Multistep Detector for Linear ISI-Channels Incorporating Degrees of Belief in Past Estimates

Daniel E. Quevedo, Member, IEEE, Graham C. Goodwin, Fellow, IEEE, and José A. De Doná, Member, IEEE

Abstract—This paper formulates the channel equalization problem in the framework of constrained maximum-likelihood estimation. This allows us to highlight key issues including the need to summarize past data and to apply a finite alphabet constraint over a sliding optimization window. The approach adopted here leads to embellishments of the usual (nonadaptive) decision-feedback equalizer and its multistep extensions. It includes a provision for degrees of belief in past estimates, which addresses the problem of error propagation.

Index Terms—Decision-feedback equalizers (DFEs), equalizers, intersymbol interference, maximum-likelihood (ML) detection.

I. INTRODUCTION

We consider the problem of transmitting a signal drawn from a given finite alphabet over a noisy dispersive channel. The goal is to recover the input to the channel from the received signal in an online fashion.

A common approach to this question is to set the problem up as one of maximum likelihood (ML) sequence detection, which leads to a growing memory structure. In the case of finite impulse response (FIR) channels, the Viterbi algorithm (VA) is often used to solve the resultant optimization problem [1], [2]. The VA uses a finite state machine description of the channel and has a complexity that is exponential in the channel impulse response length. As a consequence, the VA is impractical for channels with long impulse response. With infinite impulse response (IIR) channels, where state variables are continuously distributed, the complexity of the VA is infinite. The complexity of the VA can be reduced by artificially limiting the number of states in the trellis [3]–[5]. The methods obtained, though suboptimal, are practical for a larger number of situations, including IIR channels [4].

Another limitation of the VA, and its reduced complexity variants, resides in the fact that as a consequence of performing sequence detection, in principle, the entire sequence needs to be received to decide upon channel inputs. To provide an algorithm which is applicable in real time, the traceback length of the VA needs to be fixed to a finite number of steps. This fixes the decision delay, often without significant loss of performance [6], [7]; compare also with the approach described in [8]. With this modification, multistep detectors, in which channel inputs are estimated based upon blocks of sampled channel outputs, are obtained. The block size is determined by the traceback length of the (modified) VA. Alternatively, multistep methods can also be developed directly, without using trellis motivations. This is the case for the block decision feedback equalizers (DFEs) presented in [9], and [10]. In general, the performance and complexity of multistep detectors resides between that of the extremely simple DFE [11], [12] and that of the ML sequence detection methods. The block size allows the system designer to tradeoff performance for complexity. It is interesting to note that in the simplest case, where the block size is one, block DFEs reduce to DFES.

Common to (most of) the multistep schemes is the deployment of decision feedback. Here, the information contained in the sampled channel outputs received before the block where the constraints are taken into account explicitly, is summarized by means of an estimate of the channel state. This estimate is based upon previous decisions and is assumed to be equal to the channel state so that it can be used to cancel the intersymbol interference exactly. Thus, decision feedback does not take into account the possibility that previous symbols may be detected with errors and, hence, that the channel state cannot be exactly known. As a consequence, the errors tend to propagate in time, leading to stability problems, which typically degrade performance [13], [14]. As in other engineering problems, feedback structures need to be employed very carefully.

A well-documented way of alleviating the error propagation problem is to develop soft-decision versions of equalization structures. In their basic form, rather than utilizing a hard-decision quantizer to provide the feedback signal, a (monotonic) soft-decision device is used. Outside the loop, soft-decisions are converted into finite-set constrained decisions by means of a standard hard quantizer, which provides the equalizer output. Since the signal fed back is not restricted to belong to a finite set, it may provide information about the reliability of symbols detected. Possible choices of decision devices, which have been utilized to enhance DFEs include hyperbolic tangent functions [15], [16], cubic nonlinearities [17], and piecewise linear (saturation-type) functions [18], [19]. The latter have also been utilized in decoding problems; see earlier work documented in [20] and [21].

In alternative developments of soft-decision structures, reliability measures of decisions can be employed directly in order to allow the decisions already made to be changed a posteriori in the feedback path. To be more precise, in [22], the authors propose that uncertainty concepts be included in a DFE configuration by inspecting the input to the quantizer. If the input
lies near a decision boundary, then the neighbor of the output of the quantizer is used in a parallel DFE loop. After a decision delay, the output of the DFE loop with lower accumulated error is chosen as correct. A different approach is described in [23]. In this latter work, the DFE quantizer is replaced by a Viterbi sequence detector. In this case, rather than deploying symbol-by-symbol feedback, sequences are fed back. These sequences are provided by the Viterbi detector and can be changed after output decisions are made. A somewhat related equalizer has also been proposed in [24], where a Bayesian detector combined with soft-decisions is deployed. Furthermore, in [25], a fractionally spaced equalizer is presented, which combines a hyperbolic tangent decision device with a Kalman filter.

In the present work, we extend the ideas underlying multistep detectors as described, e.g., in [9], [10], and propose a new detector structure. In our approach, the information contained in past data is propagated by means of an a priori state estimate and a measure of its degree of belief. As in [22]–[25], the procedure does not rely upon the assumption that past decisions are correct. Indeed, we permit the a priori state estimate to be replaced by an a posteriori one, once more channel outputs have been processed. This way of incorporating a level of confidence in past data is a distinctive feature of our scheme. It addresses the error propagation problem and leads, in general, to enhanced performance. The resulting scheme, called sceptical multistep detector (SMD), embeds DEEs and the multistep detectors [9], [10] in a broader framework. By utilizing a Gaussian approximation, a fixed complexity structure is obtained, which can be implemented as a closed loop comprised of a vector quantizer and linear filters. For the equalization of FIR channels, the SMD can also be implemented through a trellis-based algorithm, which is related to the VA.

An overview of the remainder of this paper is as follows: The problem of ML detection utilizing an a priori state estimate is formulated in Section II as a finite horizon optimization problem. Section III describes how information contained in past channel output samples can be utilized, leading to the SMD. In Section IV, it is shown how this detector can be implemented as a closed loop. Relationships between the SMD and earlier equalization methods are established in Section V. Examples are presented in Section VI. Section VII draws the conclusions.

II. ML DETECTION UTILIZING AN a priori CHANNEL STATE ESTIMATE

In this section, we will introduce the channel model employed and the concept of multistep detection, which takes into account uncertainty in the initial channel state. This notion will be used in Section III to propose a novel sample-by-sample channel equalization scheme.

A. Channel Model

Consider a discrete-time linear dispersive channel model (which typically includes a noise whitening filter) with scalar\(^1\)

\[ y(t) = C x(t) + w(t) \]

where \( x(t) \) is the input and \( y(t) \) is the output. This model is known as a finite-impulse-response (FIR) channel. The channel impulse response is given by \( h(t) \), which is a discrete-time sequence.

\[ h(t) = \begin{bmatrix} h_0 & h_1 & \cdots & h_N \end{bmatrix}^T \]

and

\[ N \triangleq L_1 + L_2 \]

is the horizon length. Note that the columns of \( \Psi \) contain truncated versions of the impulse response of the channel model.

\(^1\)For ease of exposition, we restrict channel inputs and outputs to be scalar and real-valued. The extension to multiple-input multiple-output and/or complex-valued baseband signals and channels presents no conceptual difficulties.
From (4), it can be seen that the channel output depends upon the channel input, the noise and the channel state. Thus, a detector should incorporate the available knowledge of all these quantities. This can be accomplished as detailed in the sequel.

B. Incorporation of an a priori State Estimate

To develop a multistep detector, which takes into account the fact that past decisions may be incorrect and, hence, that the channel state is uncertain, we incorporate an a priori state estimate into the problem formulation. This is achieved as follows.

We suppose, for the moment, that \( z(k-L_1) \) has a Gaussian distribution

\[
  z(k-L_1) \sim \mathcal{N}(\bar{z}(k-L_1), P)
\]

and that \( \bar{z}(k-L_1) \) is a given a priori estimate for \( z(k-L_1) \). The matrix \( P^{-1} \) reflects the degree of belief in this a priori state estimate. Infinite belief in \( \bar{z}(k-L_1) \) corresponds to setting \( P = 0_0 \), where \( 0_0 \triangleq 0 \cdot I_0 \). Absence of prior knowledge of \( z(k-L_1) \) can be accommodated by using \( P^{-1} = 0_0 \).

The maximum a posteriori (MAP) sequence detector, which at time \( \ell = k \) provides an estimate of the input sequence \( u(k) \) and the channel state \( z(k-L_1) \) based upon the received data contained in \( y(k) \), maximizes the probability density function \(^3\)

\[
  P \left( \begin{bmatrix} u(k) \\ z(k-L_1) \end{bmatrix} \bigg| y(k) \right) = \frac{P \left( y(k) \bigg| \begin{bmatrix} u(k) \\ z(k-L_1) \end{bmatrix} \right) \cdot P \left( \begin{bmatrix} u(k) \\ z(k-L_1) \end{bmatrix} \right)}{P \left( y(k) \right)} \tag{8}
\]

where we have utilized Bayes’ rule.

It is to be noted that only the numerator of this expression influences the maximization. Assuming that \( u(k) \) and \( z(k-L_1) \) are independent (which is a consequence of (1) if \( \{ u(\ell) \} \) is independent), it follows that

\[
  P \left( \begin{bmatrix} u(k) \\ z(k-L_1) \end{bmatrix} \bigg| y(k) \right) = P \left( z(k-L_1) \right) \cdot P \left( u(k) \right).
\]

Hence, if all finite alphabet constrained symbol sequences \( u(k) \) are equally likely (an assumption that we make for the remainder of this paper), then the MAP detector (8) is equivalent to the following ML sequence detector

\[
  \begin{bmatrix}
  \hat{u}(k) \\
  \hat{z}(k-L_1)
  \end{bmatrix} \triangleq \arg \max_{u(k), z(k-L_1)} \left\{ P \left( y(k) \bigg| \begin{bmatrix} u(k) \\ z(k-L_1) \end{bmatrix} \right) \cdot P \left( \begin{bmatrix} u(k) \\ z(k-L_1) \end{bmatrix} \right) \right\}.
\]

Here

\[
  \hat{u}(k) \triangleq [\hat{u}(k-L_1) \ldots \hat{u}(k) \ldots \hat{u}(k+L_2-1)]^T.
\]

In accordance with the restriction \( u(\ell) \in \mathcal{U} \), \( \mathcal{U}^N \triangleq \mathcal{U} \times \ldots \times \mathcal{U} \) needs to satisfy the constraint

\[
  \mathcal{U}(k) \in \mathcal{U}^N, \quad \mathcal{U}^N \triangleq \mathcal{U} \times \ldots \times \mathcal{U} \tag{11}
\]

where \( N \) is defined in (6).

Our working assumption [see (7)] is that the channel state \( z(k-L_1) \) has a Gaussian probability density function \(^4\)

\[
  P \left( z(k-L_1) \right) = \exp \left( -\| z(k-L_1) - \bar{z}(k-L_1) \|^2_{L-1} / 2 \right) / (2\pi)^{N/2} (\det P)^{1/2}.
\]

On the other hand, given the block model (4) and the fact that the channel noise in the discrete-time model (1) is independent identically distributed (i.i.d.) Gaussian with zero mean and variance \( r \), it follows that

\[
  \begin{bmatrix}
  \hat{z}(k-L_1) \\
  \hat{y}(k)
  \end{bmatrix} = \arg \min_{z(k-L_1), \hat{y}(k)} \left( V(u(k), z(k-L_1)) \right) \tag{14}
\]

subject to (11). Here, the cost function \( V \) is defined as

\[
  V(u(k), z(k-L_1)) \triangleq \| z(k-L_1) - \bar{z}(k-L_1) \|^2_{L-1} + \| y(k) - \psi u(k) - \gamma z(k-L_1) \|^2_{R-1} + r^{-1} \sum_{\ell=k-L_1}^{k+L_2-1} (y(\ell) - Cz'(\ell) - Du(\ell))^2
\]

and the vectors \( \{ \hat{x}(j) \} \) denote the predictions of the channel states \( \{ x(j) \} \). These predictions satisfy (1), i.e.

\[
  \hat{x}'(j+1) = A\hat{x}'(j) + Bu(j),
\]

\[
  j \in \{ k-L_1, k-L_1+1, \ldots, k+L_2-1 \}
\]

\[
  \hat{x}'(k-L_1) = z(k-L_1).
\]

We emphasize here that the values of the optimizers \( \hat{u}(k) \) and \( \hat{z}(k-L_1) \) depend upon the constraints imposed on them. While \( \hat{u}(k) \) is finite-set constrained (11), in Section IV, we will utilize two different choices for constraints on the a posteriori state estimate \( \hat{z}(k-L_1) \).

It can be seen that as a consequence of considering the joint probability density (8), the cost (15) includes a term that allows one to obtain an a posteriori channel state estimate \( \hat{z}(k-L_1) \), which differs from the a priori estimate \( \hat{z}(k-L_1) \) as permitted by the confidence matrix \( P^{-1} \). This cost, which alternatively

\(^4\)We use the notation \( ||z||^2 = z^T \Sigma z \) where \( z \) is a vector and \( \Sigma \) is a matrix.
could be motivated directly by setting the problem up as a mean-square detection problem, forms the basis of the SMD to be developed in Section III.

Remark (Notation): Since \( \hat{u}(k) \) and \( \tilde{x}(k - L_1) \) in (9) are calculated using data up to time \( \ell = k + L_2 - 1 \), they could perhaps be more insightfully denoted by \( \hat{u}(k|k + L_2 - 1) \) and \( \tilde{x}(k - L_1|k + L_2 - 1) \), respectively. However, to keep the notation simple, we will here avoid double indexing, in anticipation that the context will always allow for correct interpretation.

III. THE SCETICAL MULTISTEP DETECTOR

Having set up the fixed horizon detection problem as the optimizer (14)–(16), in this section, we describe a way to utilize the information contained in the sampled channel output. To provide a sample-by-sample detector, we will employ moving horizon implementation in conjunction with information propagation through the initial state estimate.

A. Moving Horizon Implementation

Minimization of the cost function \( V \) yields the entire optimizing sequence \( \hat{u}(k) \) defined in (14). However, we will make use of a moving horizon approach in which only the present value\(^5\)

\[
\hat{u}^* (k) \triangleq \begin{bmatrix} 0_{L_1} & 1 & 0_{L_2-1} \end{bmatrix} \hat{u}(k)
\]  

will be delivered at the output of the detector.

At the next time instant, i.e., at \( \ell = k + 1 \), the optimization is repeated providing \( \hat{u}^* (k + 1) \) and so on. Thus, the data window “slides” (or moves) forward in time. The scheme previews \( L_2 - 1 \) samples and, hence, has a decision delay of \( L_2 - 1 \) time units.

Fig. 1 depicts the resultant detector, which is the main contribution of this work. We call it the SMD. As can be seen in Fig. 1, the SMD is a sample-by-sample detector, which provides \( \hat{u}^* (k) \) by processing the block of channel outputs contained in \( \bar{y}(k) \). The contribution of previous channel outputs to present decisions is summarized via the feedback path, which provides the a priori state estimate \( \tilde{z}(k - L_1) \) needed in the decision process (14)–(16).

As in the case of standard multistep detectors, the horizon length \( N \) [see (6)] fixes the complexity of the computations needed in order to minimize (15). It is intuitively clear that good performance of the detector can be ensured if \( N \) is sufficiently large (cf. [26] and compare with the use of a finite traceback length in VA equalizers [12]). However, in practice, there is a strong incentive to use small values for \( L_1 \) and \( L_2 \), since large values give rise to high complexity in the associated computations to be performed at each time step.

The main advantage of the SMD presented here, when compared to earlier schemes, resides in the fact that the pair \( (\tilde{z}(k - L_1), P) \) provides an effective larger horizon than prescribed by \( N \). This aspect is further discussed in Section III-B.

B. Information Propagation Mechanism

The provision of an a priori estimate \( \tilde{z}(k - L_1) \), together with an associated degree of belief via the term \( ||x(k - L_1) - \tilde{z}(k - L_1)||_{P}^{-1} \), in (15), provides a means of propagating the information contained in the data received before \( \ell = k - L_1 \). Consequently, an information horizon of a growing length is obtained. In contrast to the growing memory structure obtained in [8], here the computational effort is fixed by means of the horizon length \( N \).

1) A Priori State Estimate: As depicted in Fig. 1, to propagate information, i.e., to formulate a priori estimates for the channel state, the SMD feeds back estimates of the channel state and input sequence, see (14), (15). We propose to use block propagation according to\(^6\)

\[
\tilde{z}(k - L_1) = A^N \tilde{x}(k - L_1 - N) + M \hat{u}(k - N)
\]  

where \( M \triangleq [A^{N-1}B \ A^{N-2}B \ldots \ AB \ B] \).

In this way, the estimate obtained in the previous block is rolled forward. Indeed, in order to operate in a moving horizon manner, it is necessary to store \( N \) a priori estimates. This way of propagating information is depicted graphically in Fig. 2 for the case \( N = 3 \). It is worth noting that the SMD is a sample-by-sample detector and that there are no parallel decision processes. Coupling between optimization steps merely exists through the feedback path, which provides the a priori estimates (Fig. 1).

Although the associated a priori estimate used in this scheme is decision directed, it should be emphasized here that the mechanism described above is more general than that traditionally employed in decision-feedback schemes, since a degree of belief is associated with the past estimates. Thus, in the SMD, posterior state estimates are allowed to differ from the corresponding a priori estimates. In principle, this idea fits into the

\(^5\)The row vector \( 0_{n} \in \mathbb{R}^{1 \times n} \) contains only zeros.

\(^6\)Since \( \tilde{z}(k - L_1) \) is based upon channel outputs up to time \( k - L_1 - 1 \), it could alternatively be denoted as \( \tilde{z}(k - L_1|k - L_1 - 1) \); see also Remark 1.
framework of the soft-decision schemes described in [22]–[25], where past decisions regarding the channel input are allowed to be changed a posteriori. The principal differences follow from the fact that we propose to reevaluate channel state estimates, rather than detected symbols. Note that since the a priori channel state estimates are constructed based on hard-decisions on \( \hat{u}(k) \) [see (18) and (11)], the SMD is not a soft-decision detector in that sense.

**Remark (Smoothing update)** Rather than propagating information in blocks as in (18), one can also propagate the state estimate via the smoothing update

\[
z(k - L_1) = A\hat{z}(k - L_1 - 1) + B[1 \quad 0_{N-1}] \hat{u}(k - 1),
\]

(19)

Which of the two possibilities (block propagation or smoothing update) gives better performance is problem dependent. The block propagation mechanism tends to error propagation in cyclic patterns (when errors occur), whereas the smoothing update tends to error propagation in adjacent symbols (when errors occur).

2) **Matrix \( P \) as a Design Parameter:** Since channel states depend on the channel input which belongs to \( \mathbb{U} \), one may well question the assumption made in Section II that \( z(k - L_1) \) is Gaussian. (This situation is similar to that of other detectors, which utilize Gaussian forcing approximations [24], [25], [27].) However, we can always use this structure by interpreting the matrix \( P \) in (15) as a design parameter.

As a guide for tuning \( P \), we recall the unconstrained case, where the channel input and initial state are Gaussian with covariances \( Q \) and \( P_0 \), respectively, i.e., \( u(\ell) \sim \mathcal{N}(0,Q), \forall \ell \in \mathbb{N} \) and \( z(0) \sim \mathcal{N}(\Phi_0, P_0) \). In this situation, the Kalman filter provides the minimum variance estimate for \( z(k - L_1) \) [28]–[30]. Its covariance matrix \( P(k - L_1) \) obeys the Riccati difference equation

\[
P(\ell + 1) = AP(\ell)AT - K(\ell)(CP(\ell)CT + r + DQDT)\]

\[
+ K(\ell)^T BQB^T, \quad P(0) = P_0
\]

(20)

where

\[
K(\ell) \triangleq (AP(\ell)CT + BQDT)(CP(\ell)CT + r + DQDT)^{-1}.
\]

A further simplification occurs if we replace the recursion (20) by its steady-state equivalent. In particular, it is well known [30] that under reasonable assumptions, \( P(\ell) \) converges to a steady-state value \( P_R \), as \( \ell \to \infty \). The matrix \( P_R \) satisfies the following algebraic Riccati equation:

\[
P_R = A P_R A^T + BQB^T - (A P_R C^T + BQD^T)\]

\[
(C P_R C^T + r + DQD^T)^{-1}(A P_R C^T + BQ D^T).
\]

(21)

Of course, the Gaussian assumption on the channel input is not valid in the constrained case. However, as will be apparent from the simulation studies included in Section VI, the choice \( P = P_R \) in (15) may still provide good performance.

It should be emphasized that the Riccati equations (20) and (21) only serve for tuning purposes. As will be apparent in Section IV, the detector structure that we are proposing here has nonlinear dynamics, which takes into account the finite-set nature of the channel input and, thus, differs significantly from the Kalman-filter-based equalizer proposed in [27].

As a further embellishment of the SMD, in Section IV, we will develop compact expressions for \( \hat{u}(k) \) and \( \hat{z}(k - L_1) \) that minimize the cost (15) under appropriate constraints.

**IV. CLOSED-LOOP IMPLEMENTATION OF THE SMD**

The optimizers \( \hat{u}(k) \) and \( \hat{z}(k - L_1) \) in (14), which are utilized in the SMD, depend upon the constraints imposed on them.\(^7\) In the present contribution, we focus on the case where \( \{u(\ell)\} \) is drawn from the finite alphabet \( \mathbb{U} \), which motivates the constraint (11). As a consequence, minimizing the cost (15) becomes a nonconvex problem.

In the remainder of this section, we consider two alternatives for the revised (a posteriori) state estimate \( \hat{z}(k - L_1) \), i.e., unconstrained and finite-set constrained. In both cases, a closed-form expression for the solution will be obtained. These expressions utilize a vector quantizer, defined as follows.

**Definition 1 (Nearest Neighbor Vector Quantizer):** Given a countable (not necessarily finite) set of nonequal column vectors \( \mathcal{B} = \{b_1, b_2, \ldots \} \subset \mathbb{R}^n \times 1 \), the nearest neighbor quantizer is defined as a mapping \( g_q; \mathbb{R}^n \times 1 \to \mathcal{B} \), which assigns to each vector \( g \in \mathbb{R}^n \times 1 \) the closest element of \( \mathcal{B} \) (as measured by the Euclidean norm), i.e.,

\[
\{ a \in \mathbb{R}^n \times 1 : \|a - b_i\| \leq \|a - b_j\|, \forall b_j \neq b_i, b_i \in \mathcal{B} \}
\]

\[
\{ a \in \mathbb{R}^n \times 1 : \exists j < i \text{ such that } \|a - b_i\| = \|a - b_j\| \}.
\]

(22)

**A. Revised Initial State Estimate—Unconstrained**

For general IIR recursive channels, it is useful to assume that while the input is always constrained to a finite alphabet, the channel state \( z(\ell) \) in (1) is left unconstrained. In this case, the optimizers (14) are characterized as follows.

**Lemma 1 (Unconstrained Initial State Estimate):** The optimizers corresponding to (14) with the constraint \( \hat{u}(k) \in \mathbb{U}^N \) are given explicitly by

\[
\hat{u}(k) = \Omega^{-1/2}q_{\mathbb{U}} \left( \Omega^{-T/2}(\Lambda_1 \hat{y}(k) - \Lambda_2 \hat{z}(k - L_1)) \right)
\]

(23)

\[
\hat{z}(k - L_1) = (P^{-1} - R^{-1}\Gamma \Psi^{-T} R^{-1}) \Psi \hat{y}(k)
\]

(24)

where

\[
\Omega = \Psi^T \left( R^{-1} - R^{-1} \Gamma \Psi \Gamma^T R^{-1} \right) \Psi
\]

\[
\Psi = \left( P^{-1} + \Gamma^T R^{-1} \Gamma \right)^{-1}
\]

\(^7\)In particular, if \( \{u(\ell)\} \) is considered as a Gaussian sequence, then in the absence of constraints, the Kalman filter can be derived. When \( \{u(\ell)\} \) is allowed to belong to a polytope, then the approach leads to a constrained moving horizon estimator [31]–[33].

\(^8\)Note that the zero measure sets of points, which satisfy (22) with equality, have been arbitrarily assigned to the element having the smallest index. This is done in order to avoid ambiguity in the case of frontier points, i.e., points \( g \in \mathbb{R}^n \times 1 \), which are equidistant from two or more elements of \( \mathcal{B} \) [34].
Fig. 3. Optimization of (14)–(16) following Lemma 1 and leading to the SMD-US.

\[
\begin{align*}
\Lambda_1 &= \Psi^T (R^{-1} - R^{-1} \Gamma \Psi^T R^{-1}) \\
\Lambda_2 &= \Psi^T R^{-1} \Gamma \Psi^T P^{-1}
\end{align*}
\]

and \( \Omega^{1/2} \) is obtained by the factorization \( \Omega = \Omega^{1/2} \Omega^{1/2} \). The nonlinear function \( q_{\Omega^N}(\cdot) \) is the nearest neighbor quantizer as described in Definition 1. The image of this mapping is the set

\[
\tilde{U}^N = \Omega^{1/2} U^N \triangleq \{ \tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_p \} \subset \mathbb{R}^{N \times 1},
\]

with \( \tilde{u}_i = \Omega^{1/2} u_i \), \( u_i \in U^N \).

**Proof:** The proof is given in Appendix A.

Fig. 3 depicts the aforementioned result. By joining Fig. 3 with Fig. 1 and the information propagation mechanism (18), one obtains the SMD with \textit{unconstrained state} decision directed feedback (SMD-US).

The complexity of calculations needed for the SMD-US algorithm to detect each channel input value is essentially proportional to \((n_U)^N\), i.e., to the size of the set \(U^N\). In principle, exhaustive comparisons of \((n_U)^N\) likelihoods need to be performed to find \(\hat{u}(k)\). In practice, the vector quantizer \(q_{\Omega^N}(\cdot)\) in (23) can be implemented through linear operations, followed by scalar quantizers and logical operations [34]. Moreover, efficient binary tree-search methods can be used, and also fast algorithms that yield suboptimal solutions can be conceived.

**B. Revised Initial State Estimate—Constrained**

If the channel (1) is FIR, i.e., if there exists a finite value \(m \in \mathbb{N}\) such that in (2), \(h_j = 0, \forall j > m\), then not only the channel input but also its state can be restricted to belong to a finite set. More precisely, in this case, it is possible to choose \(\eta = m\) and the channel state vector

\[
\varphi(k) = [u(k - 1) \ u(k - 2) \ldots u(k - m)]^T \in U^m.
\]

With this, \(A\) is an \(m \times m\) shift matrix, and \(C\) and \(D\) contain the channel impulse response

\[
A = \begin{bmatrix}
0 & \cdots & 0 \\
1 & \cdots & 0 \\
\vdots & \ddots & \ddots \\
0 & \cdots & 0 \\
0 & 1 & \cdots & 0 \\
\end{bmatrix}, \quad B = \begin{bmatrix}
1 \\
0 \\
\vdots \\
0 \\
0 \\
\end{bmatrix}
\]

\[
C = [h_1 \ h_2 \ldots h_m], \quad D = h_0.
\]

While one can certainly utilize the SMD-US in the FIR case, it is also possible to formulate a detector in which the revised initial channel state is constrained.

**Lemma 2 (Constrained Initial State Estimate):** The optimizer to (14), given the constraints \(\tilde{u}(k) \in \tilde{U}^N\) and \(z(k - L_1) \in U^m\), is

\[
\begin{bmatrix}
\hat{u}(k) \\
\hat{z}(k - L_1)
\end{bmatrix} = \Theta^{-1/2} q_{\tilde{U}^N}(\Theta^{-1/2} \varphi(k))
\]

where \(\Theta, \varphi, \text{ and } \tilde{u}(k)\) are

\[
\Theta = \begin{bmatrix}
\Psi^T R^{-1} \Psi & \Psi^T R^{-1} \Gamma \\
\Gamma^T R^{-1} \Psi & \Gamma^T R^{-1} P^{-1}
\end{bmatrix}, \quad \varphi = \begin{bmatrix}
\Psi^T R^{-1} \\
\Gamma^T R^{-1}
\end{bmatrix}, \quad \tilde{u}(k) = \begin{bmatrix}
u(k) \\
\tilde{z}(k - L_1)
\end{bmatrix}
\]

\(\Theta = \Theta^{1/2} \Theta^{1/2}\), \(q_{\tilde{U}^N}(\cdot)\) is the nearest neighbor quantizer as described in Definition 1 \(\kappa = N + m\), and the finite set \(\tilde{U}^N\) is defined term-wise, as in (26).

**Proof:** The proof is contained in Appendix B.

The resulting SMD with \textit{constrained state} decision directed feedback (SMD-CS) is obtained by bringing together (18) with the circuits of Figs. 1 and 4.

In comparison to the SMD-US described earlier in Section IV-A, here the search set of the vector quantizer has more elements. If \textit{brute-force} exhaustive search was to be used, then \((n_U)^N\) comparisons would be needed to detect each symbol. For channels with long impulse responses, this may be too involved. Fortunately, in the FIR channel case, a trellis-based optimization algorithm can be developed. This idea is explored in Section IV-C.
C. Trellis-Based Optimization Algorithm

FIR channels can be regarded as finite-state machines and are, therefore, amenable to a trellis-based interpretation of the equalization problem [12]. Indeed, the optimization contemplated in Lemma 2 can be carried out by using a trellis, which contains all possible values for each value of

\[ \{ \bar{z}(l) \}, \quad l \in \{ k - L_1, k - L_1 + 1, \ldots, k + L_2 \}. \]

Each of these state values belongs to the set \( \mathbb{U}^m \), whose elements we will denote as \( s_j \), with \( j \in \{ 1, 2, \ldots, (n_U)^m \} \).

Fig. 5 depicts this situation. It illustrates one possible path through the trellis, for the case \( m = 2, U = \{ 0, 1 \} \), tuning parameters \( L_1 = 1, L_2 = 2 \), \( P = I_q \), and with an \( a \) priori estimate \( \hat{z}(k - 1) = [z_1 \ z_2]^T \).

By the principle of optimality “all parts of an optimal path are optimal paths” [35], it follows that only one partial path per state needs to be stored. As a consequence, and mirroring the use of dynamic programming as used in the VA [1], [2], it is possible to derive an efficient implementation for the SMD-CS. More precisely, the following algorithm computes the optimizer \( \hat{u}(k) \) recursively by performing only \( N(n_U)^m \) comparisons, i.e., the computational complexity is exponential in the channel length \( m \) and only linear in the horizon length \( N \) (compare to the SMD-US).

Step 1: Initialize each of the \( (n_U)^m \) state metrics \( J(j) \) and survivor paths \( x_s(j) \) via

\[ J(j) \leftarrow \| s_j - \bar{z}(k - L_1) \|^2 \], \quad x_s(j) \leftarrow \{ \}, \quad \forall j \in \{ 1, 2, \ldots, (n_U)^m \}. \]

Set \( \ell \leftarrow k - L_1 \).

Step 2: For each \( s_j \in \mathbb{U}^m \) and each \( u_i \in \mathbb{U} \), calculate the branch metric \( b(s_j, u_i, y(\ell)) \) defined as

\[ b(s_j, u_i, y(\ell)) \triangleq r^{-1}(y(\ell) - C s_j - D u_i)^2. \]

Step 3: For each \( p \in \{ 1, 2, \ldots, (n_U)^m \} \), set

\[ J(p) = \min_{s_j, u_i} \{ J(j) + b(s_j, u_i, y(\ell)) \} \]

(30)

where \( s_j \in \mathbb{U}^m \) and \( u_i \in \mathbb{U} \) are all \( n_U \) combinations, which satisfy \( s_j = As_j + Bu_i \), i.e., which determine all branches that arrive at \( s_j \). Append the most likely index \( j \), determined through (30), to \( x_s(p) \).

Step 4: If \( \ell < k + L_2 - 1 \), then set \( \ell \leftarrow \ell + 1 \) and go to Step 2.

Step 5: Output the surviving path \( x_s(p^*) \), which corresponds to the index \( p^* \in \{ 1, 2, \ldots, (n_U)^m \} \) having the minimum metric \( J(p^*) \).

The symbolic survivor path provided by this algorithm uniquely determines the optimal state sequence and, due to (27), also the optimizer \( \hat{u}(k) \). Depending on whether the block-updating scheme of Fig. 2 or the smoothing update (19) is utilized, the first or the last element of the survivor path serves to pass on information to another optimization step within the SMD-CS.

It is worth emphasizing that this implementation of the SMD-CS operates over a sequence of length \( N \) and is started at every time step. This is fundamentally different to the usage of the VA as a sequence detector (see [1] or also the algorithms presented in [4] and [3]), where the VA is run only once on the entire channel output sequence.

V. RELATIONSHIP TO OTHER EQUALIZATION METHODS

In this section, we will show how the SMD is related to other equalization schemes. As will be apparent, the main difference to existing sample-by-sample detectors lies in the fact that the SMD does not give an infinite degree of belief in previous decisions.

A. Zero-Forcing DFE

The zero-forcing DFE is a simple nonlinear sample-by-sample detector. It corresponds to a feedback loop comprised of linear filters and a scalar quantizer (Fig. 6). The signal feed back is aimed at removing the intersymbol interference.

The DFE is related to the SMD as follows: The simplest SMD-US is obtained by setting \( L_1 = 0, L_2 = 1, \) and \( P = I_q \) in (15), in which case the detector becomes independent of the channel noise covariance \( r \). The SMD-US is then characterized by information propagation through (17) and (18), i.e., \( \hat{u}^*(\ell) = \hat{u}(\ell) \)

\[ \bar{z}(\ell) = \Gamma \bar{z}(\ell - 1) + B \hat{u}^*(\ell - 1) \]

(31)

and Lemma 1, where the following simplifications can be made

\[ \Gamma = C, \quad \Psi = h_0, \quad \Upsilon = \eta, \quad \Upsilon P^{-1} = I_q, \]

\[ \Omega = 1, \quad \Lambda_1 = h_0^{-1}, \quad \Lambda_2 = h_0^{-1} C, \quad \tilde{U}^N = \Omega. \]

Expressions (23) and (24) give

\[ \hat{u}^*(k) = q(U(h_0^{-1}(y(k) - C \bar{z}(k)))), \quad \bar{z}(k) = \bar{z}(k). \]

For simplicity, \( r = h_0^2 \) is used.
Substitution of (2) and (31) then provides
\[ \hat{u}^*(k) = q_0 \left( h_0^{-1} \left( y(k) - \sum_{i=1}^{\infty} h_i \hat{u}^*(k-i) \right) \right). \]

This structure corresponds to the zero-forcing DFE depicted in Fig. 6.

As a consequence, the zero-forcing DFE is equivalent to the SMD-US with horizon one (i.e., with \( L_1 = 0, L_2 = 1 \)) and with \( P = 0_\eta \). A potential problem associated with choosing \( P = 0_\eta \) is that previous decisions are assumed to be exact and error free. Thus, the DFE does not take into account that estimates are subject to errors. As shown in the examples included in Section VI, this may deteriorate performance. The main problem resides in the propagation of the state estimation errors [13], [19]. On the other hand, the SMD with horizon one and \( P \neq 0_\eta \) addresses this problem and will, in general, give better results than the DFE. It is worth mentioning that enhanced performance is obtained here without significant increase in equalizer complexity. As in the case of a DFE, the SMD with horizon one can be implemented via linear filtering and a scalar quantizer and, thus, has similar computational and memory requirements.

B. Block DFE

Block DFEs have been described in the literature and shown to generalize and outperform conventional DFEs [9], [10]. As with the SMD presented here, block DFEs estimate channel inputs by processing blocks of channel output data. The main difference to the SMD resides in the fact that in the block DFE formulation, past decisions are assumed correct. This leads to a decision-feedback scheme, which may be affected by error propagation.

Not surprisingly, block DFEs are related to the special case of the SMD-US, where \( P = 0_\eta \) in which case the optimizer (14) becomes independent of \( r \). More precisely, the High SNR (p,1)-DFE of [9] (and also System 1 of [10]), are equivalent to the SMD-US with \( P = 0_\eta, L_1 = 0, L_2 = p \), and the smoothing update (19).

As in the case of DFE discussed in Section VI, to address error propagation, it seems desirable not to take \( P = 0_\eta \), and this is indeed the approach that we advocate here. With this modification, the SMD will, in general, outperform block DFEs. The complexity of the SMD, both from a calculation point of view and also with regard to memory requirements, is similar to that of a block DFE with equal horizon length, i.e., with \( p = L_1 + L_2 \) (using the notation of [9]).

Remark 3 (Closed-Form Solution for Block DFE): It is interesting to note that Lemma 1 applies to the aforementioned block DFEs (and also to the \((p,p)\)-DFE described in [9]). In fact, with \( P = 0_\eta, L_1 = 0, \) and \( r = 0 \), the matrices defined in (25) reduce to
\[ \Upsilon = 0_\eta, \quad \Upsilon P^{-1} = I_{\eta}, \quad \Omega = \Psi^T \Psi, \]
\[ \Omega^{1/2} = \Psi, \quad \Lambda_1 = \Psi^T, \quad \Lambda_2 = \Psi^T \Gamma. \] (32)
Expression (24) then yields \( \hat{\varepsilon}(k) = \varepsilon(k) \), and the result (23) provides
\[ \hat{u}(k) = \Psi^{-1} q_{\eta} \left( y(k) - \Gamma \hat{\varepsilon}(k) \right) \]
which is, to the authors knowledge, a novel representation of the algorithms of [9] and [10].

Remark 4 (Decision Regions): At finite SNRs, the optimal MAP detector with a fixed decision delay has decision regions separated by curved hypersurfaces [36], [37], and hence, it is difficult to implement it exactly in practice. The adoption of decision feedback can transform these into hyperplanes, which are easier to implement [9], [38]–[41]. On the other hand, we have shown in Section IV that the SMD in its two forms can be implemented via a closed loop comprised of only linear filtering and a vector quantizer and, therefore, characterized by polyhedral decision regions. As a consequence, both decision feedback and the Gaussian approximation of the a priori state can be regarded as geometric approximations to the optimal signal-space partition of the MAP detector.

C. Delayed Decision-Feedback Sequence Estimation

On the other side of the complexity-performance spectrum lie ML-sequence detectors, such as the VA detector described in [1] and [2]. The VA detector in its basic form is suitable for FIR channels with relatively short impulse responses and outperforms DFEs and block DFEs. To extend the applicability of the VA detector, several reduced complexity VAs have been proposed [3]–[5]. Among these methods, the delayed decision-feedback sequence estimator (DDFSE) introduced in [4] deserves special attention [42]–[45].

The DDFSE utilizes a trellis-based interpretation of the equalization problem and extends the zero-forcing DFE. The rationale behind the DDFSE lies in decomposing the channel model into the sum of an FIR channel of length \( \mu \) and a residual, possibly IIR, channel. Associated with the FIR part is a state vector, which consists of previous channel inputs, compare to (27). The residual channel is summarized via a partial state. The key point to reduce complexity is that in the DDFSE, only the reduced states are used when defining the trellis nodes. The contribution of the residual channel to the branch metrics is approximated via estimates of the partial state at each node. These estimates are constructed based upon the history contained in the surviving paths. Since the surviving paths contain tentative decisions, this information propagation mechanism can be regarded as decision feedback, delayed by \( \mu \) time units.

The parameter \( \mu \) allows the system designer to tradeoff detection performance for complexity of the algorithm. For small \( \mu \), the DDFSE may be affected by error propagation as a consequence of the decision-feedback mechanism [43], [45]. However, since information is propagated through the trellis state metrics, the effect of error propagation is less severe for DDFSEs than for DFEs and block DFEs. Indeed, in the case of minimum phase channels where most of the energy is concentrated in the first \( \mu \) terms of the impulse response—prefiltering may be useful [42], [46]—near-optimal ML performance can often be achieved. As a consequence, the DDFSE will often perform
close to the VA sequence detector and outperform sample-by-sample equalizers such as the DFE, block DFEs, and also the SMD presented here.

Being a sequence detector, in principle, the DDFSE estimates channel inputs only after having received the entire sequence transmitted. Nevertheless, in the case of minimum-phase channels, the decision delay can be reduced by fixing the traceback length to a finite value (akin to what is done in practical implementations of the VA). Fixing the traceback length also reduces memory requirements of the DDFSE, since survivor paths of only fixed, rather than infinitely growing, length need to be stored. The number of calculations per channel input value required by the DDFSE is roughly proportional to \((n\mu)^p\), i.e., it is exponential in \(\mu\). Thus, from a computational complexity point of view, the role of \(\mu\) is comparable to that of the horizon \(N\) within the SMD-US and to the channel length \(m\) in case of the SMD-CS (see Section IV).

VI. EXAMPLES

We study the bit error rate (BER) performance versus the output SNRs of various detectors through simulations. The channel input consists of i.i.d. equiprobable binary (\(U = \{-1, 1\}\)) sequences of length 10 000. The channel noise is i.i.d. Gaussian with zero-mean and variance \(\sigma^2\).

A. One-Pole Channel

First, consider the one-pole IIR channel

\[
H(z) = \frac{z}{z - 0.9} \tag{34}
\]

which was used in [4] as a simple model for an optical recording channel. Fig. 7 depicts the BERs achieved by the SMD-US with block information propagation as in (18). Six designs are included, having parameters \(L_1 = 0\) and \(L_2 \in \{1, 2, 3\}\). In three designs, the matrix \(P\) is chosen according to the algebraic Riccati equation (21) as \(P = P_R\) with \(Q = 1\). In the other designs, the infinite degree of belief case \(P = 0\) is chosen, in which case the SMD-US reduces to DFE (when \(L_2 = 1\)) or to block-DFE (see Section V).

As can be seen in Fig. 7, performance improves with larger horizons. Moreover, the incorporation of a finite degree of belief in past estimates gives lower BERs. We, thus, see that the SMD-US outperforms DFEs and block DFEs by addressing error propagation. It is worth emphasizing here that the performance enhancement obtained by choosing \(P = P_R\) does not require a significant increase in equalizer complexity. In particular, as already mentioned in Section VI, both the DFE and the SMD-US with \(L_1 = 1, L_2 = 0,\) and \(P = P_R\) consist of only linear filtering and a scalar quantizer.

Fig. 8 compares the performance of the SMD-US (with \(P = P_R\)) with a zero-forcing DFE, with a DDFSE (where \(\mu = 3\)), and with the idealized case of a DFE where feedback symbols are error free. While at low SNRs the performace of the SMD-US \((L_2 = 2\) and \(L_2 = 3\)) is similar to that achieved by the DDFSE, at higher SNRs, the sequence detector gives lower BERs. Following the discussion included in Section V-C, both methods have comparable computational complexity, although the memory requirements of the DDFSE are much higher than those of the SMD. All methods are outperformed by the idealized situation of DFE with error-free feedback. However, the latter is a benchmark algorithm, which cannot be implemented in practice.

B. FIR Channel

We next turn our attention to the FIR channel model

\[
H(z) = 0.176 + 0.316z^{-1} + 0.476z^{-2} + 0.532z^{-3} + 0.476z^{-4} + 0.316z^{-5} + 0.176z^{-6} \tag{35}
\]

\[10\log_{10} \left( \sum_{i=0}^{\infty} h_i^2 / \sigma^2 \right)\].

\[\frac{1}{10} \log_{10} \left( \sum_{i=0}^{\infty} h_i^2 / \sigma^2 \right)\].
used in [47]. Fig. 9 depicts the BERs achieved when equalizing this channel with the SMD-CS with horizons $L_1 = 0$ and $L_2 \in \{1, 2, 3\}$, where the matrices $P$ are either set equal to zero, or tuned according to (21) as $P = P_R$ with $Q = 1$. It can be seen from Fig. 9 that performance improves with increasing horizon length and that it is beneficial to incorporate a finite degree of belief in initial estimates.

In Fig. 10, performance of the SMD-CS (with $P = P_R$, $L_1 = 0$, and $L_2 \in \{1, 2, 5, 10\}$) is compared to that of the DFE, and of DDFSE (with $\mu = 3$). As benchmarks, we have also included the results obtained by using a DFE with error-free feedback symbols, and by deploying the optimal VA sequence detector. Not surprisingly, the SMD-CS gives better performance than DFE. Moreover, with horizon length $L_2 = 5$, the performance of the SMD-CS is similar to that of the DDFSE and, with larger horizons, approximates that of the VA sequence detector. We emphasize that as outlined in Section IV-C, the computational complexity of the SMD-CS is only linear in the horizon length and that its memory requirements and decision delays are, in principle, smaller than those of the sequence detectors, such as the VA or the DDFSE.\textsuperscript{11} It is interesting to note that in this case, and unlike the situation examined in Section VI-A, the idealized method of DFE with error-free feedback symbols is outperformed by the SMD-CS (with $L_2 > 2$) and by sequence detectors.

VII. CONCLUSION

This paper has formulated the channel equalization problem as a moving horizon optimization scheme. A key ingredient has been the inclusion of degrees of belief in feedback state estimates. This feature takes into account the fact that past decisions may be incorrect and, thus, addresses the problem of error propagation. Simulation examples have been used to illustrate the relative merits of the two forms of the SMD proposed.

Further work may include the extension of the architecture presented to possibly complex-valued multiple-input multiple-output systems, to blind and semiblind equalization, and to coding problems.

APPENDIX A

PROOF OF LEMMA 1

The cost (15) can be expanded as

$$V(u(k), \hat{z}(k-L_1)) = \|\hat{z}(k-L_1)^2 - 1 + \|\hat{z}(k-L_1)^2 - 1 +$$

$$+ \|\hat{y}(k)^2 - 1 + \|\hat{y}(k)^2 - 1 + u(k)^2 \Psi_{R^{-1}} - u(k)^2 \Psi_{R^{-1}} -$$

$$+ \hat{z}(k-L_1)^2 \Gamma^T R^{-1} \Psi_{u(k)} - 2\hat{u}(k)^2 \Psi_{R^{-1}} \Psi_{u(k)} - 2\hat{u}(k)^2 \Psi_{R^{-1}} \Psi_{u(k)} -$$

$$- 2\hat{z}(k-L_1)^2 (P^{-1} \hat{z}(k-L_1) + \Gamma R^{-1} \hat{y}(k)) \text{ (36)}$$

with $T$ defined in (25). This expression can be written as

$$V(u(k), \hat{z}(k-L_1)) = \alpha(u(k), y(k), \hat{z}(k-L_1))$$

$$+ \|\hat{z}(k-L_1)^2 - 1 + 2\hat{z}(k-L_1)^2 (P^{-1} \hat{z}(k-L_1)$$

$$+ \Gamma R^{-1} \hat{y}(k) - \Gamma R^{-1} \Psi_{u(k)} \text{ (36)}$$

where $\alpha(u(k), y(k), \hat{z}(k-L_1))$ does not depend on $\hat{z}(k-L_1)$.

Since the channel state $\hat{z}(k-L_1)$ is assumed unconstrained, it follows that for every fixed value of $u(k)$, the cost is minimized by choosing $\hat{z}(k-L_1)$ as

$$\hat{z}_{\text{uc}}(k-L_1) = \hat{z}(k-L_1) \hat{z}(k-L_1) + \Gamma R^{-1} \hat{y}(k) - \Gamma R^{-1} \Psi_{u(k)} \text{ (24)}$$

from where (24) follows.

In order to obtain the constrained optimizer $\hat{u}(k) \in \mathbb{C}^N$, observe that

$$\hat{u}(k) = \arg \min_{\hat{u}(k) \in \mathbb{C}^N} J(\hat{u}(k)) \text{ (37)}$$

\textsuperscript{11}In the situation examined, the DDFSE requires less computations per symbol detected than the SMD-CS with large horizons.
where $J(u(k)) \triangleq V(u(k), z^\text{UC}(k - L_1))$. Substitution of (24) into (36) yields

$$J(u(k)) = \beta(y(k), z(k - L_1)) + \|u(k)\|^2_\Omega - 2u(k)^T(\Lambda_1y(k) - \Lambda_2z(k - L_1))$$

(38)

where $\Omega$, $\Lambda_1$, and $\Lambda_2$ are defined in (25), and $\beta(y(k), z(k - L_1))$ does not depend upon $u(k)$.

It is useful to introduce the change of variables

$$\tilde{\mu}(k) \triangleq \Omega^{-1/2} \mu(k).$$

This transforms $\Omega^N$ into $\overline{\Omega}^N$ defined in (26). Equation (38) then allows one to rewrite (37) as

$$\bar{u}(k) = \Omega^{-1/2} \arg \min_{\bar{\mu}(k) \in \overline{\Omega}^N} \tilde{J}(\bar{\mu}(k)) \text{ with } \tilde{J}(\mu(k)) \triangleq \|\mu(k)\|^2_{\overline{\Omega}^N} - 2\mu(k)^T \Omega^{-T/2}(\Lambda_1y(k) - \Lambda_2z(k - L_1))$$

(39)

The level sets of $\tilde{J}$ are spheres in $\mathbb{R}^{N \times 1}$, centered at $\Omega^{-T/2}(\Lambda_1y(k) - \Lambda_2z(k - L_1))$. Hence

$$\arg \min_{\tilde{\mu}(k) \in \overline{\Omega}^N} \tilde{J}(\tilde{\mu}(k)) = q_{\overline{\Omega}^N} \left(\Omega^{-T/2}(\Lambda_1y(k) - \Lambda_2z(k - L_1))\right)$$

which, after substituting into (39), yields (23).

**APPENDIX B**

**PROOF OF LEMMA 2**

The cost (36) can be written in compact form as

$$V(u(k), z(k - L_1)) = \|y(k)\|_{R^{-1}}^2 + \|z(k - L_1)\|_{\varrho^{-1}}^2 - 2 \left[ u(k)^T \varrho(k - L_1) \right] \psi(u(k))$$

(40)

where $\Theta$, $\varrho$, and $\varphi(k)$ are defined in (29). Given the similarity of expressions (38) and (40), the rest of this lemma can be proved as in the second part of the proof of Lemma 1 in Appendix A. For the sake of brevity, the details are omitted.

**ACKNOWLEDGMENT**

The authors would like to thank L. Hanlen, G. Wade, and the anonymous reviewers for their valuable suggestions on this work.

**REFERENCES**


Daniel E. Quevedo (S’97–M’05) received the Ingeniero Civil Electrónico and Magíster en Ingeniería Electrónica degrees from the Universidad Técnica Federico Santa María, Valparaíso, Chile, in 2000, and the Ph.D. degree in electrical engineering from the University of Newcastle, Callaghan, NSW, Australia, in 2005. He is currently with the University of Newcastle, on a Research Academic position. He has lectured at the Universidad Técnica Federico Santa María and the University of Newcastle. His research interests include several areas of automatic control, signal processing, and communications.

Dr. Quevedo received several university-wide prizes during graduation. He received the IEEE Conference on Decision and Control Best Student Paper Award in 2003 and was also a finalist in 2002.

Graham C. Goodwin (M’74–SM’84–F’86) received the B.Sc. degree in physics, the B.E. degree in electrical engineering, and the Ph.D. degree in electrical engineering from the University of New South Wales, Sydney, Australia, in 1964, 1966, and 1970, respectively.

He is currently a Professor of electrical engineering with the University of Newcastle, Callaghan, NSW, Australia, and holds honorary doctorates from Lund Institute of Technology, Lund, Sweden, and the Technion, Haifa, Israel. He is the coauthor of eight books, four edited volumes, and many technical papers.

Prof. Goodwin is the recipient of the Control Systems Society 1999 Hendrik Bode Lecture Prize, the Best Paper Award by the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the Best Paper Award by the Asian Journal of Control, and two Best Engineering Text Book awards from the International Federation of Automatic Control. He is an Honorary Fellow of the Institute of Engineers, Australia, and a Fellow of the Australian Academy of Science, the Australian Academy of Technology, Science and Engineering, and the Royal Society, London. He is a member of the International Statistical Institute and a Foreign Member of the Royal Swedish Academy of Sciences.

José A. De Doná (S’98–M’99) was born in Argentina. He received the B.E. degree in electrical engineering from Universidad Nacional del Comahue, Argentina, in 1989, and the Ph.D. degree in automatic control from the University of Newcastle, Callaghan, NSW, Australia, in 2000.

In 2000, he was a Postdoctoral Researcher with the Universities of Liege and Gent, Belgium. During 2001, he held a Research Academic position with the Centre for Integrated Dynamics and Control, The University of Newcastle. Since 2002, he has been a Senior Lecturer with the School of Electrical Engineering and Computer Science, The University of Newcastle.