defined very precisely in OSI-terms. They test their proposed framework using modifications to UDP and TCP to decrease packet losses and delays, respectively, in case of WLAN hand-overs. The hand-overs are signaled from the network interface driver by means of ICMP messages. This approach is thus related to the previously mentioned [28] by their attempts to mitigate unwanted effects of hand-overs in the network. This is also an example of exchanging information through mechanisms of existing protocols.

2.3.5 Link/Application Layers

An example of a link-/application- cross-layer design is Liebl et. al.'s [30] from 2004 where they consider VoIP over a packet-switched connection in a GERAN scenario with AMR speech coding. They attempt to optimize the utilization ratio of radio link control (RLC) data segments. This approach is centered around the application layer exploiting information (current segment size and utilization) from the link layer (upward information flow). The AMR speech coder's mode is selected to fit the resulting speech frames into a number of RLC segments minimizing the bit-stuffing of these. It is not mentioned how the relevant information should be exchanged between the involved layers.

Ksentini et. al.'s recent work published in 2006 combines layered video coding at the application layer with classification of the video layers[4] into different QoS classes at the MAC layer [3], so this can be considered a downward information flow. Here, an implicit cross-layer information exchange through existing protocols is employed since they use a priority field set by the NAL of the video coder to classify the importance of the video packets at the MAC layer. One more example is found in Liu et. al.'s 2006 paper [31]. They employ information on the importance of video frames from the application layer at the link layer. As such, it can be compared to [3], but it is slightly different in the sense that Liu et. al. consider differentially encoded video frames in a video group-of-pictures (GOP) to have decreasing importance according to their sequence number within the GOP. Based on this importance they use an ARQ scheme in which retransmission attempts

[3] A data service standard for a wide-band spread spectrum system.

[4] This is not layers in the network stack sense; this is a way of partitioning the data produced by the video coder.

are spent on video frames in decreasing order of importance in order to minimize the impact of packet losses on video quality. The link layer needs to know which position in a GOP each video frame has, but the paper does not specify how this information is obtained from the application layer. Haratcherev et. al. present a somewhat more advanced scheme in which both video coding rate and link layer transmission rate are adapted based on CSI and link throughput provided by a so-called channel state predictor and a medium sharing predictor [32]. This employs mainly an upward information flow. The paper does not specify exactly how the information exchange between application and link layer was achieved.

Jenkac et. al. explore FEC in [33]. They consider several error correction suggestions at different layers, but their most significant contribution in this article is their so-called permeable layer receiver in which error correction is performed at the application layer. The transport layer is also involved to some degree since the FEC produces extra parity data packaged in extra RTP packets associated with data RTP packets. The most important cross-layer aspect is in the receiver where erroneously received link layer segments (destroyed due to one or more radio bursts with errors) do not cause the entire corresponding RTP packet to be discarded. In stead, the data is passed up from the link layer with erasure symbols in the missing segments and the FEC mechanism attempts to reconstruct the erroneous data. It will require the link layer to be aware of the mechanism, but the cross-layer data exchange as such is implicit since it merely requires passing packets with erasure symbols up the stack in case of errors in the same way as with correctly received packets. This also illustrates the advantages of using the existing protocols since this is completely transparent to intermediate layers and utilizes mechanisms that are already available.

2.3.6 Physical/Link/Application Layers

All so far mentioned examples of specific cross-layer optimizations have only incorporated two layers. There are also recent examples of research involving three layers.

Jiang et. al. explore a concept in which video is classified into different priority classes [34], similarly to other examples mentioned in the previous section. Here, transmission is adapted at link layer according to both the video data importance and the users' CSI on a cell-wide basis in a CDMA cellular system. This is accomplished by dynamic-weight generalized processor sharing (DWGPS). It is not directly addressed how to exchange the required information between the layers. Likewise, Khan et. al. focus on transmission of video in [35]. They do so by jointly adjusting video source rate at application layer, time slot allocation at data link layer, and modulation scheme at physical layer through observation of abstracted layer parameters. The cross-layer information exchange in this work is explicit

and accomplished by letting the involved layers communicate with a common cross-layer optimizer through so-called layer abstractions. These serve the purpose of reducing the amount of parameters involved in the optimization in order to reduce the computational demands of the optimization operation.

Kwon et. al. consider sub-carrier allocation in an OFDMA system – 802.16e (WiBro) [36]. They do so according to users' CSI obtained from PHY and they furthermore control adaptive modulation and coding and prioritize users' data streams according to QoS demands from the application layer. In addition, they suggest a protocol for implementing uplink channel sounding and downlink Channel Quality Information (CQI) feedback. Cross-layer information is exchanged between the involved layers through dedicated communication mechanisms between their "control information controller" at the physical layer and "MAC-c controller" at the link layer which also gathers application parameters. Hui et. al. also consider OFDMA resource allocation in [37] where they maximize average total system throughput as a sub-carrier and power allocation problem under delay and queueing constraints, taking CSI and application layer source data rate into account. However, this paper does not describe how to exchange the required information between layers.

Schaar & Shankar explore different aspects in a wireless LAN setting in [38] where they incorporate the three layers and look at selecting the optimal modulation scheme, optimizing power consumption, and optimizing fairness among users, respectively. In [39], Schaar & Tekalp address the complexity of optimizing many layers' parameters jointly and suggest an offline learning-based approach to the joint minimization of distortion, delay, rate, and complexity. The suggested scheme is simulated in a scenario where selection of application layer priorities and MAC layer retransmission limits is based on a training-based classification of low level video content features, channel condition, and maximum available bit-rate. None of these two mentioned papers directly address the method for exchanging the required parameters between the involved layers.

2.3.7 Network/Transport/Application Layers

The so far only work, we have seen, centered around user input is the very recent [40] by Hasswa et. al. Their proposal is to manage primarily vertical handovers[5] based on user-defined preferences from the application layer. These preferences concern cost of service, security, power consumption, network conditions, and network performance. The proposal handles handovers at the transport layer by using SCTP with mobility extensions.

[5]Handovers between networks with different wireless technologies, e.g. WLAN ↔ UMTS.

This deals with handovers through IP multi-homing, i.e. the mobile device has multiple IP addresses registered at which it can be reached (one in each network), performing hand-over by redefining which address is its primary one. The network layer is involved in evaluation of some of the parameters for which the user has defined preferences. It is also involved in detection of the current availability of the different networks and acquisition of addresses within these. The whole framework consists of an application layer part called the "Handover Manager" and a transport layer part called the "Connection Manager". The former deals with the user preferences and controls handover decisions and the latter inter-works with the SCTP protocol modified for this purpose which in turn interacts with the network layer. The two parts of the management framework apparently communicate with each other through an explicit mechanism developed for this purpose.

2.3.8 Overview

The preceding sections of course merely provide a taste of currently ongoing work. There are naturally hundreds of additional references that could not all be covered here. For example, the presented literature does not cover very much of the contributions within sensor and ad-hoc networks.

In relation to Section 2.2, table D.1 gives an overview of the described works on cross-layer design in terms of the type of cross-layer information exchange the respective articles employ. As the table shows, a large portion of the existing and very recent work on cross-layer protocol design does not consider how the actual exchange of information and setting of parameters are to be accomplished. This is mainly due to the fact that many of them have simulated their suggested concepts and not yet considered this aspect. This however underlines that there is a need for research in these practical mechanisms. However, while it may be debatable how to actually implement cross-layer design, researchers should definitely think cross-layer-wise.

Existing	**Dedicated**	**Not considered**
[3–5, 25, 27, 33]	[15, 16, 28, 35, 36, 40]	[14, 17–24, 30–32, 34, 37–39]

Table D.1: Mechanisms for cross-layer information exchange in covered literature.

3 Acknowledgements

This work was partially financed by the Danish government on behalf of the FTP activities within the X3MP project.

References

[1] *Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.* 7498-1. Version 2. ISO/IEC, Nov. 1994.

[2] V. Kawadia and P. R. Kumar. „A cautionary perspective on cross-layer design". In: *IEEE Wireless Communications* 12.1 (2005), pp. 3–11. ISSN: 1536-1284. DOI: 10.1109/MWC.2005.1404568.

[3] A. Ksentini, M. Naimi, and A. Guéroui. „Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross-layer architecture". In: *IEEE Communications Magazine* 44.1 (2006), pp. 107–114. ISSN: 0163-6804. DOI: 10.1109/MCOM.2006.1580940.

[4] G. Wu et al. „Interactions between TCP and RLP in wireless Internet". In: *IEEE Global Telecommunications Conference, 1999. GLOBECOM '99.* Vol. 1B. 1999, 661–666 vol. 1b. DOI: 10.1109/GLOCOM.1999.830139.

[5] P. Sudame and B. R. Badrinath. „On Providing Support for Protocol Adaptation in Mobile Wireless Networks". In: *Mobile Networks and Applications* 6.1 (Jan. 2001), pp. 43–55. DOI: 10.1023/A:1009861720398.

[6] S. Mérigeault and C. Lamy. „Concepts for exchanging extra information between protocol layers transparently for the standard protocol stack". In: *10th International Conference on Telecommunications, 2003. ICT 2003.* Vol. 2. 2003, 981–985 vol.2. DOI: 10.1109/ICTEL.2003.1191572.

[7] M. G. Martini et al. „A Demonstration Platform for Network Aware Joint Optimization of Wireless Video Transmission". In: *IST Mobile Summit 2006.* June 2006.

[8] Q. Wang and M. A. Abu-Rgheff. „Cross-layer signalling for next-generation wireless systems". In: *IEEE Wireless Communications and Networking, 2003. WCNC 2003.* Vol. 2. Mar. 2003, 1084–1089 vol.2. DOI: 10.1109/WCNC.2003.1200522.

[9] J. Inouye, J. Binkley, and J. Walpole. „Dynamic network reconfiguration support for mobile computers". In: *MobiCom '97: Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking.* (Budapest, Hungary). New York, NY, USA: ACM Press, 1997, pp. 13–22. ISBN: 0-89791-988-2. DOI: 10.1145/262116.262122.

[10] G. Carneiro, J. Ruela, and M. Ricardo. „Cross-layer design in 4G wireless terminals". In: *IEEE Wireless Communications* 11.2 (2004), pp. 7–13. ISSN: 1536-1284. DOI: 10.1109/MWC.2004.1295732.

[11] V. T. Raisinghani and S. Iyer. „Cross-layer feedback architecture for mobile device protocol stacks“. In: *IEEE Communications Magazine* 44.1 (2006), pp. 85–92. ISSN: 0163-6804. DOI: 10.1109/MCOM.2006.1580937.

[12] R. Winter et al. „CrossTalk: cross-layer decision support based on global knowledge“. In: *IEEE Communications Magazine* 44.1 (2006), pp. 93–99. ISSN: 0163-6804. DOI: 10.1109/MCOM.2006.1580938.

[13] V. T. Raisinghani and S. Iyer. „Cross-layer design optimizations in wireless protocol stacks“. In: *Computer Communications* 27.8 (May 2004), pp. 720–724. DOI: 10.1016/j.comcom.2003.10.011.

[14] L. Alonso and R. Agustí. „Automatic rate adaptation and energy-saving mechanisms based on cross-layer information for packet-switched data networks“. In: *IEEE Communications Magazine* 42.3 (2004), S15–S20. ISSN: 0163-6804. DOI: 10.1109/MCOM.2004.1273769.

[15] L. Alonso and R. Agustí. „Optimization of wireless communication systems using cross-layer information“. In: *Signal Processing* 86.8 (Aug. 2006), pp. 1755–1772. DOI: 10.1016/j.sigpro.2005.09.029.

[16] E. Kartsakli et al. „A Cross-Layer Scheduling Algorithm for DQCA-based WLAN Systems with Heterogeneous Voice-Data Traffic“. In: *The 14th IEEE Workshop on Local and Metropolitan Area Networks, 2005. LANMAN 2005.* 2005, pp. 1–6. DOI: 10.1109/LANMAN.2005.1541518.

[17] I. Toufik and R. Knopp. „Channel allocation algorithms for multi-carrier multiple-antenna systems“. In: *Signal Processing* 86.8 (Aug. 2006), pp. 1864–1878. DOI: 10.1016/j.sigpro.2005.09.036.

[18] G. Song and Y. Li. „Adaptive resource allocation based on utility optimization in OFDM“. In: *IEEE Global Telecommunications Conference, 2003. GLOBECOM '03.* Vol. 2. 2003, 586–590 Vol.2. DOI: 10.1109/GLOCOM.2003.1258306.

[19] G. Song et al. „Joint channel-aware and queue-aware data scheduling in multiple shared wireless channels“. In: *IEEE Wireless Communications and Networking Conference, 2004. WCNC.* Vol. 3. 2004, 1939–1944 Vol.3.

[20] G. Song and Y. Li. „Utility-based resource allocation and scheduling in OFDM-based wireless broadband networks“. In: *IEEE Communications Magazine* 43.12 (2005), pp. 127–134. ISSN: 0163-6804. DOI: 10.1109/MCOM.2005.1561930.

[21] S. A. Filin et al. „QoS-Guaranteed Cross-Layer Transmission Algorithms with Adaptive Frequency Subchannels Allocation in the IEEE 802.16 OFDMA System“. In: *IEEE International Conference on Communications, 2006*. 2006.

[22] X. Zhang et al. „Cross-layer-based modeling for quality of service guarantees in mobile wireless networks“. In: *IEEE Communications Magazine* 44.1 (2006), pp. 100–106. ISSN: 0163-6804. DOI: 10.1109/MCOM.2006.1580939.

[23] A. El Fawal et al. „Trade-off analysis of PHY-Aware MAC in low-rate low-power UWB networks“. In: *IEEE Communications Magazine* 43.12 (2005), pp. 147–155. ISSN: 0163-6804. DOI: 10.1109/MCOM.2005.1561932.

[24] J. Mišić, S. Shafi, and V. B. Mišić. „Cross-layer activity management in an 802-15.4 sensor network“. In: *IEEE Communications Magazine* 44.1 (2006), pp. 131–136. ISSN: 0163-6804. DOI: 10.1109/MCOM.2006.1580943.

[25] M. G. Martini et al. „Controlling Joint Optimization of Wireless Video Transmission: the PHOENIX Basic Demonstration Platform“. In: *14th IST Mobile & Wireless Communication Summit*. 2005.

[26] PHOENIX. *Jointly optimising multimedia transmissions in IP based wireless networks*. http://www.ist-phoenix.org/index.html.

[27] E. Roddolo et al. „Joint Source and Channel (de)coding in 4G Networks: the Phoenix Project“. In: *The Seventh International Symposium on Wireless personal multimedia communications,* 2004.

[28] C.-C. Tseng et al. „Topology-aided cross-layer fast handoff designs for IEEE 802.11/mobile IP environments“. In: *IEEE Communications Magazine* 43.12 (2005), pp. 156–163. ISSN: 0163-6804. DOI: 10.1109/MCOM.2005.1561933.

[29] IEEE. *Media Independent Handover Services*. http://www.ieee802.org/21/.

[30] G. Liebl, M. Kaindl, and W. Xu. „Enhanced packet-based transmission of multirate signals over GERAN“. In: *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004.* Vol. 3. 2004, 1812–1816 Vol.3.

[31] H. Liu et al. „Channel-Aware Frame Dropping for Cellular Video Streaming“. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2006. ICASSP.* 2006, pp. V–409–V–412.

[32] L. Haratcherev et al. „Optimized video streaming over 802.11 by cross-layer signaling“. In: *IEEE Communications Magazine* 44.1 (2006), pp. 115–121. ISSN: 0163-6804. DOI: 10.1109/MCOM.2006.1580941.

[33] H. Jenkač, T. Stockhammer, and W. Xu. „Cross-layer assisted reliability design for wireless multimedia broadcast“. In: *Signal Processing* 86.8 (Aug. 2006), pp. 1933–1949. DOI: 10.1016/j.sigpro.2005.09.039.

[34] H. Jiang, W. Zhuang, and X. Shen. „Cross-layer design for resource allocation in 3G wireless networks and beyond“. In: *IEEE Communications Magazine* 43.12 (2005), pp. 120–126. ISSN: 0163-6804. DOI: 10.1109/MCOM.2005.1561929.

[35] S. Khan et al. „Application-driven cross-layer optimization for video streaming over wireless networks“. In: *IEEE Communications Magazine* 44.1 (2006), pp. 122–130. ISSN: 0163-6804. DOI: 10.1109/MCOM.2006.1580942.

[36] T. Kwon et al. „Design and implementation of a simulator based on a cross-layer protocol between MAC and PHY layers in a WiBro Compatible.IEEE 802.16e OFDMA system“. In: *IEEE Communications Magazine* 43.12 (2005), pp. 136–146. ISSN: 0163-6804. DOI: 10.1109/MCOM.2005.1561931.

[37] D. S. W. Hui, V. K. N. Lau, and W. H. Lam. „Cross Layer Designs for OFDMA Wireless Systems with Heterogeneous Delay Requirements“. In: *IEEE International Conference on Communications, 2006*. 2006.

[38] M. van der Schaar and S. S. N. „Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms“. In: *IEEE Wireless Communications* 12.4 (2005), pp. 50–58. ISSN: 1536-1284. DOI: 10.1109/MWC.2005.1497858.

[39] M. van der Schaar and M. Tekalp. „Integrated multi-objective cross-layer optimization for wireless multimedia transmission“. In: *IEEE International Symposium on Circuits and Systems, 2005. ISCAS 2005*. 2005, 3543–3546 Vol. 4. DOI: 10.1109/ISCAS.2005.1465394.

[40] A. Hasswa, N. Nasser, and H. Hassanein. „Tramcar: A Context-Aware Cross-Layer Architecture for Next Generation Heterogeneous Wireless Networks“. In: *IEEE International Conference on Communications, 2006*. Vol. 1. June 2006, pp. 240–245. DOI: 10.1109/ICC.2006.254734.