Multi-Step Optimal Analog-to-Digital Conversion

Daniel E. Quevedo, Student Member, IEEE,
Graham C. Goodwin, Fellow, IEEE

Abstract

An important aspect of analog-to-digital conversion is the impact of quantization errors. This paper outlines how finite horizon constrained optimization methods can be utilized to design converters which minimize a weighted measure of the quantization distortion. We propose a novel converter, which can be implemented as a feedback loop. It embeds ΣΔ-conversion in a more general setting and typically provides better performance. We also examine the role played by the associated design parameters in ensuring error convergence.

Index Terms

Analog-to-digital (A/D) converter, noise shaping, spectral shaping of errors, sigma-delta modulation, discrete-time systems.

The authors are with the School of Electrical Engineering & Computer Science, The University of Newcastle, Callaghan, NSW 2308, Australia; TEL: +61 2 49217072; FAX: +61 2 49601712; E-MAILS: dquevedo@ieee.org; eegcg@ee.newcastle.edu.au
Multi-Step Optimal Analog-to-Digital Conversion

Corresponding Author:

Daniel E. Quevedo  
School of Electrical Engineering and Computer Science  
The University of Newcastle  
Callaghan, NSW 2308  
Australia

Tel: +61 2 4921 7072  
Fax: +61 2 4960 1712

E-mail: dquevedo@ieee.org
Multi-Step Optimal Analog-to-Digital Conversion

I. INTRODUCTION

When analog signals are digitized unavoidable information loss occurs and the resultant signal will invariably differ from the original, i.e. it will be distorted. In many cases, such as in several digital audio problems [1], or in oversampling applications, it is preferable to minimize the impact of the distortion by *pushing* its frequency content into appropriate regions.

A widely-adopted analog-to-digital converter architecture aimed at addressing these questions is the ΣΔ-Converter in its various forms, see e.g. [2]–[4]. Fig. 1 depicts a sampled data diagram of a general (high-order) single-stage ΣΔ-Converter, as described e.g. in Ch. 3.2 of [3]. In this scheme, the basic quantizer is immersed in a feedback loop which comprises the linear filters $F$ and $G$. A common way of analyzing and designing such a converter is to approximate the quantization errors by white noise which is independent of the signal, see also [5]–[7]. This approximate modeling paradigm leads to a linear quantization model\(^1\). It motivates one to manipulate the frequency content of the quantization noise present in the digital output of the converter by tuning $F$ and $G$.

![Diagram of a general (high-order) single-stage ΣΔ-Converter.](image)

In this paper, we pursue the same general idea of designing an analog-to-digital converter

\(^1\)In some multi-bit applications with high order filters and/or dithering this paradigm has proved to be useful. It fails, however, to predict behaviour such as the appearance of idle tones and instability, which arise particularly when a single-bit quantizer is used. In these cases, non-linear techniques become more appropriate, see e.g. [8].
so as to minimize the impact of quantization in different frequency bands. However, we will
not follow the standard, approximate, line of reasoning referred to above. Instead, we will pose
the problem directly as a weighted $\mathcal{H}_2$ optimization problem. By using Parseval’s Theorem,
we relax the original problem to that of constrained multi-step optimization. We show how
the solution to this problem can be implemented as a feedback loop which embodies a new
analog-to-digital conversion architecture, which we designate as a Multi-Step-Optimal-Converter
(MSOC). It encompasses the $\Sigma\Delta$-Converter as a special case.

The framework adopted here allows us to re-interpret the general $\Sigma\Delta$-Converter as a one-step-
optimal solution to the optimization problem considered. The MSOC extends $\Sigma\Delta$-conversion by
utilizing multi-step optimality and the stability-related concept of error convergence.

This paper builds on early work of the current authors documented in [9], [10]. The novel tech-
nical contribution resides in including pre-filtering and final state weighting in the performance
measure. These enhancements enable us to:

- incorporate Lyapunov-based stability concepts in the design stage of the analog-to-digital
  converter,
- embed several $\Sigma\Delta$-converter structures in the proposed MSOC in precise terms,

An outline of the remainder of the paper is as follows: The frequency weighted conversion
problem is presented in Section II. In Section III the MSOC is developed by using a moving
horizon concept, which here leads to a multi-step quadratic optimization problem. Section IV
shows how the MSOC can be implemented as a closed loop. In Section V we explore, in
detail, how to embed several $\Sigma\Delta$-Converter structures within this framework. We also develop
a multi-step extension to the single-loop case. Section VI presents stability properties of the
resultant converter architecture. Simulation studies are included in Section VII. Section VIII
draws conclusions.

II. FREQUENCY SELECTIVE CONVERSION

Consider a scalar analog continuous-time signal $\tilde{a}_a$, defined over $t \in \mathbb{R}^+$. The purpose of
analog-to-digital conversion is to obtain a discrete-time digital signal $u$. Each value $u(\ell), \ell \in \mathbb{N}$
is restricted to belong to a given finite set of scalars

$$\mathbb{U} = \{s_1, \ldots, s_m\},$$
where \( n_U \) denotes the cardinality of \( U \). The scalar \( u(\ell) \) approximates, in some well defined sense, the sampled signal \( \tilde{a} \) defined via \( \tilde{a}(\ell) \triangleq \tilde{a}_a[\ell \Delta] \), where \( \Delta \) is the sampling period and \( \ell \in \mathbb{N} \).

An immediate solution to the above problem would be to simply approximate each sampled analog value by the nearest element in \( U \), i.e. to set, at sample \( k = \ell \),

\[
u(k) \leftarrow q_U(\tilde{a}(k)),
\]

where \( q_U(\cdot) \) is a quantizer, as defined (in a general vector setting) below.

**Definition 1 (Nearest Neighbour Vector Quantizer):** Consider a countable set of non-equal vectors, each containing \( n_B \in \mathbb{N} \) elements, i.e.: \( B = \{b_1, b_2, \ldots\} \subset \mathbb{R}^{n_B} \).

The nearest neighbour vector quantizer is defined as a mapping \( q_B : \mathbb{R}^{n_B} \rightarrow B \) which assigns to each vector \( c \in \mathbb{R}^{n_B} \) the closest element of \( B \) (as measured by the Euclidean norm), i.e., \( q_B(c) = b \in B \) if and only if \( c \) satisfies:

\[

\|c - b\| \leq \|c - b_i\|, \quad \forall b_i \in B.

\]

(A thorough treatment of quantizers and their features can be found in the book [12]. Results from vector quantization theory have also been applied to the analysis of \( \Sigma\Delta \)-conversion, see [13].)

While the naive decision rule (1) minimizes the overall (unweighted) distortion introduced in the conversion process, this distortion is, in general, highly correlated with the signal \( \tilde{a} \). (It is, indeed, deterministically related.) As a consequence, the spectrum of the distortion can be expected to overlap significantly with that of \( \tilde{a} \). This situation is specially harmful, since it makes it difficult to recover the original analog signal \( \tilde{a}_a \) via filtering.

Our goal here is to optimize an appropriate measure of the distortion errors. Specifically, we will aim at an optimal approximation of the pre-filtered signal:

\[
a \triangleq H \tilde{a},
\]

where \( H \) is a discrete-time filter. Furthermore, we weight the conversion errors via a stable, causal, linear, time-invariant filter \( W \), which can be characterized via\(^2\):

\[
W(\rho) = D + C(\rho I - A)^{-1} B.
\]

\(^2\)Here, and in the remainder of this paper, \( \rho \) denotes the forward shift operator, \( \rho v(k) = v(k + 1) \).
This filter can e.g. represent the typical low-pass filter utilized in oversampled conversion, see e.g. [14] in order to decimate the converter output. In audio applications it makes sense to choose \( W \) as a psycho-acoustic model of the human ear, compare also with work in [9], [15].

We introduce an \( H_2 \) measure of performance:

\[
V \triangleq \frac{1}{2\pi} \int_{0}^{2\pi} |W(e^{j\omega}) (A(e^{j\omega}) - U(e^{j\omega}))|^2 d\omega. \tag{5}
\]

In this measure, \( W(e^{j\omega}) \) denotes the frequency response of the filter \( W \), while \( A(e^{j\omega}) \) and \( U(e^{j\omega}) \) are the (discrete) Fourier transforms of the signals \( a \) and \( u \), respectively. The cost \( V \) penalizes the distortion introduced in the conversion process in a frequency-selective manner.

A converter that minimizes the performance measure \( V \) is certainly appealing. The quantized signal \( u \) would approximate the filtered input \( H\bar{a} \) and the distortion would tend to have a spectrum similar to that of the inverse of the filter \( W \). Unfortunately, it is not possible to directly minimize \( V \) in practical terms by using expression (5), due to the complexity of solving the associated combinatorial optimization problem. Furthermore, the optimal converter would need to pre-view the entire signal \( a \) in order to calculate its Fourier transform. It would therefore be unsuitable for on-line applications.

In the sequel, we will develop a practical approach to solving the above optimal analog-to-digital conversion problem. The goal is to develop a converter which approximately minimizes the performance measure, but is amenable to on-line applications.

### III. THE MULTI-STEP-OPTIMAL CONVERTER

In this section, we will utilize the time-domain counterpart of the performance measure given in Eq. (6) in order to develop a recursive conversion method, which can be implemented on-line. For that purpose we will deploy the concept of moving horizon approximation.

#### A. Time-Domain Formulation

The quadratic performance measure \( V \) can be translated into the time-domain by using Parseval’s theorem. This leads to:

\[
V = \sum_{\ell=0}^{\infty} (e(\ell))^2, \tag{6}
\]
where \( e(\ell) \) are samples of the filtered conversion distortion \( e \), which is the inverse Fourier transform of
\[
E(e^{j\omega}) \triangleq W(e^{j\omega}) (\mathcal{A}(e^{j\omega}) - \mathcal{U}(e^{j\omega}))
\]
and can alternatively be defined via:
\[
e \triangleq W(a - u) = W(H\tilde{a} - u).
\]

Fig. 2 depicts this approach to measuring the efficacy of analog-to-digital conversion. In this figure, \( f_s \triangleq 1/\Delta \) is the sampling frequency.

![Weighted analog-to-digital conversion](image)

Fig. 2. Weighted analog-to-digital conversion.

Naturally, analog-to-digital conversion based on minimization of \( V \) in Eq. (6) suffers from identical shortcomings as conversion based on Eq. (5) and is therefore unsuitable for practical implementation. This is due to the fact that the value of \( V \) depends upon an infinite number of decision variables and also requires that a pre-view of the entire future input \( a \) be available. However, we show below, that we can build on the basic time-domain description (6) to develop a scheme suitable for practical cases.

**B. Finite Horizon Formulation**

In order to develop a practical on-line conversion scheme, it is necessary to restrict the number of decision variables as well as the number of future values of \( a \) utilized in the optimization.
Towards that goal, it is convenient to characterize $e$ as the output of the state space system:

$$x(\ell + 1) = Ax(\ell) + B(a(\ell) - u(\ell))$$

$$e(\ell) = Cx(\ell) + D(a(\ell) - u(\ell)).$$

This relation follows directly from Eq. (4). In Eq. (7), $x \in \mathbb{R}^n$ is the state vector, and $n \in \mathbb{N}$ is the state dimension, i.e. the order of the filter $W$. Note that, due to the Markovian structure of (7), at time $\ell = k$ the impact of the past trajectories of $a$ and $u$ on the future trajectory of $e$ is exactly summarized by means of the present state value, $x(k)$.

With this as a background, we next replace the infinite horizon cost function (6) by the following quadratic cost function defined over a finite horizon $N$:

$$V_N \triangleq \|x'(k + N)\|^2_P + \sum_{\ell=k}^{k+N-1} (e'(\ell))^2,$$

where $N \in \mathbb{N}$ is fixed and $P$ is a given positive semidefinite matrix.

Starting from the current state value $x(k)$ (see Eq. (7)), $V_N$ quantifies a measure of the predicted filtered distortion, $e$, together with the final state, $x(k + N)$. These predicted quantities are formed based upon the model (7), i.e. they satisfy:

$$x'(\ell + 1) = Ax'(\ell) + B(a(\ell) - u'(\ell)), \quad \ell = k, k + 1, \ldots, k + N - 1$$

$$e'(\ell) = Cx'(\ell) + D(a(\ell) - u'(\ell)),$$

with initial condition: $x'(k) = x(k)$.

The finite horizon cost $V_N$ proposed in (8) takes into account only a finite number $N$ of constrained values $u$. These decision variables can be grouped into the vector

$$\bar{u}(k) \triangleq \begin{bmatrix} u'(k) & u'(k + 1) & \ldots & u'(k + N - 1) \end{bmatrix}^T.$$

In order to make this dependence explicit, we will write $V_N(\bar{u}(k))$. Since the converter output signal is finite-set constrained, so is $\bar{u}(k)$, i.e.

$$\bar{u}(k) \in \mathbb{U}^N,$$

where $\mathbb{U}^N \subset \mathbb{R}^N$ is defined via the Cartesian product: $\mathbb{U}^N \triangleq \mathbb{U} \times \cdots \times \mathbb{U}$.

In the above formulation, only the elements contained in $\bar{u}(k)$ are finite-set-constrained. Thus, we will call $N$ the constraint horizon. The value of $N$ fixes the complexity of the calculations.

$\|x\|^2_P$ denotes the quadratic form $x^T P x$, where $x$ is any vector and $P$ is a matrix.
needed in order to minimize $V_N(\bar{u}(k))$. (Compare with the infinite decision variables in the original cost $V$). It also reduces the required pre-viewing of $a$ to $N - 1$ samples. Since $N$ is left free to the designer, it can be chosen so that the minimization can be carried out on-line.

C. Moving Horizon Approach

Minimization of the multi-step cost introduced in eq. (8) gives rise to the optimizer

$$\bar{u}^*(k) \triangleq \arg \min_{\bar{u}(k) \in \mathbb{U}^N} V_N(\bar{u}(k)),$$

which contains $N$ elements. Thus, in principle, one could think of an implementation *in blocks*, where the optimization is carried out every $N$ sampling instants. Unfortunately, the last few elements of $\bar{u}^*(k)$ depend only on a small window of the filtered distortion, $e$. Thus, to enhance performance we propose to utilize only the first element of $\bar{u}^*(k)$, namely:

$$u^*(k) \triangleq \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix} \bar{u}^*(k).$$

(11)

It is this quantity, whose effect on the entire signal $e$ is, in principle, best captured with the $N$ predicted values contained within the horizon examined by the finite horizon cost $V_N(\bar{u}(k))$.

The *scalar* $u^*(k)$ is delivered to the output of the converter, by setting:

$$u(k) \leftarrow u^*(k)$$

(12)

and is also utilized to update the state according to Eq. (7), i.e. we set:

$$x(k + 1) = Ax(k) + B(a(k) - u^*(k)).$$

(13)

At the next sampling instant, this new state value is used to minimize the cost $V_N(\bar{u}(k + 1))$, yielding $u(k + 1)$. This procedure is repeated *ad-infinitum*. As illustrated in Fig. 3 for the case $N = 5$, the constraint horizon of the criterion $V_N(\bar{u}(k))$ moves (slides) forward as $k$ increases. The past is propagated forward in time via the state sequence $x$, thus, yielding a recursive scheme.

The resultant architecture defines the *Multi-Step-Optimal-Converter* (MSOC), which is of principal interest in the present work. It constitutes an analog-to-digital converter architecture which optimizes the frequency weighted conversion distortion.

It is easy to see that, in general, larger values for $N$ provide better performance, since more data is taken into account in the bit allocation process. In fact, one can expect that, if $N$
is chosen large enough relative to the time scale of $W$, then the effect of $u(k)$ on $e(\ell)$ for $\ell \geq N$ will be negligible and the performance of the MSOC will approach that obtained, if the infinite horizon measure of Eq. (5) or (6), were to be minimized directly (which, for the reasons explained above, is impractical.) This asymptotic behaviour is confirmed by simulation studies, see e.g. Section VII. Indeed, we will see that modest values of $N$ give near asymptotic optimal performance, thus rendering the scheme quite easy to implement in practical cases.

In summary, the constraint horizon, $N$, allows the designer to trade-off performance versus on-line computational effort and, as will be demonstrated in Section VII, excellent performance can often be achieved with relatively small horizons.
Remark 1 (Relationship to Model Predictive Control): The moving horizon idea advocated above, mirrors the strategy used in Model Predictive Control schemes, see e.g. [16] or [17, Chap. 23]. Model Predictive Control has actually been a powerful tool for dealing with constraints in the design of control systems [16]–[20].

IV. IMPLEMENTATION OF THE MSOC AS A FEEDBACK LOOP

The MSOC described in the previous section requires that one solve the finite-set-constrained quadratic optimization problem (10). In this section, we will develop a solution to this problem, which will allow us to implement the MSOC as a closed loop. Towards this end, it is convenient to define the matrices:

\[
\Gamma = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix}, \quad \Phi = \begin{bmatrix} w_0 & 0 & \ldots & 0 \\ w_1 & w_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ w_{N-1} & \ldots & w_1 & w_0 \end{bmatrix},
\]

\[
M = \begin{bmatrix} A^{N-1}B & A^{N-2}B & \ldots & AB & B \end{bmatrix}.
\]

In the above definition,

\[
w_0 = D, \quad w_i = CA^{i-1}B, \quad i = 1, 2, \ldots, N - 1,
\]

see Eq. (4). Thus, the columns of \( \Phi \) are truncated impulse responses of the filter \( W \).

The following result gives a dynamical system whose output is the solution to the non-convex optimization problem (10)–(12).

Theorem 1 (Closed Form Solution): Suppose \( \mathbb{U}^N = \{v_1, v_2, \ldots, v_r\} \), where \( r = (n_u)^N \) and \( W \) has realization (4), then the output of the MSOC, see Eq. (12), satisfies:

\[
u(k) = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix} \Psi^{-1} \tilde{q}_{\mathbb{U}^N} (\mathcal{W}(\rho)H(\rho)\tilde{a}(k) - \mathcal{F}(\rho)u(k)).
\]

In this expression, the transfer functions \( \mathcal{W} \) and \( \mathcal{F} \) are defined via:

\[
\mathcal{W}(\rho) \triangleq \Psi \begin{bmatrix} 1 & \rho & \ldots & \rho^{N-1} \end{bmatrix}^T + \Psi^{-T}(\Phi^T \Gamma + M^T PA^N)(\rho I - A)^{-1} B
\]

\[
\mathcal{F}(\rho) \triangleq \Psi^{-T}(\Phi^T \Gamma + M^T PA^N)(\rho I - A)^{-1} B,
\]

each having 1 input and \( N \) outputs. The matrix \( \Psi \) is square and defined implicitly via:

\[
\Psi^T \Psi \triangleq \Phi^T \Phi + M^T PM.
\]
The nonlinearity $q_{\mathcal{U}}(\cdot)$ is the nearest neighbour quantizer described in Definition 1. The image of this memoryless mapping is the set:

$$\mathcal{U} \triangleq \{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_r\} \subset \mathbb{R}^N,$$

with $\bar{v}_i = \Psi v_i$, $v_i \in \mathcal{U}^N$. \hfill (19)

**Proof:** The proof is included in Appendix I.

As a consequence of this Theorem, we can implement the MSOC as the feedback loop depicted in Fig. 4, where we have utilized thick lines to denote vector-signal paths.

Note that the complexity of the calculations which need to be performed on-line is mainly determined by the vector quantizer $q_{\mathcal{U}}(\cdot)$. It therefore depends upon the cardinality of the search-set $\mathcal{U}^N$, which is equal to $(n_{\mathcal{U}})^N$.

\[ \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \Psi^{-1} = \begin{bmatrix} 1/w_0 & 0 & \cdots & 0 \end{bmatrix}. \hfill (20) \]

Furthermore, the transfer function matrices defined in Eq. (17) simplify to:

$$W(\rho) \triangleq \Phi \begin{bmatrix} 1 & \rho & \cdots & \rho^{N-1} \end{bmatrix}^T + \Gamma (\rho I - A)^{-1} B, \quad \mathcal{F}(\rho) \triangleq \Gamma (\rho I - A)^{-1} B. \hfill (21)$$

Note that in this case, $W$ and $\mathcal{F}$ depend only on the impulse response of $W$ and are therefore independent of the particular realization chosen in Eq. (4).

Fig. 4 obviously bears some similarities to the $\Sigma\Delta$-Converter depicted earlier in Fig. 1. Indeed, as will be shown in the following section, the MSOC may be regarded as a generalization of the $\Sigma\Delta$-architecture.
V. RELATIONSHIP TO $\Sigma\Delta$-CONVERSION

The MSOC has been developed having distortion spectra in mind. Perhaps not surprisingly, it encompasses, as a special case, several $\Sigma\Delta$-converter structures, which also aim to shape the quantization noise. In this section, we will first show how the general high-order single-stage $\Sigma\Delta$-Converter can be immersed into the proposed architecture. This insight will allow us to develop, as a special member of the MSOC-family, a multi-step-optimal extension to the widespread Single-loop $\Sigma\Delta$-Converter.

A. General Filter

A particularly simple case of the MSOC is obtained when a unitary constraint horizon, $N = 1$, is employed with no final state weighting (i.e. $P = 0$ in Eq. (8)). Next, consider a filter $W$ with unitary feed-through, i.e. with $D = 1$ in Eq. (4). In this case, it follows from Eqs. (14) and (18) that $\Gamma = C$ and $\Phi = \Psi = 1$. Hence, the set $\tilde{U}^N$ simplifies to $\tilde{U}$ and the vector quantizer reduces to a standard scalar quantizer. Eqs. (16), (20) and (21) then show that the output of the resultant, one-step-optimal analog-to-digital converter satisfies:

$$u(k) = q_{\tilde{U}}(W(\rho)H(\rho)\tilde{a}(k) - (W(\rho) - 1)u(k)).$$

This special case is depicted in Fig. 5. By comparing this diagram with that of the general (also termed high order) $\Sigma\Delta$-converter presented earlier in Fig. 1, we may conclude that the special case of the MSOC described above is equivalent to the latter, if $W$ and $H$ are chosen according to:

$$W = 1 + FG, \quad H = \frac{G}{1 + FG}. \quad (22)$$

![Fig. 5. One-step-optimal converter without final state weighting.](image-url)
Thus, the general single-stage $\Sigma\Delta$-Converter of Fig. 1 can be embedded as a special case of the more general frequency weighted analog-to-digital conversion problem. As opposed to existing approaches, see e.g. [7], we can do this in precise terms, without having to adopt a white-noise model. In particular, we see that the $\Sigma\Delta$-conversion scheme precisely implements the **horizon-one solution** to (16) as a closed loop.

Different filters $W$ and $H$ allow one to recover specific $\Sigma\Delta$-Converter structures. In particular, Table I includes the choices which lead to the standard first-order Single-loop $\Sigma\Delta$-Converter$^4$, see e.g. [11], [21]–[24], the second-order Double-loop $\Sigma\Delta$-Converter of [25] (in [26]–[33] particular cases within that structure are considered), the multi-loop architecture described in [34] and bandpass conversion, as described e.g. in [35]–[37]. (In this table we have characterized the converters using the standard symbols used elsewhere in the literature.)

It is worth emphasizing here that, besides the converters included in Table I, other schemes such as, e.g. the high order $\Sigma\Delta$ converters analyzed in [38], [39] can also be included within our setting.

### Table I

**Specific Filter Choices**

<table>
<thead>
<tr>
<th></th>
<th>$H(\rho)$</th>
<th>$W(\rho)$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single-loop</strong></td>
<td>$\rho^{-1}$</td>
<td>$\frac{\rho}{\rho - 1}$</td>
</tr>
<tr>
<td><strong>Double-loop</strong></td>
<td>$\frac{Gg \rho}{\rho^2 + ((BG + b)g - \alpha - \beta)\rho + (\alpha - bg)\beta}$</td>
<td>$\frac{\rho^2 + ((BG + b)g - \alpha - \beta)\rho + (\alpha - bg)\beta}{(\rho - \alpha)(\rho - \beta)}$</td>
</tr>
<tr>
<td><strong>Multi-loop</strong></td>
<td>$1$</td>
<td>$(\frac{\rho}{\rho - 1})^L$</td>
</tr>
<tr>
<td><strong>Bandpass</strong></td>
<td>$\frac{2\tau \cos(\theta)\rho - r^2}{\rho^2 - 2(1 - g)\tau \cos(\theta)\rho + (1 - g)r^2}$</td>
<td>$\frac{\rho^2 - 2(1 - g)\tau \cos(\theta)\rho + (1 - g)r^2}{\rho^2 - 2\tau \cos(\theta)\rho + r^2}$</td>
</tr>
</tbody>
</table>

From the above development, it is apparent that the MSOC captures standard $\Sigma\Delta$-architectures as a special case and also generalizes this paradigm. Indeed, by simply writing the filters $W$

---

$^4$Although this filter is only marginally stable, we can still deploy our framework by using the cost function (8) directly as a starting point.
contained in Table I, in the form (4) and choosing \( N > 1 \) we can immediately develop multi-step-optimal extensions to these converters. Moreover, in case of the MSOC, a nonzero final state weighting term \( \| x'(k + N) \|_P^2 \) can also be chosen, thus giving extra design flexibility.

The main advantage of using a horizon larger than 1 derives from the fact that, with \( N > 1 \), not only the present, but also future values of \( \bar{a} \) are taken into account in the conversion process. Also, choosing \( P \neq 0 \) may enhance stability properties, see Section VI. As a consequence, and as illustrated in Section VII, the MSOC will in general outperform existing \( \Sigma \Delta \)-conversion schemes.

Remark 2 (Other Vector Quantizer Converters): It should be emphasized here that the MSOC differs from other converter structures which deploy vector quantizers. In particular, in [40], \( \Sigma \Delta \)-conversion is generalized to two-input systems, which arise in power electronics applications. The resultant scheme incorporates a hexagonal quantizer, which partitions its two dimensional input space into 7 regions. Alternatively, in [41], the authors propose to implement multi-bit high-order \( \Sigma \Delta \)-converters (see Fig. 1), with a vector quantizer. However, as shown above, these cases are only one-step optimal when viewed from the MSOC perspective. The latter scheme, as described here, allows for multi-step-optimality.

B. The Single-Loop MSOC

The first-order Single-loop \( \Sigma \Delta \)-converter included in the first row of Table I deserves special attention. It is commonly characterized via the feedback loop depicted in Fig. 6 and has been studied extensively, see e.g. [11], [21]–[24].

![Fig. 6. Single-loop \( \Sigma \Delta \)-Converter.](image)

Beyond analog-to-digital conversion, this structure has also been utilized in several other applications areas, such as discrete coefficient FIR filter design [42], [43], Switch-mode power-supplies [44] and control of solenoid valves for odor blending [45].
1) Arbitrary horizon: In the first order case, \( W(\rho) = \rho(\rho - 1)^{-1} \) one can use the realization (4) with \( A = B = C = D = 1 \). Thus, for arbitrary constraint horizons, the matrices defined in Eq. (14) take the form:

\[
\Phi = \begin{bmatrix}
1 & 0 & \ldots & 0 \\
1 & 1 & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
1 & \ldots & 1 & 1
\end{bmatrix}, \quad \Gamma = \begin{bmatrix}
1 \\
\vdots \\
1
\end{bmatrix}.
\]

If we further restrict \( P = 0 \), then, following Corollary 1, we can set \( \Psi = \Phi \) and obtain that:

\[
\mathcal{W}(\rho) = \Phi \left[ 1 \quad \rho \quad \ldots \quad \rho^{N-1} \right]^T + \frac{1}{\rho - 1} \left[ 1 \quad \ldots \quad 1 \right]^T
\]

\[
\mathcal{F}(\rho) = \frac{1}{\rho - 1} \left[ 1 \quad \ldots \quad 1 \right]^T.
\]

Direct calculation yields:

\[
\Phi \begin{bmatrix}
1 \\
\rho \\
\rho^2 \\
\vdots \\
\rho^{N-1}
\end{bmatrix} = \begin{bmatrix}
1 \\
1 + \rho \\
1 + \rho + \rho^2 \\
\vdots \\
1 + \rho + \ldots + \rho^{N-1}
\end{bmatrix} = \frac{1}{1 - \rho} \begin{bmatrix}
1 - \rho \\
1 - \rho^2 \\
1 - \rho^3 \\
\vdots \\
1 - \rho^N
\end{bmatrix},
\]

and Eq. (16) provides:

\[
u(k) = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix} q_{\mathcal{B}N}(d(k)), \tag{23}\]

with:

\[
d(k) \triangleq \frac{1}{\rho - 1} \left( \left[ 1 \quad \rho \quad \ldots \quad \rho^{N-1} \right]^T \tilde{a}(k) - \left[ 1 \quad \ldots \quad 1 \right]^T u(k) \right). \tag{24}\]

The resultant scheme can therefore be implemented as in Fig. 7. It is easy to see, that with \( N = 1 \), this scheme reduces to the standard single-loop converter of Fig. 6. It corresponds to an \( N \)-step-optimal generalization of the single-loop case. We will call it the Single-loop MSOC and note that it is a particular case of the MSOC.

2) Horizon two: It is worthwhile exploring the choice \( N = 2 \) in greater detail. In this case, and with \( \mathbb{U} = \{-1, 1\} \), we obtain that:

\[
\mathbb{U}^2 = \left\{ \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}, \quad \bar{\mathbb{U}}^2 = \left\{ \begin{bmatrix} -1 \\ -2 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}.
\]
The vector quantizer $q_{\mathbb{R}^2}(\cdot)$ which defines the conversion rule (23) partitions its input space, $\mathbb{R}^2$, into four regions according to the nearest neighbour rule (2), see e.g. [12]. Since we are only interested in the first element of $\bar{u}^*(k)$, see Eq. (11), only two regions are of significance. These are shown in Fig. 8. Depending upon past converter inputs and outputs, the vector $d(k)$ in (24) will adopt different values. The converter output, $u(k)$, is then given by:

$$u(k) = \begin{cases} 
-1, & \text{if } d(k) \in \mathcal{D}_1 \\
+1, & \text{if } d(k) \in \mathcal{D}_2
\end{cases}$$

Geometrical arguments allow us to describe the partition of Fig. 8 by means of:

$$u(k) = q_{\mathbb{U}} \left( q_{\mathbb{U}}(d_1(k) + d_2(k) - 1) + 2(q_{\mathbb{U}}(d_1(k)) + q_{\mathbb{U}}(d_1(k) + d_2(k) + 1)) \right).$$
As a consequence, the vector quantizer $q_{\mathbb{R}^2}(\cdot)$ can be realized with 4 standard scalar quantizers $q_{\mathbb{R}}(\cdot)$, which operate on auxiliary scalar signals. The corresponding diagram of the entire two-step-optimal Single-loop converter can be appreciated as in Fig. 9.

![Fig. 9. Diagram of the two-step-optimal Single-loop converter.](image)

**Remark 3 (Leakage in the integrator):** In practical applications, it is often the case that the integrator utilized in single-loop $\Sigma\Delta$-Converters is not ideal but is affected by leakage and has a non-unity feed-forward gain. As described in [46]–[48], this phenomenon can be modeled by using a first order filter, with gain $g$ and pole located at $0 < \alpha < 1$. We note that, although these are basically undesired phenomena, their effect can be explicitly taken into account within our framework. In particular, one can choose

$$H(\rho) = \frac{g}{\rho + g - \alpha}, \quad W(\rho) = 1 + \frac{g}{\rho - \alpha}$$

and follow similar lines as those described above. (More details on modeling and simulation of non-ideal $\Sigma\Delta$-converters can be found e.g. in [49]).

VI. ACHIEVING STABILITY THROUGH FINAL STATE WEIGHTING

It is well known that systems, such as the ones described above, incorporating quantization in the feedback loop can exhibit highly non-trivial dynamic behaviour. In particular, the case of $\Sigma\Delta$-conversion has been studied extensively, see e.g. [8], [21], [22], [24], [39], [50]–[53]. It turns out, that the distortion introduced by a $\Sigma\Delta$-converter is deterministically related to the input and depends upon its amplitude and frequency. Even in idle-state (i.e. with no input) the
output stream may exhibit tones and, in general, it cannot be guaranteed that conversion errors will decay.

One way to overcome this stability problem in ΣΔ converters resides in utilizing a dithering signal, see e.g. [3], [6], [54]–[56]. This inclusion of an exogenous noise at the input to the non-linearity makes trajectories behave more random, and become less periodic. As a consequence, tone-like behaviour can be avoided but at the expense of a greater noise floor.

For the case of the MSOC considered here, and in light of the relationship established in Section V, one may expect that this scheme will also exhibit complex dynamic behaviour. Indeed, the results of [57] and of [58] regarding piecewise-linear mappings can be applied to the MSOC loop of Fig. 4. Inter alia, this predicts the possibility of attractive periodic orbits.

By extending our work related to the Moving Horizon Optimal Quantizer described in [9] it is possible to include random dithering in the MSOC, which then can be analyzed following techniques described e.g. in [59]. Alternatively, and unlike the case of ΣΔ-conversion and the Moving Horizon Optimal Quantizer, the MSOC can be stabilized without any architectural modifications. To achieve this goal, the final weighting $P$ will turn out to play a pivotal role as detailed in the following Theorem.

**Theorem 2 (Error Convergence):** Suppose that the signal $a$ is such that it can be eventually quantized, i.e. there exists a finite value $k$ such that $a(\ell) \in U, \forall \ell \geq k$. Then, if $P$ is chosen to satisfy the Lyapunov Equation

$$A^T P A + C^T C = P,$$

(25)

then $e(\ell) \to 0$, as $\ell \to \infty$.

**Proof:** The proof is included in Appendix II. 

It is worth emphasizing here that this result applies to the general case of the MSOC with any (stable) high-order filter $W$ and any finite set $U$. In particular, if $P$ is chosen as a solution to Eq. (25) and $0 \in U$, then the output of any member of the MSOC-family exhibits no idle-tones. This is in stark contrast to the bulk of the work available in the existing literature, see e.g. [8],

---

The stabilizing effect of final state weighting is known in other contexts. For example, in Model Predictive Control schemes having saturation type constraints [18] it is known that a final state weighting is important to establish stability. However, the choice of $P$ in the finite set constrained case is considerably more difficult and, to the best of our knowledge, has not been addressed, save for our own work in [20]. The latter paper deals with the stability of a related dynamical system when the input is removed. Here, we extend the result to the more practical case of non-zero inputs.
which aims at establishing stability related concepts only for specific $\Sigma\Delta$-converter choices.

Although Theorem 2 is of significance only for input signals which give rise to an eventually quantized sequence $a$, simulation studies indicate that setting $P$ as the positive definite solution to the Lyapunov Equation (25) is, in general, a good choice. As shown in Section VII-B below, it helps to avoid limit cycles and tonal distortion. Thus, the MSOC provided with the stabilizing matrix $P$ given in (25) can be regarded as a deterministically dithered converter. Unlike other such converter architectures, see e.g. the bit-flipping technique of [62], [63], the adaptive integrator-output bounding converter [64] or the stabilizing $\Sigma\Delta$-conversion scheme of [65], the MSOC is a non-adaptive time-invariant structure, and is, therefore, easier to implement.

Remark 4 (Infinite horizons): The choice for the final state weighting matrix suggested by Theorem 2 can be related to the infinite horizon measure $V$ in Eq. (6) by using standard results related to Lyapunov stability theory, see e.g. [66]. More precisely, it follows that, if $u'(\ell)$ is equal to $a(\ell)$, for every $\ell \geq k + N$ and $P$ satisfies (25), then:

$$\|x'(k + N)\|_P^2 = \sum_{\ell=k+N}^{\infty} (e'(\ell))^2,$$

(26)

where the terms in the sum are subject to the filter dynamics as in Eq. (9), starting at $x'(k + N)$.

Despite the fact that the finite set constraints preclude $u'(\ell)$ to be equal to $a(\ell)$ so that Eq. (26) can only hold approximately, this expression allows one to interpret Theorem 2 as a stability result based upon infinite horizon optimality. Viewed from that angle, one can think of the term $\|x'(k + N)\|_P^2$ as a mechanism for summarizing the infinite horizon behaviour. 

VII. SIMULATION RESULTS

In order to illustrate characteristics of various members of the MSOC-family, we will measure the performance using an input signal $\tilde{a}(\ell)$ defined over $\ell = 1, 2, \ldots, T_f$ via the sample variance of the distortion defined as:

$$S \triangleq \frac{1}{T_f} \sum_{\ell=1}^{T_f} (H(p)\tilde{a}(\ell) - u(\ell))^2$$

(27)

and by means of the sample variance of the filtered distortion:

$$S_f \triangleq \frac{1}{T_f} \sum_{\ell=1}^{T_f} (e(\ell))^2,$$

(28)

value, which is related to the cost (6).
A. The Single-Loop MSOC

In what follows we will analyze the behavior of the Single-loop MSOC described in Section V-B (see Fig. 7) with the binary alphabet $\mathbb{U} = \{-1, 1\}$.

1) Periodic Inputs: We first utilize a periodic input signal of the form:

$$\tilde{a}(\ell) = (10\pi)^{-1} + A \sin (2\pi f \ell), \quad \ell = 1, 2, \ldots, T_f = 5000,$$

which corresponds to a sampled sinusoid with a small offset. Its amplitude is $A$ and its frequency is equal to $f$ times the sampling frequency $f_s$. Thus, $f$ denotes relative frequency. If the Nyquist criterion is to be satisfied, then $f$ should lie in the range $0 \leq f < 0.5$. Having in mind oversampling applications, we restrict $f$ further.\(^7\)

Fig. 10 illustrates the effect of the input frequency $f$ for a fixed amplitude, $A = 2$. It shows the values of $S$ defined in (27) for the case of Single-loop $\Sigma\Delta$-conversion as in Fig. 6 and for the Single-loop MSOC with horizons $N = 2$ and $N = 3$. It can be seen in the figure that the MSOC introduces less distortion than the $\Sigma\Delta$-Converter. Moreover, choosing $N = 3$ outperforms the smaller constraint horizon, $N = 2$. These results are confirmed in Fig. 11, which shows the effect of the input amplitude, $A$. Again, the MSOC schemes outperform the $\Sigma\Delta$-Converter, although the effect becomes significant only at larger amplitudes.

2) Random Inputs: Next, we examine the case of a random input. For that purpose, we choose $\tilde{a}$ to be a white noise process, where each observation $\tilde{a}(\ell), \ell = 1, \ldots, T_f = 10^4$ is uniformly distributed in the interval $[-2, 2]$.

Typical power spectral densities of the resultant (unfiltered) distortion, $a - u$, for the Single-loop $\Sigma\Delta$-Converter and the Single-loop MSOC with $N = 10$ are included in Fig. 12. As might have been anticipated, while both schemes shape the quantization effect, the overall distortion level is lower in case of the Single-loop MSOC. The performance improvement is, in this case, typically circa 4 dB in the relevant frequency band. This is further documented in Fig. 13 which illustrates the effect of the constraint horizon $N$ on performance, as measured by the filtered variance $S_f$ defined in Eq. (28). It shows that the performance of the Single-loop MSOC is monotonic and asymptotic in $N$ (The effect of $N$ on $S$ is similar.)

\(^7\)Following the results of [21], [67] (see also [31]), we included an irrational dc-component in order to avoid singular behaviour, which especially in the Single-loop $\Sigma\Delta$-converter case, would deteriorate performance significantly. For the same reason, unless otherwise stated, in all simulations, $f$ is chosen to be irrational.
Fig. 10. Effect of $f$, Single-loop $\Sigma\Delta$ conversion and Single-loop MSOC with $N = 2, 3$ ($A = 2$).

Fig. 11. Effect of $A$, Single-loop $\Sigma\Delta$ conversion and Single-loop MSOC with $N = 2, 3$ ($f = 7e^{-4} \approx 0.128$).

B. The MSOC

A more general case results when using the 3-bit alphabet $\mathbb{U} = \{-4, -3, \ldots, 3\}$ and the filters:

$$W(\rho) = \frac{\rho + 1.6}{\rho + 0.3}, \quad H(\rho) = 1. \quad (30)$$

This filter $W$ is low-pass, stable, and non-minimum phase. The solution to the associated Lyapunov Equation (25) of Theorem 2 is, in this case, $P \approx 1.8571$. In the sequel, we will denote it as the stabilizing $P$. 
1) DC inputs: We first choose $\bar{a}$ to be a dc input ($T_f = 5000$) and simulate three cases: the general $\Sigma\Delta$ converter of Fig. 1 with $G$ and $F$ chosen according to Eqs. (30) and (22), the MSOC with $N = 2$ and $P = 0$, and the MSOC with $N = 1$ and with the stabilizing $P$.

Fig. 14 illustrates the results. As can be seen, the effect of the final state weighting matrix $P$ is remarkable. Although the result of Theorem 2 is useful here, in principle, only for the integer valued dc levels, the MSOC with stabilizing $P$ far outperforms the other two converters. The
main reason for this, is that in case of the ΣΔ-converter and, to a lesser degree also for the MSOC with $P = 0$, the output stream converges for most input streams to periodic sequences comprising mainly the extremal values $-4$ and $3$. This phenomenon is avoided by choosing the stabilizing $P$, leading to significant performance enhancement.

It should be emphasized here that the MSOC with stabilizing $P$ used only a scalar quantizer, since a unitary constraint horizon $N = 1$ was chosen. Thus, the improvement in performance in comparison with ΣΔ-conversion is obtained with negligible increase in on-line complexity.

![Fig. 14. Effect of dc level on $S_f$.](image)

2) Periodic Inputs: Similar results can be observed for other input signals. In particular, Figs. 15 and 16 illustrate the performance achieved in the three cases analyzed above when sinusoidal signals, as in Eq. (29) are applied to the input. As can be seen, the stabilizing effect of $P$ yields performance improvement which is typically more than 20 dB.

Further insight can also be gained by inspecting the signal spectra. Fig. 17 is obtained by choosing a sinusoidal input as in Eq. (29) with $A = 2$ and $f = 1/7$. As can be seen, the ΣΔ converter gives rise to a signal with many (direct and shifted) harmonics (compare to the analytical results of [68]). In particular, it contains a period-two component, which dominates the input-frequency component by more than 20 dB.\(^8\) In comparison, the MSOC with unitary horizon

\(^8\)Note that these high frequency tones are specially dangerous, since they may couple easily into the baseband due to clock noise.
and $P$ set to the stabilizing value yields an output signal whose most significant contribution is at the input frequency.

3) Random Inputs: Finally we choose the signal $\tilde{a}$, to consist of 4000 observations of an independent random variable, uniformly distributed over the interval $[-4.5, 3.5]$. Fig. 18 shows the performance achieved with $\Sigma\Delta$-conversion and for the MSOC with stabilizing $P$ and different constraint horizons. Again the MSOC clearly outperforms the general $\Sigma\Delta$-Converter. As depicted in Fig. 19, the power spectrum density of the distortion caused by $\Sigma\Delta$-conversion has a large
component at the Nyquist frequency. This is due to an instability of the loop, which manifests itself as intermittent limit cycles (short term periodicities), which comprise only extremal values of $U$ and lead to performance degradation. The stabilizing choice for $P$ given by the Lyapunov Equation (25) suppresses this phenomenon.

![Fig. 17. Spectra of periodic input and converted signals.](image)

![Fig. 18. Performance of the stabilizing MSOC and $\Sigma\Delta$-conversion (random input).](image)
VIII. CONCLUSIONS

In this paper, we have presented a novel analog-to-digital conversion methodology, which is designed by minimizing a measure of the frequency-weighted distortion. Based upon moving horizon optimization ideas, the proposed Multi-Step-Optimal Converter (MSOC) can be implemented as a feedback loop, comprising two linear filters and a vector quantizer. The resultant architecture is amenable to on-line applications.

Furthermore, we have shown that $\Sigma\Delta$-conversion corresponds to a particular case of the new MSOC-scheme. In particular, various $\Sigma\Delta$-converters that have been proposed in the literature have been shown to correspond to the one-step-optimal form of the multi-step optimization problem examined here.

The MSOC has two prime advantages. Firstly, it allows for multi-step optimality. Secondly, final state weighting can be used to ensure stability-like properties of the MSOC and, thus, avoid tone-like behaviour without requiring the use of random dithering. As apparent from the simulation studies included, both aspects allow for enhanced performance.
APPENDIX I

PROOF OF THEOREM 1

By iterating Eq. (7), one can see that the cost function $V_N(\bar{u}(k))$ can be written in vector form as:

$$V_N(\bar{u}(k)) = (\Phi(\bar{a}(k) - \bar{u}(k)) + \Gamma x(k))^T(\Phi(\bar{a}(k) - \bar{u}(k)) + \Gamma x(k))$$

$$+ \|M(\bar{a}(k) - \bar{u}(k)) + A^N x(k)\|^2_2, \quad (31)$$

where the matrices $\Phi, \Gamma$ and $M$ are defined in Eqs. (18) and (15), while:

$$\bar{a}(k) \triangleq [a(k) \ a(k+1) \ldots \ a(k+N-1)]^T. \quad (32)$$

Direct manipulation of Expression (31) yields that:

$$V_N(\bar{u}(k)) = \|\bar{u}(k)\|^2_{\Psi_T \Psi} - 2(\bar{u}(k))^T((\Psi^T \Psi)\bar{a}(k) + (\Phi^T \Gamma + M^T P A^N)x(k))$$

$$+ \|x(k)\|^2_{\Gamma_T \Gamma + (A^N)^T P A^N} + 2(\bar{a}(k))^T(\Phi^T \Gamma + M^T P A^N)x(k) + \|\bar{a}(k)\|^2_{\Psi_T \Psi}, \quad (33)$$

where $M$ is defined in Eq. Note that only the first two terms on the right-hand-side of Eq. (33) depend upon the decision variable $\bar{u}(k)$.

We introduce the change of variables, $\bar{\mu}(k) = \Psi \bar{u}(k)$. This transforms $\bar{U}^N$ into $\bar{\bar{U}}^N$ defined in Eq. (19). Eq. (33) then allows us to rewrite the optimizer (10) as:

$$\bar{u}^*(k) = \Psi^{-1} \arg \min_{\bar{\mu}(k) \in \bar{\bar{U}}^N} J_N(\bar{\mu}(k)), \quad \text{where:}$$

$$J_N(\bar{\mu}(k)) \triangleq (\bar{\mu}(k))^T \bar{\mu}(k) - 2(\bar{\mu}(k))^T(\Psi \bar{a}(k) + \Psi^{-T}(\Phi^T \Gamma + M^T P A^N)x(k)).$$

The level sets of $J_N$ are spheres in $\mathbb{R}^N$, centred at $\Psi \bar{a}(k) + \Psi^{-T}(\Phi^T \Gamma + M^T P A^N)x(k)$. Hence, the constrained optimizer is given by the nearest neighbour:

$$\arg \min_{\bar{\mu}(k) \in \bar{\bar{U}}^N} J_N(\bar{\mu}(k)) = q_{\bar{\bar{U}}^N}(\Psi \bar{a}(k) + \Psi^{-T}(\Phi^T \Gamma + M^T P A^N)x(k)).$$

Eqs. (I) and (11) then yield:

$$u(k) = \left[1 \ 0 \ \ldots \ 0\right] \Psi^{-1} q_{\bar{\bar{U}}^N}(\Psi \bar{a}(k) + \Psi^{-T}(\Phi^T \Gamma + M^T P A^N)x(k)). \quad (34)$$

It follows directly from Eqs. (3), (13) and (32) that:

$$\bar{a}(k) = \left[1 \ \rho \ \ldots \ \rho^{N-1}\right]^T H(\rho)\bar{a}(k),$$

$$x(k) = (\rho I - A)^{-1} B(H(\rho)\bar{a}(k) - u(k)).$$

The result (16) follows by substituting these expressions into Eq. (34).
APPENDIX II

PROOF OF THEOREM 2

The proof is based upon Lyapunov stability [69] and utilizes the sequence of optimal costs defined as:

\[ V_N^*(\ell) \triangleq V_N(\bar{u}^*(\ell)), \quad \ell \in \mathbb{N}. \]

Suppose that an optimal sequence \( \bar{u}^*(k) \) has been found, where

\[ \bar{u}^*(k) = [u_k \quad u_{k+1} \ldots \quad u_{k+N-1}]^T. \]

Next, at sample \( \ell = k + 1 \), consider a related sequence

\[ u^* = [u_{k+1} \quad u_{k+2} \ldots \quad u_{k+N-1} \quad a(k+N)]^T \tag{35} \]

which belongs to \( \mathbb{U}^N \). Due to optimality, it follows that \( V_N^*(k + 1) \leq V_N(u^*) \). Direct calculation then yields:

\[
V_N(u^*) = V_N^*(k) + \|Ax(k+N)\|^2_P - \|x(k+N)\|^2_P + (e(k+N))^2 - (e(k))^2
\]
\[
= V_N^*(k) + \|Ax(k+N)\|^2_P - \|x(k+N)\|^2_P + (Cx(k+N))^2 - (e(k))^2
\]

since, with \( u^* \) given by (35), predictions of \( e \) used in \( V_N^*(k) \) and in \( V_N(u^*) \) coincide. Hence,

\[
V_N^*(k + 1) - V_N^*(k) \leq \|x(k+N)\|^2_{APAtP+PCTC} - (e(k))^2 = -(e(k))^2 \leq 0, \tag{36}
\]

after replacing (25) and the sequence \( V_N^*(k) \). As a consequence, \( \lim_{\ell \to \infty} V_N^*(\ell) \) exists and \( V_N^*(\ell + 1) - V_N^*(\ell) \to 0 \). Due to (36), also \( e(\ell) \to 0 \), which completes the proof.

REFERENCES


