INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING
Int. J. Adapt. Control Signal Process. 2005; **19**:125–152
Published online 12 November 2004 in Wiley InterScience (www.interscience.wiley.com) DOI:10.1002/acs.841

A fast-array Kalman filter solution to active noise control

Rufus Fraanje*,†, Ali H. Sayed, Michel Verhaegen and Niek J. Doelman

Delft University of Technology, Delft Center for Systems and Control, Mekelweg 2, 2628 CD, Delft, The Netherlands

SUMMARY

A Kalman filter solution to active control and its fast-array implementation are provided. The adaptive control problem is formulated as a state-estimation problem and no interchanging of the adaptive filter and the secondary-path is imposed. Moreover, no estimate of the disturbance signal is needed, and we exploit the structure in the state–space matrices to derive a fast-array implementation. A minimum variance estimate of the controller coefficients and the secondary path state is obtained. When there is no uncertainty in the secondary path, state equivalence with the modified filtered-RLS algorithm is proven. Using exponential forgetting, the analysis shows that in the generation of the filtered reference signal in the modified filtered-RLS, exponential forgetting should be incorporated too. Simulations show the superiority in convergence of the fast-array Kalman algorithm over the fast-array modified filtered-RLS algorithm.

KEY WORDS: adaptive filters; state–space; array algorithm; Kalman; RLS; tracking algorithm; structured state–space model

1. INTRODUCTION

Active noise and vibration control (ANVC) systems usually deal with a large amount of dominant, relatively weakly damped, resonance modes (in the order of 10–50) and need controllers with a large impulse response to obtain good disturbance rejection (in the order of 100–1000 taps). Furthermore, sampling rates are often in the order of 1–10 kHz to have sufficient control bandwidth. Besides, the controller should be able to adjust for variations in the system, like temperature variations. These constraints make ANVC a challenging control problem even in a time of fast increasing computer power.

Because of its computational efficiency and robustness properties, the filtered-X LMS (FxLMS) algorithm is very popular in ANVC systems (see, e.g. Reference [1]). However, in applications with broadband disturbances, and especially in the case of multiple channels, the convergence and tracking capacity of FxLMS is poor. This problem has encouraged many

*Correspondence to: Rufus Fraanje, Delft University of Technology, Delft Center for Systems and Control, Mekelweg 2, 2628 CD, Delft, The Netherlands.
†E-mail: r.fraanje@dcsc.tudelft.nl

Contract/grant sponsor: NSF; contract/grant numbers: CCR-0208573, ECS-0401188
Contract/grant sponsor: TNO and European Commission

researchers to develop alternatives to the FxLMS algorithm, see, e.g. References [2–8]. Also IIR adaptive LMS algorithms are proposed, which may allow a reduction of the number of filter coefficients. However, it is hard to analyse its convergence rate and the adaptive algorithm may very well convergence to a local minimum due to the non-linearity of the IIR filter structure.

Almost all algorithms proposed for ANVC are of the so-called filtered-X or filtered-reference type. They rely on the assumption that the adaptive filter and the controlled system, the so-called secondary path, may be interchanged. Neglecting transients from initial states, this is true for systems that are constant in time. However, since the adaptive filter varies in time this is not true anymore and algorithms based on this assumption may yield bad performance. This observation has motivated the introduction of the so-called *modified* filtered-X algorithms, in which the disturbance is estimated from the residual signal and an internal model of the secondary path, see References [2, 9]. It is interesting to note that in Reference [10] (especially Equation (13)) for detereministic disturbances, the modified filtered-X algorithms, though not named this way, were already derived from a self tuning regulator point of view. Although this is still an approximation in non-stationary applications where the optimal controller is varying in time, the modified filtered-X algorithms yield better convergence than the filtered-X algorithms, at the expense of increased computational load.

The problem of interchanging the adaptive filter and the secondary path was addressed in Reference [6] by reformulating the ANVC control problem as a state estimation problem. The state to be estimated contains the unknown filter coefficients and the unknown state of the secondary path system. However, this approach required an estimate of the disturbance signal.

In this paper, we reformulate the ANVC control problem also as a state-estimation problem, but without using an estimate of the disturbance. The state-estimation problem is solved by a Kalman filter, which has been chosen from an optimality point of view. Uncertainty in the secondary-path state, due to initial state uncertainty and/or noise, can be taken into account explicitly with improved convergence. We also show that in case there is no uncertainty in the secondary-path state, then the Kalman algorithm is equivalent to a modified filtered-RLS algorithm. The analysis of the equivalence of both algorithms also shows that the exponential forgetting needs to be applied to filtering the reference signal as well. Furthermore, a fast-array implementation of the Kalman filter algorithm is derived, which enables practical application of the algorithm. The derivation of this fast algorithm is based on the observation that although the underlying state–space model is *not* time-invariant, it is nevertheless a structured model in the sense defined in References [11, 12]. This article only addresses the single-channel case to focus on the concepts. However, the analysis and the proposed algorithms can be straightforwardly extended to the multiple-channel case, though might ask for carefull bookkeeping.

The paper is organized as follows. Section 2 formulates the estimation problem, provides necessary and sufficient conditions for a solution of this problem in terms of observability and persistency of excitation and presents the Kalman algorithm to solve this problem. Section 3 derives the new fast-array implementation of the Kalman algorithm. Section 4 compares the Kalman algorithm and a modified filtered-RLS algorithm and presents the conditions for equivalence of both algorithms. Section 5 illustrates the (fast-array) Kalman algorithm by simulation and Section 6 concludes the paper.

2. THE KALMAN FILTER SOLUTION

2.1. The state estimation problem

Consider the active feedforward control problem illustrated in Figure 1. The objective is to counteract the disturbance signal $d(k)$ by a secondary signal $y(k)$, which yields the residual signal $e(k)$. It is assumed, that the primary path can be decomposed into a series connection of an optimal feedforward controller $W^{o}(q^{-1})$ and a secondary path $S(q^{-1})$, which contains the dynamics between the actuators and the sensors. Usually this assumption is not satisfied, but the error due to imposing this assumption can often be neglected. Assuming that the noise signals $v_s(k)$ and $v_m(k)$ are uncorrelated with the disturbance reference signal $r(k)$, it can be verified easily that the residual signal is minimized if the feedforward controller $\hat{W}_k(q^{-1})$ equals $W^{o}(q^{-1})$. In this way, the feedforward control problem is reformulated in an estimation context [6].

We will assume that the unknown optimal controller is a FIR filter of length n_w

$$W^{o}(q^{-1}) = w_0 + w_1 q^{-1} + \cdots w_{n_w-1} q^{-n_w+1} \tag{1}$$

and that $\hat{W}_k(q^{-1})$ has the same structure

$$\hat{W}_k(q^{-1}) = \hat{w}_0(k) + \hat{w}_1(k) q^{-1} + \cdots \hat{w}_{n_w-1}(k) q^{-n_w+1} \tag{2}$$

Let

$$w^{o} = [w_0 \;\; w_1 \;\; \cdots \;\; w_{n_w-1}]^{T} \tag{3}$$

$$\hat{w}(k) = [\hat{w}_0(k) \;\; \hat{w}_1(k) \;\; \cdots \;\; \hat{w}_{n_w-1}(k)]^{T} \tag{4}$$

$$r_{n_w} = [r(k) \;\; r(k-1) \;\; \cdots \;\; r(k-n_w+1)]^{T} \tag{5}$$

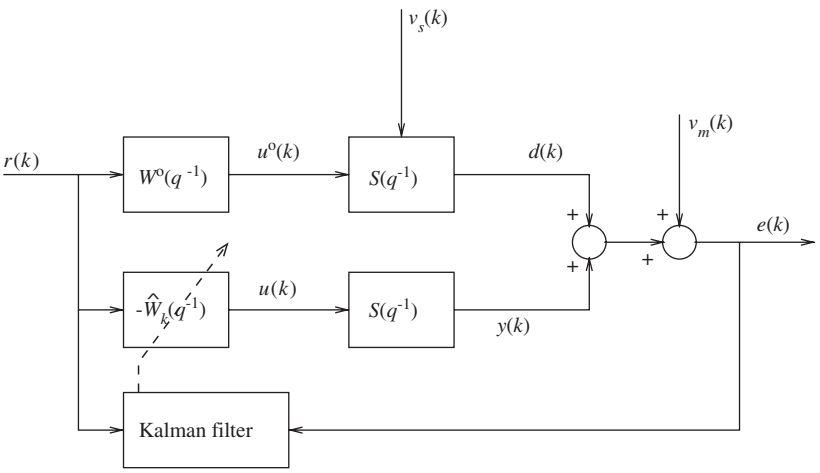

Figure 1. A block diagram representation of a feedforward active control configuration with a feedforward controller $\hat{W}_k(q^{-1})$ estimated by a Kalman filter to minimize the residual disturbance $e(k)$. The primary path is assumed to consist of a series connection of the (unknown) optimal feedforward controller $W^{o}(q^{-1})$ and the secondary path $S(q^{-1})$.

then the optimal control signal is given by

$$u^{\mathrm{o}}(k) = r_{n_w}^{\mathrm{T}}(k) w^{\mathrm{o}} \tag{6}$$

and the actual control signal by

$$u(k) = -r_{n_w}^{\mathrm{T}}(k) \hat{w}(k) \tag{7}$$

We will also allow for small time variations in the optimal filter coefficients, which are taken into account by an exponential forgetting factor λ:

$$0 \ll \lambda \leqslant 1$$

Then (6) is replaced by

$$u^{\mathrm{o}}(k) = r_{n_w}^{\mathrm{T}}(k) w(k) \tag{8}$$

where $w(k)$ is iteratively defined by

$$w(k+1) = \lambda^{-1/2} w(k), \quad w(0) = w^{\mathrm{o}} \tag{9}$$

This model of the dynamics in the optimal filter coefficients may allow better tracking capacity, see, e.g. Reference [11]. In Equation (9) also an innovation term can be included to model a larger set of variations in the optimal filter coefficients, which may further improve tracking behavior. However, in the derivation of the fast-array algorithm in Section 3 an innovation term in (9) cannot be taken into account.

The secondary path $S(q^{-1})$ will be described in state–space form. Usually, the state dimension n_s is high (in the range 20–100) for acoustical or vibrational systems. For this reason, using a FIR model of sufficient length to model the dynamics of $S(q^{-1})$ can help lower the computational complexity, especially for well damped systems. Because the FIR model structure is contained in the state–space model structure, i.e. a FIR model is a state–space model with particular structure, it is just a matter of straightforward computation to derive the expressions for secondary path models with FIR structure. Other (canonical) parameterizations contained in the state–space structure can be used as well.

The noise signal $v_s(k) \in \mathbb{R}^{n_v}$ distorts the secondary-path state and $v_m(k)$ distorts the measured output $e(k)$. We assume $v_s(k)$ and $v_m(k)$ are both stationary zero-mean white-noise signals that are independent of $r(k)$ and satisfy

$$E\left(\begin{bmatrix} v_s(k) \\ v_m(k) \end{bmatrix} \begin{bmatrix} v_s(l) \\ v_m(l) \end{bmatrix}^{\mathrm{T}} \right) = \begin{bmatrix} Q & 0_{n_v \times 1} \\ 0_{1 \times n_v} & R \end{bmatrix} \delta_{kl}, \quad Q \geqslant 0, \quad R > 0 \tag{10}$$

with δ_{kl} the Kronecker delta function (if $k = l$ then $\delta_{kl} = 1$, otherwise $\delta_{kl} = 0$). Note that the assumption that $v_s(k)$ is white is not restrictive, since this can always be assured by incorporating the noise-shaping filter into the secondary path system. The disturbance reference signal $r(k)$ may be white or colored noise, a sinusoid, stationary or non-stationary. We will only impose a persistency of excitation condition on $r(k)$, which will be stated in Theorem 1 in Section 2.1.

Let (A_s, B_s, C_s, D_s) be the state–space matrices that model the secondary path $S(q^{-1})$, then the disturbance $d(k)$ is written as

$$\theta^1(k+1) = A_s \theta^1(k) + B_s u^{\mathrm{o}}(k) + G_s v_s(k), \qquad \theta^1(0) = \theta_0^1 \tag{11}$$

$$d(k) = C_s \theta^1(k) + D_s u^{\mathrm{o}}(k) \tag{12}$$

and the secondary signal $y(k)$ as

$$\theta^2(k+1) = A_s\theta^2(k) + B_s u(k), \qquad \theta^2(0) = \theta_0^2 \tag{13}$$

$$y(k) = C_s\theta^2(k) + D_s u(k) \tag{14}$$

with $\theta^1(k), \theta^2(k) \in \mathbb{R}^{n_s}$. Since

$$e(k) = d(k) + y(k) + v_m(k) \tag{15}$$

and the state–space matrices in (11)–(12) and(13)–(14) are the same, we can write

$$\theta(k+1) = A_s\theta(k) + B_s(u^{\mathrm{o}}(k) + u(k)) + G_s v_s(k), \qquad \theta(0) = \theta_0 \tag{16}$$

$$e(k) = C_s\theta(k) + D_s(u^{\mathrm{o}}(k) + u(k)) + v_m(k) \tag{17}$$

with $\theta_0 = \theta_0^1 + \theta_0^2$ and $\theta(k) = \theta^1(k) + \theta^2(k)$.

Substituting (7)–(9) into the state–space equations (16)–(17) gives the final state–space description of the active control system considered in this paper

$$\begin{bmatrix} w(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} \lambda^{-1/2} I_{n_w} & 0_{n_w \times n_s} \\ B_s r_{n_w}^{\mathrm{T}}(k) & A_s \end{bmatrix} \begin{bmatrix} w(k) \\ \theta(k) \end{bmatrix} - \begin{bmatrix} 0_{n_w \times n_w} \\ B_s r_{n_w}^{\mathrm{T}} \end{bmatrix} \hat{w}(k) + \begin{bmatrix} 0_{n_w \times n_v} \\ G_s \end{bmatrix} v_s(k), \qquad \begin{bmatrix} w(0) \\ \theta(0) \end{bmatrix} = \begin{bmatrix} w^{\mathrm{o}} \\ \theta_0 \end{bmatrix} \tag{18}$$

$$e(k) = [D_s r_{n_w}^{\mathrm{T}}(k) \;\; C_s] \begin{bmatrix} w(k) \\ \theta(k) \end{bmatrix} - D_s r_{n_w}^{\mathrm{T}}(k)\hat{w}(k) + v_m(k) \tag{19}$$

For ease of notation, we define

$$A_k = \begin{bmatrix} \lambda^{-1/2} I_{n_w} & 0_{n_w \times n_s} \\ B_s r_{n_w}^{\mathrm{T}}(k) & A_s \end{bmatrix} \tag{20}$$

$$B_k = \begin{bmatrix} 0_{n_w \times n_w} \\ -B_s r_{n_w}^{\mathrm{T}}(k) \end{bmatrix} \tag{21}$$

$$G = \begin{bmatrix} 0_{n_w \times n_w} \\ G_s \end{bmatrix} \tag{22}$$

$$C_k = [D_s r_{n_w}^{\mathrm{T}}(k) \quad C_s] \tag{23}$$

$$D_k = -D_s r_{n_w}^{\mathrm{T}}(k) \tag{24}$$

and the augmented state

$$x(k) = \begin{bmatrix} w(k) \\ \theta(k) \end{bmatrix} \tag{25}$$

With these definitions, (18) and (19) are rewritten more compactly as

$$x(k+1) = A_k x(k) + B_k \hat{w}(k) + G v_s(k) \tag{26}$$

$$e(k) = C_k x(k) + D_k \hat{w}(k) + v_m(k) \tag{27}$$

2.2. The relation between observability and persistency of excitation

Before we derive algorithms to estimate the augmented state $x(k)$, we investigate the observability of system (26)–(27).

Definition 1 (Rugh [13])
The state–space system (26)–(27) is said to be *observable over the interval* $[k_0, k_0+N]$ if any initial state $x(k_0)$ is uniquely determined by the corresponding zero-input response $e(k)$ for $k = k_0, \ldots, k_0+N-1$ (i.e. the response corresponding to $\hat{w}(k) = 0$, $v_s(k) = 0$ and $v_m(k) = 0$).

Since the zero-input response for $k = k_0, \ldots, k_0+N-1$ of (26)–(27) can be written as

$$\begin{bmatrix} e(k_0) \\ e(k_0+1) \\ e(k_0+2) \\ \vdots \\ e(k_0+N-1) \end{bmatrix} = \underbrace{\begin{bmatrix} C_{k_0} \\ C_{k_0+1}A_{k_0} \\ C_{k_0+2}A_{k_0+1}A_{k_0} \\ \vdots \\ C_{k_0+N-1}A_{k_0+N-2}\cdots A_{k_0} \end{bmatrix}}_{\coloneqq \Gamma(k_0, k_0+N-1)} x(k_0) \tag{28}$$

a necessary and sufficient condition for observability on $[k_0, k_0+N]$ is that $\Gamma(k_0, k_0+N-1)$ should have full column rank. Hence, C_k and A_k, $k = k_0, \ldots, k_0+N-1$ should be such that there exists an N such that $\Gamma(k_0, k_0+N-1)$ has full column rank. Before stating the theorem which provides necessary and sufficient conditions for observability in terms of conditions on (A_s, C_s) and the reference signal $r(k)$, let us have a closer look at $\Gamma(k_0, k_0+N-1)$.

Using definitions (20) and (23) it can be verified (e.g. by induction) that

$$\begin{aligned}
C_{k_0} &= [D_s r_{n_w}^{\mathrm{T}}(k_0) \quad C_s] \\
C_{k_0+1}A_{k_0} &= [C_s B_s r_{n_w}^{\mathrm{T}}(k_0) + \lambda^{-1/2} D_s r_{n_w}^{\mathrm{T}}(k_0+1) \quad C_s A_s] \\
C_{k_0+2}A_{k_0+1}A_{k_0} &= [C_s A_s B_s r_{n_w}^{\mathrm{T}}(k_0) + \lambda^{-1/2} C_s B_s r_{n_w+1}^{\mathrm{T}}(k_0+1) + \lambda^{-1} D_s r_{n_w+2}^{\mathrm{T}}(k_0+2) \;\; C_s A_s^2] \\
&\vdots \qquad\qquad\qquad \vdots \\
C_{k_0+N-1}A_{k_0+N-2}\cdots A_{k_0} &= [C_s A_s^{N-2} B_s r_{n_w}^{\mathrm{T}}(k_0) + \lambda^{-1/2} C_s A_s^{N-3} B_s r_{n_w+1}^{\mathrm{T}}(k_0+1) \\
&\quad + \cdots + \lambda^{-(N-2)/2} C_s B_s r_{n_w}^{\mathrm{T}}(k_0+N-2) \\
&\quad + \lambda^{-(N-1)/2} D_s r_{n_w}^{\mathrm{T}}(k_0+N-1) \;\; C_s A_s^{N-1}]
\end{aligned}$$

Hence, $\Gamma(k_0, k_0+N-1)$ can be written as

$$\Gamma(k_0, k_0+N-1) = [H_N \mathscr{R}_N(k_0) \quad \Gamma_N] \tag{29}$$

where $\mathscr{R}_N(k_0)$ is the Toeplitz matrix defined by

$$\mathscr{R}_N(k_0) \coloneqq \begin{bmatrix} r_{n_w}^{\mathrm{T}}(k_0) \\ r_{n_w}^{\mathrm{T}}(k_0+1) \\ \vdots \\ r_{n_w}^{\mathrm{T}}(k_0+N-1) \end{bmatrix} = \begin{bmatrix} r(k_0) & r(k_0-1) & \cdots & r(k_0-n_w+1) \\ r(k_0+1) & r(k_0) & \cdots & r(k_0-n_w+2) \\ \vdots & \vdots & & \vdots \\ r(k_0+N-1) & r(k_0+N-2) & \cdots & r(k_0+N-n_w) \end{bmatrix} \tag{30}$$

H_N is the lower triangular matrix with impulse response coefficients weighted by powers of the inverse square-root of the forgetting factor λ

$$H_N \coloneqq \begin{bmatrix} D_s & & & & \\ C_sB_s & \lambda^{-1/2}D_s & & & \\ C_sA_sB_s & \lambda^{-1/2}C_sB_s & \lambda^{-1}D_s & & \\ \vdots & \vdots & \ddots & \ddots & \\ (C_sA_s^{N-2}B_s) & (\lambda^{-1/2}C_sA_s^{N-3}B_s) & \cdots & (\lambda^{-(N-2)/2}C_sB_s) & (\lambda^{-(N-1)/2}D_s) \end{bmatrix} \tag{31}$$

and Γ_N is the extended observability matrix

$$\Gamma_N \coloneqq \begin{bmatrix} C_s \\ C_sA_s \\ \vdots \\ C_sA_s^{N-1} \end{bmatrix} \tag{32}$$

Suppose that the secondary path has t samples delay. Then, the first t terms of the impulse response

$$S(q^{-1}) = D_s + C_sB_sq^{-1} + C_sA_sB_sq^{-2} + C_sA_s^2B_sq^{-3} + \cdots$$

are zero and thus the first t rows and the last t columns of H_N are filled by zeros. In this case, the product $H_N\mathscr{R}(k_0)$ can be simplified to

$$\bar{H}_N\bar{\mathscr{R}}_N(k_0) = H_N\mathscr{R}(k_0) \tag{33}$$

where $\bar{\mathscr{R}}_N(k_0)$ ($\bar{H}_N$) is defined as the matrix which consists of the first $N-t$ rows (columns) of $\mathscr{R}_N(k_0)$ (H_N). Since we do not consider the case $S(q^{-1})=0$ and since (A_s, C_s) will be assumed to be observable we have that $t \leqslant n_s$, i.e. the number of pure delays is smaller than or equal to the order of $S(q^{-1})$. Now we establish the following result.

Theorem 1

Let $k \geqslant 0$ and $N \geqslant n_w + n_s$. Then, the state–space system (26)–(27) is observable on $[k, k+N]$ if and only if the following conditions are satisfied:

- the pair (A_s, C_s) is observable;
- the Toeplitz matrix $\bar{\mathscr{R}}_N(k)$ has full rank n_w and is such that the columns of $\bar{H}_N\bar{\mathscr{R}}_N(k)$ are not in the range space of Γ_N.

Proof
if: First, note that $\bar{H}_N \in \mathbb{R}^{N\times N-t}$ is guaranteed to have full rank $N-t$, which can be directly verified from its structure. Then, using Sylvester's rank condition we have

$$\begin{aligned} \operatorname{rank}(\bar{H}_N) + \operatorname{rank}(\bar{\mathscr{R}}_N(k)) - (N-t) \quad &\leqslant \quad \operatorname{rank}(\bar{H}_N\bar{\mathscr{R}}_N(k)) \quad \leqslant \quad \min(\operatorname{rank}(\bar{H}_N), \operatorname{rank}(\bar{\mathscr{R}}_N(k))) \\ &\Leftrightarrow \\ n_w \quad &\leqslant \quad \operatorname{rank}(\bar{H}_N\bar{\mathscr{R}}_N(k)) \quad \leqslant \quad n_w \end{aligned}$$

and thus $\bar{H}_N\bar{\mathscr{R}}_N(k)$ has full (column) rank n_w. Since, the pair (A_s, C_s) is observable and $N = n_w + n_s \geqslant n_s$ the extended observability matrix Γ_N has full (column) rank n_s. Finally, using the assumption that $\bar{\mathscr{R}}_N(k)$ is such that the columns of $\bar{H}_N\bar{\mathscr{R}}_N(k)$ are not in the range of Γ_N, we can conclude that

$$\Gamma(k, k+N-1) = [\bar{H}_N\bar{\mathscr{R}}_N(k)\ \Gamma_N] \tag{34}$$

has full column rank $n_w + n_s$. Hence, the state–space system (26)–(27) is observable on $[k, k+N]$.

only if: This can be proven by contradiction. First, assume that (A_s, C_s) is not observable. Then, Γ_N does not have full rank and thus $\Gamma(k, k+N-1)$ does not have full rank. Second, assume that $\bar{\mathscr{R}}_N(k)$ does not have full rank. Then by using Sylvester's rank condition we have $\operatorname{rank}(\bar{H}_N\bar{\mathscr{R}}_N(k)) < n_w$ and thus $\Gamma(k, k+N-1)$ does not have full rank. Third, assume that $\bar{\mathscr{R}}_N(k)$ has full rank and (A_s, C_s) is observable, but $\bar{\mathscr{R}}_N(k)$ is such that at least one column of $\bar{H}_N\bar{\mathscr{R}}_N(k)$ is in the range of Γ_N. Then, $\Gamma(k, k+N-1)$ has columns which are linearly dependent and thus does not have full rank. Concluding, all conditions stated in the theorem are necessary to guarantee that $\Gamma(k, k+N-1)$ has full column rank, and thus necessary to guarantee observability of the state–space model (26)–(27) on $[k, k+N]$. □

The condition that $\bar{R}_N(k)$ has full rank n_w can be seen as a *persistency of excitation* condition on the reference signal $r(k)$, which is natural in system identification and adaptive filtering algorithms, see, e.g. [14]. For example, when $r(k)$ is a single sinusoid n_w should be $n_w \leqslant 2$, which is in agreement with the fact that a single sinusoid can be perfectly cancelled using an FIR filter with two taps. Furthermore, it may be possible that, even in case n_w is chosen properly, $\bar{R}_N(k) \in \mathbb{R}^{N-t\times n_w}$ will not be full rank for $N = n_w + n_s$, e.g. due to the $t \leqslant n_s$ samples pure delay. But in general for each signal $r(k)$ there exists $N > n_w + n_s$ such that $\bar{R}_N(k)$ is guaranteed to be full rank (for all k).

Note that $\bar{H}_N$ is a convolution matrix. Hence, each column of $\bar{H}_N\bar{\mathscr{R}}_N(k)$ can be interpreted as a filtering operation of the corresponding column of $\bar{\mathscr{R}}_N(k)$ by the secondary path system weighted by exponential forgetting (determined by λ). In addition, the column space of Γ_N is equivalent to the space spanned by the initial state responses up to N. From these facts, it follows that it is quite accidental that there should exists a column of $\bar{H}_N\bar{\mathscr{R}}_N(k)$ which is contained in the range of Γ_N. Or, stated otherwise, that there exists a reference signal $r(k)$ such that filtering it by the secondary path and weighted by the exponential forgetting factor is the same as an initial state response. In the following we assume that the conditions stated in Theorem 1 are satisfied.

Note, that from Theorem 1 it is inferred that at least $n_w + n_s$ samples are necessary to estimate the state. In practice more samples are necessary due to the presence of state and measurement noise.

2.3. The Kalman filter

Since the Kalman filter provides a minimum variance estimate of the state at every sampling instant given the model of the system and the covariances of the white-noise signals $v_s(k)$ and $v_m(k)$, we will use this filter to estimate $w(k)$ as well as $\theta(k)$.

To apply the Kalman filter, we assume that the initial state is uncorrelated with $v_s(k)$ and $v_m(k)$, i.e.

$$\mathrm{E}\left(\begin{bmatrix} x(0) \\ v_s(k) \\ v_m(k) \end{bmatrix}\begin{bmatrix} x(0) \\ v_s(l) \\ v_m(l) \end{bmatrix}^{\mathrm{T}}\right) = \begin{bmatrix} \Pi_0 & 0_{n_w+n_s\times n_v} & 0_{n_w+n_s\times 1} \\ 0_{n_v\times n_w+n_s} & Q\delta_{kl} & 0_{n_v\times 1} \\ 0_{1\times n_w+n_s} & 0_{1\times n_v} & R\delta_{kl} \end{bmatrix}$$

where

$$\Pi_0 = \begin{bmatrix} \Pi_0^{ww} & \Pi_0^{w\theta} \\ \Pi_0^{\theta w} & \Pi_0^{\theta\theta} \end{bmatrix} > 0$$

$$\mathrm{E}(w(0)w^{\mathrm{T}}(0)) = \Pi_0^{ww}, \quad \mathrm{E}(w(0)\theta^{\mathrm{T}}(0)) = \Pi_0^{w\theta} = \Pi_0^{\theta w^{\mathrm{T}}}, \quad \mathrm{E}(\theta(0)\theta^{\mathrm{T}}(0)) = \Pi_0^{\theta\theta}$$

The Kalman filter can be described in at least two forms: the time/measurement update form and the prediction form [15]. The output of the time/measurement update form is an estimate of $x(k)$ given the measurements $\{e(0), e(1), \ldots, e(k)\}$, denoted as $\hat{x}(k|k)$, together with its error covariance matrix $P_{k|k}$. The output of the prediction form is an estimate of $x(k+1)$ given the same measurements $\{e(0), e(1), \ldots, e(k)\}$, denoted as $\hat{x}(k+1|k)$ or just $\hat{x}(k+1)$, together with its error covariance matrix $P_{k+1|k}$ or just P_{k+1}. Because, we need an estimate of $w(k+1)$ to calculate the control signal at the (next) iteration $k+1$, we will use the prediction form in the sequel.

The Kalman filter in prediction form is given by the following equations for $k \geqslant 0$:

$$\hat{x}(0) = 0_{n_w+n_s\times 1} \tag{35}$$

$$P_0 = \Pi_0 \tag{36}$$

$$\varepsilon(k) = e(k) - C_k\hat{x}(k) - D_k\hat{w}(k) \tag{37}$$

$$R_{e,k} = R + C_kP_kC_k^{\mathrm{T}} \tag{38}$$

$$K_k = A_kP_kCk^{\mathrm{T}} \tag{39}$$

$$\hat{x}(k+1) = A_k\hat{x}(k) + B_k\hat{w}(k) + K_kR_{e,k}^{-1}\varepsilon(k) \tag{40}$$

$$P_{k+1} = A_kP_kA_k^{\mathrm{T}} - K_kR_{e,k}^{-1}K_k^{\mathrm{T}} + GQG^{\mathrm{T}} \tag{41}$$

For further reference, we partition P_k similarly to Π_0 as

$$P_k = \begin{bmatrix} P_w^{ww} & P_k^{w\theta} \\ P_k^{\theta w} & P_k^{\theta\theta} \end{bmatrix} \tag{42}$$

Note that using definitions (20)–(25) and partitioning

$$K_k = \begin{bmatrix} K_k^w \\ K_k^\theta \end{bmatrix}, \quad K_k^w \in \mathbb{R}^{n_w \times 1}, \quad K_k^\theta \in \mathbb{R}^{n_s \times 1} \tag{43}$$

then expression (37) for innovation $\varepsilon(k)$, and expression (40) for the state-estimate update equation $\hat{x}(k+1)$ can be simplified to

$$\varepsilon(k) = e(k) - C_s \hat{\theta}(k) \tag{44}$$

$$\begin{bmatrix} \hat{w}(k+1) \\ \hat{\theta}(k+1) \end{bmatrix} = \begin{bmatrix} \lambda^{-1/2} \hat{w}(k) \\ A_s \hat{\theta}(k) \end{bmatrix} + \begin{bmatrix} K_k^w \\ K_k^\theta \end{bmatrix} R_{e,k}^{-1} \varepsilon(k) \tag{45}$$

The resulting Kalman filter algorithm, to solve the active control problem, is listed in the first column of Table I.

Table I. Kalman algorithm in covariance and fast-array forms.

Kalman covariance form	Fast-array form
Assumptions:	
$0 \ll \lambda \leqslant 1$	idem
$\begin{cases} R > 0 \\ Q \geqslant 0 \end{cases}$	$\begin{cases} \text{idem} \\ Q > 0 \text{ such that the pair } (A_s, G_s Q^{1/2}) \\ \text{is unit-circle controllable;} \end{cases}$
	the pair (A_s, C_s) is observable
$\Pi_0 = \begin{bmatrix} \Pi_0^{ww} & \Pi_0^{w\theta} \\ \Pi_0^{\theta w} & \Pi_0^{\theta\theta} \end{bmatrix} > 0$	$\begin{cases} \Pi_{-1} = \begin{bmatrix} \Pi_{-1}^{ww} & 0_{n_w \times n_s} \\ 0_{n_s \times n_w} & \Pi_{-1}^{\theta\theta} \end{bmatrix} > 0 \\ \text{with} \\ \Pi_{-1}^{ww} = \sqrt{\delta} \cdot \text{diag}\{\lambda, \lambda^2, \ldots, \lambda^{n_w}\}, \quad \delta > 0 \\ \text{and } \Pi_{-1}^{\theta\theta} > 0 \text{ satisfies the DARE} \\ \Pi_{-1}^{\theta\theta} = A_s \Pi_{-1}^{\theta\theta} A_s^{\mathrm{T}} + G_s Q G_s^{\mathrm{T}} + \\ \quad -A_s \Pi_1^{\theta\theta} C_s^{\mathrm{T}} (R + C_s \Pi_{-1}^{\theta\theta} C_s^{\mathrm{T}})^{-1} C_s \Pi_{-1}^{\theta\theta} A_s^{\mathrm{T}} \end{cases}$
	$r(k) = 0 \quad$ for $-n_w - 1 \leqslant k \leqslant -1$ (i.e. prewindowed data)
Initialization:	
$\begin{cases} \hat{w}(0) = 0_{n_w \times 1} \\ \hat{\theta}(0) = 0_{n_s \times 1} \end{cases}$	idem
$r_{n_w}(-1) = [r(-1) \;\; r(-2) \;\; \cdots \;\; r(-n_w)]^{\mathrm{T}}$	$r_{n_w+1}(-1) = 0_{n_w+1 \times 1}$

Table I. *Continued.*

Kalman covariance form	Fast-array form
$P_0 = \Pi_0$	$\begin{cases} \tilde{L}_{-1} = \sqrt{\delta}\begin{bmatrix} 1 & 0 \\ 0_{n_w-1\times 1} & 0_{n_w-1\times 1} \\ 0 & \lambda^{n_w/2} \\ 0_{n_2\times 1} & 0_{n_s\times 1} \end{bmatrix} \\ R_{e,-1}^{1/2} = (R + C_s\Pi_{-1}^{\theta\theta}C_s^{\mathrm{T}})^{1/2} \\ \bar{K}_{-1}^{w} = 0_{n_w\times 1} \\ \bar{K}_{-1}^{\theta} = A_s\Pi_{-1}^{\theta\theta}C_s^{\mathrm{T}}R_{e,-1}^{-1/2} \end{cases}$
Iterate for $k \geqslant 0$:	
$r_{n_w}(k) = [r(k)\ \ r_{n_w-1}^{\mathrm{T}}(k-1)]^{\mathrm{T}}$	$r_{n_w+1}(k) = [r(k)\ \ r_{n_w}^{\mathrm{T}}(k-1)]^{\mathrm{T}}$
$\begin{cases} A_k = \begin{bmatrix} \lambda^{-1/2}I_{n_w} & 0_{n_w\times n_s} \\ B_s r_{n_w}^{\mathrm{T}}(k) & A_s \end{bmatrix} \\ C_k = [D_s r_{n_w}^{\mathrm{T}}(k)\ \ C_s] \end{cases}$	$\begin{cases} \tilde{A}_k = \begin{bmatrix} \lambda^{-1/2}I_{n_w+1} & 0_{n_w+1\times n_s} \\ B_s r_{n_w+1}^{\mathrm{T}}(k) & A_s \end{bmatrix} \\ \tilde{C}_k = [D_s r_{n_w+1}^{\mathrm{T}}(k)\ \ C_s] \end{cases}$
$\varepsilon(k) = e(k) - C_s\hat{\theta}(k)$	idem
$\begin{cases} \begin{bmatrix} K_k^w \\ K_k^\theta \end{bmatrix} = K_k = A_k P_k C_k^{\mathrm{T}} \\ R_{e,k} = R + C_k P_k C_k^{\mathrm{T}} \\ P_{k+1} = A_k P_k A_k^{\mathrm{T}} - K_k R_{e,k}^{-1} K_k^{\mathrm{T}} + GQG^{\mathrm{T}} \end{cases}$	$\begin{cases} \text{Perform } J\text{-unitary rotation to make the } 1-2 \text{ block in the post-array equal to zero, } J = (I_2 \oplus -1), \Theta_{k-1}J\Theta_{k-1}^{\mathrm{T}} = J \\ \begin{bmatrix} R_{e,k-1}^{1/2} & \tilde{C}_k\tilde{L}_{k-1} \\ \begin{bmatrix} 0 \\ \bar{K}_{k-1}^w \\ \bar{K}_{k-1}^\theta \end{bmatrix} & \tilde{A}_k\tilde{L}_{k-1} \end{bmatrix}\Theta_{k-1} = \begin{bmatrix} R_{e,k}^{1/2} & 0_{1\times 2} \\ \begin{bmatrix} \bar{K}_k^w \\ 0 \\ \bar{K}_k^\theta \end{bmatrix} & \tilde{L}_k \end{bmatrix} \end{cases}$
$\begin{bmatrix} \hat{w}(k+1) \\ \hat{\theta}(k+1) \end{bmatrix} = \begin{bmatrix} \lambda^{-1/2}\hat{w}(k) \\ A_s\hat{\theta}(k) \end{bmatrix} + \begin{bmatrix} K_k^w \\ K_k^\theta \end{bmatrix} R_{e,k}^{-1}\varepsilon(k)$	$\begin{bmatrix} \hat{w}(k+1) \\ \hat{\theta}(k+1) \end{bmatrix} = \begin{bmatrix} \lambda^{-1/2}\hat{w}(k) \\ A_s\hat{\theta}(k) \end{bmatrix} + \begin{bmatrix} \bar{K}_k^w \\ \bar{K}_k^\theta \end{bmatrix} R_{e,k}^{-1/2}\varepsilon(k)$

Now, the implementation of the Kalman filter using expressions (38), (39) and (41) is computationally complex for most practical applications. For instance, recursion (41) has at least $O((n_w + n_s)^2)$ complexity, assuming that A_s is just a shift matrix, which is the case when an FIR model is used for the secondary path (if A_s has no structure at all, the complexity will be at least $O(n_s^3 + (n_w + n_s)^2)$).

In Section 4 we will show that by setting

$$\Pi_0^{w\theta} = \Pi_0^{\theta w^{\mathrm{T}}} = 0_{n_w\times n_s}, \quad \Pi_0^{\theta\theta} = 0_{n_s\times n_s}, \quad \text{and} \quad Q = 0_{n_v\times n_v}$$

which assumes perfect knowledge of the secondary path initial state and $v_s(k) = 0$ for all $k \geqslant 0$, the Kalman algorithm simplifies to a modified filtered-RLS algorithm. However, this assumption is rather strong, and may degrade the performance of the algorithm severely in case it is not satisfied, as will be illustrated by simulation in Section 5.

For now we proceed to derive a fast-array implementation of the Kalman filter algorithm by exploiting structure in the state–space matrices, thus reducing the computational complexity down to $O(n_w + n_s)$ per iteration.

3. THE FAST-ARRAY KALMAN FILTER

3.1. The structure in the state–space model

We will base our derivation on Reference [12], where a fast implementation of the Kalman filter for certain *time-varying* systems with structure was derived. Consider the definition of $r_{n_w}(k)$ in (5). It is clear that $r_{n_w}(k)$ and $r_{n_w}(k+1)$ have $n_w - 1$ entries in common but at shifted positions of each other. Let us define the shift-matrix $Z_{n_w} \in \mathbb{R}^{n_w \times n_w}$ as the matrix with ones on its first subdiagonal and zeros elsewhere. Then, we can write

$$r_{n_w}^{\mathrm{T}}(k) = r_{n_w}^{\mathrm{T}}(k+1)Z_{n_w} + [0_{1\times n_w-1} \quad r(k-n_w+1)]$$

Using this result, we are able to relate A_k and A_{k+1} to each other as well as C_k and C_{k+1}. In our case, G does not depend on k, but should satisfy a particular condition given below. Though the state–space matrices B_k and D_k are also related to B_{k+1} and D_{k+1} we do not need this relation in the derivation of the fast-array Kalman filter, since they determine the deterministic part of the state update which does not influence the Kalman filter expressions (38)–(41). Note, that we already exploited the structure in B_k and D_k in Equations (44) and (45).

To relate A_k and A_{k+1}, and C_k and C_{k+1}, we will first introduce the augmented state–space system, which is equivalent to (26)–(27):

$$\tilde{x}(k+1) = \tilde{A}_k\tilde{x}(k) + \tilde{B}_k\hat{w}(k) + \tilde{G}v_s(k) \tag{46}$$

$$e(k) = \tilde{C}_k\tilde{x}(k) + D_k\hat{w}(k) + v_m(k) \tag{47}$$

where

$$\tilde{A}_k = \begin{bmatrix} \lambda^{-1/2}I_{n_w+1} & 0_{n_w+1\times n_s} \\ B_s r_{n_w+1}^{\mathrm{T}}(k) & A_s \end{bmatrix} \tag{48}$$

$$\tilde{B}_k = \begin{bmatrix} 0_{n_w+1\times n_w} \\ -B_s r_{n_w}^{\mathrm{T}}(k) \end{bmatrix} \tag{49}$$

$$\tilde{G} = \begin{bmatrix} 0_{n_w+1\times n_v} \\ G_s \end{bmatrix} \tag{50}$$

$$\tilde{C}_k = [D_s r_{n_w+1}^{\mathrm{T}}(k) \quad C_s] \tag{51}$$

and the augmented initial state is given by

$$\tilde{x}(0) = \begin{bmatrix} w^{\mathrm{o}} \\ 0 \\ \theta(0) \end{bmatrix} \in \mathbb{R}^{n_w+1+n_s} \tag{52}$$

Because the (n_w+1)th entry of $\tilde{x}(k)$ is uncontrollable from the deterministic input $\hat{w}(k)$ as well as from the stochastic input $v_s(k)$, it will keep its initial zero value. It can be verified easily using

(46) that $\tilde{x}(k)$ has a similar form, namely

$$\tilde{x}(k) = \begin{bmatrix} w(k) \\ 0 \\ \theta(k) \end{bmatrix}, \quad \forall k \geqslant 0$$

and thus the output given by (47) is exactly the same as the output given by (27). Because there is no uncertainty in the $(n_w + 1)$th entry of $\tilde{x}(0)$ we define

$$\mathrm{E}(\tilde{x}(0)\tilde{x}^{\mathrm{T}}(0)) = \tilde{\Pi}_0 = \begin{bmatrix} \Pi_0^{ww} & 0_{n_w \times 1} & \Pi_0^{w\theta} \\ 0_{1 \times n_w} & 0 & 0_{1 \times n_s} \\ \Pi_0^{\theta w} & 0_{n_s \times 1} & \Pi_0^{\theta\theta} \end{bmatrix}$$

and it is clear that $E(\tilde{x}(0)[v_m^{\mathrm{T}}(k) \;\; v_s(k)]) = 0_{n_w+1+n_s \times n_v+1}$.

Now, let us define

$$\Psi = \begin{bmatrix} Z_{n_w+1} & 0_{n_w+1 \times n_s} \\ 0_{n_s \times n_w+1} & I_{n_s} \end{bmatrix} \tag{53}$$

then it is straightforward to verify that

$$\tilde{A}_{k+1}\Psi + \Delta_k^a = \Psi\tilde{A}_k \tag{54}$$

$$\tilde{G} = \Psi\tilde{G} \tag{55}$$

$$\tilde{C}_k = \tilde{C}_{k+1}\Psi + \Delta_k^c \tag{56}$$

$$\Delta_k^a = \begin{bmatrix} 0_{n_w+1 \times n_w+1} & 0_{n_w+1 \times n_s} \\ [0_{n_s \times n_w} & B_s r(k - n_w)] \quad 0_{n_s \times n_s} \end{bmatrix} \tag{57}$$

$$\Delta_k^c = [0_{1 \times n_w} \quad D_s r(k - n_w) \quad 0_{1 \times n_s}] \tag{58}$$

where (54)–(56) are, up to the Δ-terms, equal to (a special case of) the relations in Reference [12].

3.2. *The fast-array iterations*

The Kalman filter equations of the augmented system (46)–(47) are given by

$$\hat{\tilde{x}}(0) = 0_{n_w+1+n_s \times 1} \tag{59}$$

$$\tilde{P}_0 = \tilde{\Pi}_0 \tag{60}$$

$$\tilde{\varepsilon}(k) = e(k) - \tilde{C}_k\hat{\tilde{x}}(k) - D_k\hat{w}(k) \tag{61}$$

$$\tilde{R}_{e,k} = R + \tilde{C}_k\tilde{P}_k\tilde{C}_k^{\mathrm{T}} \tag{62}$$

$$\tilde{K}_k = \tilde{A}_k\tilde{P}_k\tilde{C}_k^{\mathrm{T}} \tag{63}$$

$$\hat{\tilde{x}}(k+1) = \tilde{A}_k\hat{\tilde{x}}(k) + \tilde{B}_k\hat{w}(k) + \tilde{K}_k\tilde{R}_{e,k}^{-1}\varepsilon(k) \tag{64}$$

$$\tilde{P}_{k+1} = \tilde{A}_k\tilde{P}_k\tilde{A}_k^{\mathrm{T}} - \tilde{K}_k\tilde{R}_{e,k}^{-1}\tilde{K}_k^{\mathrm{T}} + \tilde{G}Q\tilde{G}^{\mathrm{T}} \tag{65}$$

Because the augmented system (46)–(47) is equivalent to the original system (26)–(27), the Kalman filter of the augmented system should provide the same state-estimate, state-error covariance, and innovations. By straightforward computation, it can be verified (e.g. by induction) that for all $k \geqslant 0$:

$$\begin{aligned} \tilde{P}_k &= \begin{bmatrix} P_k^{ww} & 0_{n_w \times 1} & P_k^{w\theta} \\ 0_{1 \times n_w} & 0 & 0_{1 \times n_s} \\ P_k^{\theta w} & 0_{n_s \times 1} & P_k^{\theta\theta} \end{bmatrix} \\ \tilde{R}_{e,k} &= R_{e,k} \\ \tilde{K}_k &= \begin{bmatrix} K_k^w \\ 0 \\ K_k^\theta \end{bmatrix} \end{aligned}$$

Hence, we also have for all $k \geqslant 0$:

$$\begin{aligned} \hat{\tilde{x}}(k) &= \begin{bmatrix} \hat{w}(k) \\ 0 \\ \hat{\theta}(k) \end{bmatrix} \\ \tilde{\varepsilon}(k) &= \varepsilon(k) \end{aligned}$$

The idea behind fast-array algorithms is to update the difference

$$\mathrm{d}\tilde{P}_k = \tilde{P}_k - \Psi \tilde{P}_{k-1} \Psi^{\mathrm{T}}$$

rather than $\tilde{P}_k$ itself. In many cases, depending on the choice of Π_0, it can be shown that $\mathrm{d}\tilde{P}_k$ has a low rank α with $\alpha \ll (n_w + 1 + n_s)$ (in the next subsection, we will exhibit a choice for Π_0 such that $\alpha = 2$). Hence, $\mathrm{d}\tilde{P}_k$, or a factorization for it, can be updated very efficiently [11, 12].

Let us define also the difference quantities

$$\begin{aligned} \mathrm{d}\tilde{R}_{e,k} &= \tilde{R}_{e,k} - \tilde{R}_{e,k-1} \\ \mathrm{d}\tilde{K}_k &= \tilde{K}_k - \Psi \tilde{K}_{k-1} \end{aligned}$$

Then, using (62), (63) and (65) together with relations (54)–(56) we get

$$\mathrm{d}\tilde{R}_{e,k} = \tilde{C}_k \, \mathrm{d}\tilde{P}_k \tilde{C}_k^{\mathrm{T}} \tag{66}$$

$$\mathrm{d}\tilde{K}_k = \tilde{A}_k \, \mathrm{d}\tilde{P}_k \tilde{C}_k^{\mathrm{T}} \tag{67}$$

$$\mathrm{d}\tilde{P}_{k+1} = \tilde{A}_k \, \mathrm{d}\tilde{P}_k \tilde{A}_k^{\mathrm{T}} + \Psi \tilde{K}_{k-1} \tilde{R}_{e,k-1}^{-1} \tilde{K}_{k-1}^{\mathrm{T}} \Psi^{\mathrm{T}} - \tilde{K}_k \tilde{R}_{e,k}^{-1} \tilde{K}_k^{\mathrm{T}} \tag{68}$$

Suppose $\mathrm{d}\tilde{P}_k$ can be factored as

$$\mathrm{d}\tilde{P}_k = \tilde{L}_{k-1} M_{k-1} \tilde{L}_{k-1}^{\mathrm{T}}$$

where $\tilde{L}_{k-1} \in \mathbb{R}^{n_w+1+n_s \times \alpha}$ and $M_{k-1} \in \mathbb{R}^{\alpha \times \alpha}$ for some $\alpha \ll n_w + n_s + 1$. Then it turns out that $\tilde{R}_{e,k-1}$, $\tilde{K}_{k-1}$ and $\tilde{L}_{k-1}$ can be updated to $\tilde{R}_{e,k}$, $\tilde{K}_k$ and $\tilde{L}_k$ as follows. Multiply the pre-array on

the left-hand side below by a transformation Θ_{k-1}

$$\begin{bmatrix} \tilde{R}_{e,k-1}^{1/2} & \tilde{C}_k\tilde{L}_{k-1} \\ \Psi\tilde{K}_{k-1}\tilde{R}_{e,k-1}^{-T/2} & \tilde{A}_k\tilde{L}_{k-1} \end{bmatrix}\Theta_{k-1} = \begin{bmatrix} \tilde{R}_{e,k}^{1/2} & 0_{1\times\alpha} \\ \tilde{K}_k\tilde{R}_{e,k}^{-T/2} & \tilde{L}_k \end{bmatrix} \tag{69}$$

so as to result in the $1 \times \alpha$ zero block in the post-array on the right hand side. The matrix Θ_{k-1} is required to be J_{k-1}-unitary, i.e. it should satisfy

$$\Theta_{k-1}J_{k-1}\Theta_{k-1}^{\mathrm{T}} = J_{k-1}$$

where

$$J_{k-1} = \begin{bmatrix} 1 & 0_{1\times\alpha} \\ 0_{\alpha\times 1} & M_{k-1} \end{bmatrix}$$

This fact can be verified by 'squaring' the left and the right hand sides of (69) and using relations (66)–(68). By 'squaring' the left hand side of (69) we obtain

$$\begin{bmatrix} \tilde{R}_{e,k-1}^{1/2} & \tilde{C}_k\tilde{L}_{k-1} \\ \Psi\tilde{K}_{k-1}\tilde{R}_{e,k-1}^{-T/2} & \tilde{A}_k\tilde{L}_{k-1} \end{bmatrix} \underbrace{\Theta_{k-1}J_{k-1}\Theta_{k-1}^{\mathrm{T}}}_{=J_{k-1}} \begin{bmatrix} \tilde{R}_{e,k-1}^{1/2} & \tilde{C}_k\tilde{L}_{k-1} \\ \Psi\tilde{K}_{k-1}\tilde{R}_{e,k-1}^{-T/2} & \tilde{A}_k\tilde{L}_{k-1} \end{bmatrix}^{\mathrm{T}}$$

$$= \begin{bmatrix} \tilde{R}_{e,k-1} + \tilde{C}_k\tilde{L}_{k-1}M_{k-1}\tilde{L}_{k-1}^{\mathrm{T}}\tilde{C}_k^{\mathrm{T}} & \tilde{K}_{k-1}^{\mathrm{T}}\Psi^{\mathrm{T}} + \tilde{C}_k\tilde{L}_{k-1}M_{k-1}\tilde{L}_{k-1}^{\mathrm{T}}\tilde{A}_k^{\mathrm{T}} \\ \Psi\tilde{K}_{k-1} + \tilde{A}_k\tilde{L}_{k-1}M_{k-1}\tilde{L}_{k-1}^{\mathrm{T}}\tilde{C}_k^{\mathrm{T}} & \Psi\tilde{K}_{k-1}\tilde{R}_{e,k-1}^{-1}\tilde{K}_{k-1}^{\mathrm{T}}\Psi^{\mathrm{T}} + \tilde{A}_k\tilde{L}_{k-1}M_{k-1}\tilde{L}_{k-1}^{\mathrm{T}}\tilde{A}_k^{\mathrm{T}} \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{R}_{e,k} & \tilde{K}_k^{\mathrm{T}} \\ \tilde{K}_k & \tilde{A}_k\mathrm{d}\tilde{P}_k\tilde{A}_k^{\mathrm{T}} + \Psi\tilde{K}_{k-1}\tilde{R}_{e,k-1}^{-1}\tilde{K}_{k-1}^{\mathrm{T}}\Psi^{\mathrm{T}} \end{bmatrix}$$

And, on the other hand, by 'squaring' the right hand side of (69) we obtain

$$\begin{bmatrix} \tilde{R}_{e,k}^{1/2} & 0_{1\times\alpha} \\ \tilde{K}_k\tilde{R}_{e,k}^{-T/2} & \tilde{L}_k \end{bmatrix} J_{k-1} \begin{bmatrix} \tilde{R}_{e,k}^{1/2} & 0_{1\times\alpha} \\ \tilde{K}_k\tilde{R}_{e,k}^{-T/2} & \tilde{L}_k \end{bmatrix}^{\mathrm{T}}$$

$$= \begin{bmatrix} \tilde{R}_{e,k} & \tilde{K}_k^{\mathrm{T}} \\ \tilde{K}_k & \tilde{K}_k\tilde{R}_{e,k}^{-1}\tilde{K}_k^{\mathrm{T}} + \tilde{L}_kM_{k-1}\tilde{L}_k^{\mathrm{T}} \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{R}_{e,k} & \tilde{K}_k^{\mathrm{T}} \\ \tilde{K}_k & \tilde{K}_k\tilde{R}_{e,k}^{-1}\tilde{K}_k^{\mathrm{T}} + \mathrm{d}\tilde{P}_{k+1} \end{bmatrix}$$

Equality (69) then holds once we make the identification

$$\mathrm{d}\tilde{P}_{k+1} = \tilde{L}_kM_k\tilde{L}_k^{\mathrm{T}} \quad \text{with } M_k = M_{k-1} \tag{70}$$

We thus conclude that if $\mathrm{d}\tilde{P}_k$ has (low) rank α, then $\mathrm{d}\tilde{P}_{k+1}$ also has (low) rank α. Furthermore, the matrix M_k in the factorization of $\mathrm{d}\tilde{P}_{k+1}$ is equal to M_{k-1} and thus we may set

$$M = M_k \quad \text{and} \quad J = \begin{bmatrix} 1 & 0_{1\times\alpha} \\ 0_{\alpha\times 1} & M \end{bmatrix} = J_k$$

for all $k \geqslant 0$.

Now, the problem is to determine an initial factorization

$$\mathrm{d}\tilde{P}_0 = \tilde{L}_{-1} M \tilde{L}_{-1}^{\mathrm{T}}$$

with $M \in \mathbb{R}^{\alpha\times\alpha}$ and α as small as possible. This problem will be solved in the next subsection.

Note that the update equation (69) is independent of R, G and Q. These parameters enter into the initialization of the algorithm.

3.3. Initialization

We now seek a matrix Π_{-1} such that the difference

$$\tilde{P}_0 - \Psi \tilde{P}_{-1} \Psi^{\mathrm{T}} = \tilde{L}_{-1} M \tilde{L}_{-1}$$

has low rank $\alpha \ll (n_w + 1 + n_s)$. Note that since we iterate beginning from $k = 0$, we only need to know $\tilde{L}_{-1}$ and M. In the following, we will assume the *prewindowed-data* case, i.e.

$$r(k) = 0, \quad -n_w - 1 \leqslant k \leqslant -1 \tag{71}$$

and thus

$$\tilde{A}_{-1} = \begin{bmatrix} \lambda^{-1/2} I_{n_w+1} & 0_{n_w+1\times n_s} \\ 0_{n_s\times n_w+1} & A_s \end{bmatrix} \quad \text{and} \quad \tilde{C}_{-1} = [0_{1\times n_w+1} \quad C_s]$$

Then, according to (62), (63), (65) and

$$\tilde{P}_k = \begin{bmatrix} P_k^{ww} & 0_{n_w\times 1} & P_k^{w\theta} \\ 0_{1\times n_w} & 0 & 0_{1\times n_s} \\ P_k^{\theta w} & 0_{n_s\times 1} & P_k^{\theta\theta} \end{bmatrix}$$

we get

$$\begin{aligned} \tilde{P}_0 &= \tilde{A}_{-1}\tilde{P}_{-1}\tilde{A}_{-1}^{\mathrm{T}} - \tilde{K}_{-1}\tilde{R}_{e,-1}^{-1}\tilde{K}_{-1}^{\mathrm{T}} + \tilde{G}Q\tilde{G}^{\mathrm{T}} \\ &= \begin{bmatrix} \lambda^{-1}P_{-1}^{ww} & 0_{n_w\times 1} & \lambda^{-1/2}P_{-1}^{w\theta}A_s^{\mathrm{T}} \\ 0_{1\times n_w} & 0 & 0_{1\times n_s} \\ \lambda^{-1/2}A_s P_{-1}^{\theta w} & 0_{n_s\times 1} & A_s P_{-1}^{\theta\theta} A_s^{\mathrm{T}} \end{bmatrix} - \begin{bmatrix} \lambda^{-1/2}P_{-1}^{w\theta}C_s^{\mathrm{T}} \\ 0 \\ A_s P_{-1}^{\theta\theta} C_s^{\mathrm{T}} \end{bmatrix} (R + C_s P_{-1}^{\theta\theta} C_s^{\mathrm{T}})^{-1} \\ &\quad \begin{bmatrix} \lambda^{-1/2}P_{-1}^{w\theta}C_s^{\mathrm{T}} \\ 0 \\ A_s P_{-1}^{\theta\theta} C_s^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} + \begin{bmatrix} 0_{n_w\times n_w} & 0_{n_w\times 1} & 0_{n_w\times n_s} \\ 0_{1\times n_w} & 0 & 0_{1\times n_s} \\ 0_{n_s\times n_w} & 0_{n_s\times 1} & G_s Q G_s^{\mathrm{T}} \end{bmatrix} \end{aligned}$$

For simplicity, we set the 1-2 and 2-1 blocks in P_{-1} to zero, i.e.

$$P_{-1} = \begin{bmatrix} \Pi_{-1}^{ww} & 0_{n_w\times n_s} \\ 0_{n_s\times n_w} & \Pi_{-1}^{\theta\theta} \end{bmatrix} = \begin{bmatrix} P_{-1}^{ww} & 0_{n_w\times n_s} \\ 0_{n_s\times n_w} & P_{-1}^{\theta\theta} \end{bmatrix}$$

with $\Pi_{-1}^{ww} > 0$ and $\Pi_{-1}^{\theta\theta} > 0$ to be determined. Then, we get

$$\tilde{P}_0 - \Psi\tilde{P}_{-1}\Psi^{\mathrm{T}} = \left[\begin{array}{c} \lambda^{-1}\begin{bmatrix}\Pi_{-1}^{ww} & 0_{n_w\times 1}\\ 0_{1\times n_w} & 0\end{bmatrix} - Z_{n_w+1}\begin{bmatrix}\Pi_{-1}^{ww} & 0_{n_w\times 1}\\ 0_{1\times n_w} & 0\end{bmatrix}Z_{n_w+1}^{\mathrm{T}} \\ \hline 0_{n_s\times n_w+1} \end{array}\right.$$

$$\left.\begin{array}{c} 0_{n_w+1\times n_s} \\ \hline A_s\Pi_{-1}^{\theta\theta}A_s^{\mathrm{T}} - A_s\Pi_{-1}^{\theta\theta}C_s^{\mathrm{T}}(R + C_s\Pi_{-1}^{\theta\theta}C_s^{\mathrm{T}})^{-1}C_s\Pi_{-1}^{\theta\theta}A_s^{\mathrm{T}} + G_sQG_s^{\mathrm{T}} - \Pi_{-1}^{\theta\theta} \end{array}\right]$$

Let us choose

$$\Pi_{-1}^{ww} = \delta\,\mathrm{diag}\{\lambda, \lambda^2, \ldots, \lambda^{n_w}\} \tag{72}$$

which yields

$$\lambda^{-1}\begin{bmatrix}\Pi_{-1}^{ww} & 0_{n_w\times 1}\\ 0_{1\times n_w} & 0\end{bmatrix} - Z_{n_w+1}\begin{bmatrix}\Pi_{-1}^{ww} & 0_{n_w\times 1}\\ 0_{1\times n_w} & 0\end{bmatrix}Z_{n_w+1}^{\mathrm{T}} = \delta\cdot\begin{bmatrix}1 & 0_{1\times n_w-1} & 0\\ 0_{n_w-1\times 1} & 0_{n_w-1\times n_w-1} & 0_{n_w-1\times 1}\\ 0 & 0_{1\times n_w-1} & -\lambda^{n_w}\end{bmatrix}$$

Furthermore, if, in addition to observability of the pair (A_s, C_s), the pair $(A_s, G_sQ^{1/2})$ is stabilizable, then there exists a unique $\Pi_{-1}^{\theta\theta} > 0$, such that the discrete algebraic Riccati equation (DARE)

$$A_s\Pi_{-1}^{\theta\theta}A_s^{\mathrm{T}} - A_s\Pi_{-1}^{\theta\theta}C_s^{\mathrm{T}}(R + C_s\Pi_{-1}^{\theta\theta}C_s^{\mathrm{T}})^{-1}C_s\Pi_{-1}^{\theta\theta}A_s^{\mathrm{T}} + G_sQG_s^{\mathrm{T}} - \Pi_{-1}^{\theta\theta} = 0_{n_s\times n_s} \tag{73}$$

holds [15, Theorem E.6.2, p. 786]. Hence, $\Pi_{-1}^{\theta\theta}$ can be found by solving the DARE (73), and the loss of freedom in choosing $\Pi_{-1}^{\theta\theta}$ is the price to be paid for the *fast*-array algorithm. Note that to ensure the pair $(A_s, G_sQ^{1/2})$ is unit-circle controllable Q should be positive definite, $Q > 0$. Let Π_{-1}^{ww} and $\Pi_{-1}^{\theta\theta}$ satisfy (72) and (73), respectively, then we have

$$\begin{aligned}\tilde{P}_0 - \Psi\tilde{P}_{-1}\Psi^{\mathrm{T}} &= \delta\begin{bmatrix}1 & & & \\ & 0_{n_w-1\times n_w-1} & & \\ & & -\lambda^{n_w} & \\ & & & 0_{n_s\times n_s}\end{bmatrix}\\ &= \tilde{L}_{-1}M\tilde{L}_{-1}^{\mathrm{T}}\end{aligned} \tag{74}$$

where

$$\tilde{L}_{-1} = \sqrt{\delta}\begin{bmatrix}1 & 0\\ 0_{n_w-1\times 1} & 0_{n_w-1\times 1}\\ 0 & \lambda^{n_w/2}\\ 0_{n_s\times 1} & 0_{n_s\times 1}\end{bmatrix}, \quad M = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \tag{75}$$

Hence, we have obtained a factorization for $\mathrm{d}\tilde{P}_0$ with rank $\alpha = 2$. Note that now we have

$$J = (1 \oplus M) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Due to the -1 in the 3–3 element of J, the transformations Θ_{k-1} in (69) are *hyperbolic*. For comments on the implementation and numerical accuracy of these rotations, we refer to Reference [14, Chapter 14], see, also Reference [11, Section 2]. The resulting fast-array implementation of the Kalman algorithm, is listed in the second column of Table I.

4. COMPARISON WITH MODIFIED FILTERED-RLS

Figures 2 and 3 show the block diagrams of the filtered-RLS and the modified filtered-RLS algorithm respectively. Because the adaptive filter $\hat{W}_k(q^{-1})$ significantly varies in time, the adaptive filter and the secondary path system may not be interchanged as assumed in the filtered-RLS algorithm. For this reason, the *modified* filtered-RLS algorithm has been proposed [2], which shows better convergence. In the filtered-RLS algorithm the reference signal $r(k)$ is

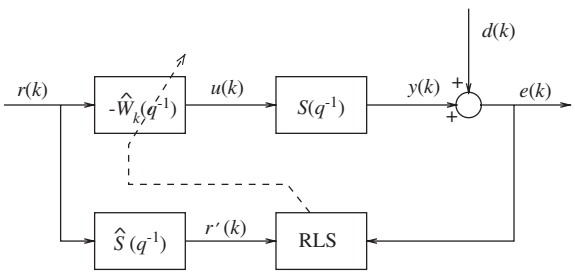

Figure 2. Block diagram of the filtered-RLS algorithm.

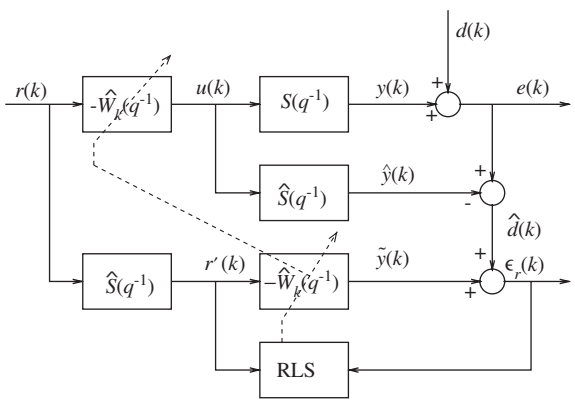

Figure 3. Block diagram of the modified filtered-RLS algorithm.

replaced by a *filtered* reference signal $r'(k)$ that is generated by‡

$$\theta'_r(k+1) = \lambda^{1/2} A_s \theta'_r(k) + \lambda^{1/2} B_s r(k), \quad \theta'_r(0) = 0_{n_s \times 1} \tag{76}$$

$$r'(k) = C_s \theta'_r(k) + D_s r(k) \tag{77}$$

Note that in the expression for $\theta'_r(k+1)$ we are using the exponential forgetting factor λ; this choice for generating $r'(k)$ is motivated by the proof of Theorem 2. The adaptive filter $\hat{W}_k(q^{-1})$ is tuned by the RLS algorithm such that the error

$$\varepsilon_r(k) = \hat{d}(k) + \tilde{y}(k) \tag{78}$$

is minimized, where $\hat{d}(k)$ is the estimated disturbance determined by

$$\hat{\theta}_r(k+1) = A_s \hat{\theta}_r(k) + B_s u(k), \quad \hat{\theta}_r(0) = 0_{n_s \times 1} \tag{79}$$

$$\hat{y}(k) = C_s \hat{\theta}_r(k) + D_s u(k) \tag{80}$$

$$\hat{d}(k) = e(k) - \hat{y}(k) \tag{81}$$

and $\tilde{y}(k)$ is the output of the adaptive filter given by

$$\tilde{y}(k) = -r'^{\mathrm{T}}_{n_w}(k) \hat{w}_r(k) \tag{82}$$

where

$$\hat{w}_r = [\hat{w}_0(k) \ \hat{w}_1(k) \cdots \hat{w}_{n_w-1}(k)]^{\mathrm{T}} \tag{83}$$

$$r'_{n_w}(k) = [r'(k) \ r'(k-1) \cdots r'(k-n_w+1)]^{\mathrm{T}} \tag{84}$$

Table II lists the modified Filtered-RLS algorithm in its standard covariance and fast-array forms, which are derived according to Reference [11]. The computational complexity of the modified Filtered-RLS algorithm can be reduced further by using the Fast Transversal Filter (FTF), see Reference [14, Chapter 14], but often at the expense of numerical accuracy. The derivation of the modified RLS algorithm is quite *ad hoc*, and no systematic derivation of the modification and conditions for its optimality have been given yet.

In this section, we will compare the Kalman algorithm with the modified RLS algorithm. Our main result in this section is that the modified RLS algorithm is a special case of the Kalman algorithm of the previous section when there is no uncertainty on the secondary-path state (due to initial-state uncertainty and/or noise). By showing the equivalence, we have thus provided a systematic derivation of the modified filtered-RLS algorithm and conditions for its optimality.

Theorem 2

The Kalman algorithm listed in Table I and the modified filtered-RLS algorithm listed in Table II are equivalent, under the condition that

$$\Pi_0^{\theta\theta} = 0_{n_s \times n_s}, \quad \Pi_0^{w\theta} = \Pi_0^{\theta w^{\mathrm{T}}} = 0_{n_w \times n_s}, \quad Q = 0_{n_v \times n_v}$$

in the Kalman algorithm and $r(k) = 0$ for $-n_w \leqslant k \leqslant 0$.

‡Variables with subscript $_r$ or superscript ^{r} refer the variables from the modified filtered-RLS algorithm (in order to prevent confusion with variables from the Kalman filter algorithm of Table I).

Table II. Modified filtered-RLS algorithm in Kalman covariance and fast-array forms.

Kalman covariance form	Fast-array form
Assumptions:	
$0 \ll \lambda \leqslant 1$	idem
$\Pi_0^{ww} > 0$	$\Pi_{-1}^{ww} = \sqrt{\delta} \cdot \mathrm{diag}\{\lambda, \lambda^2, \ldots, \lambda^{n_w}\},\ \delta > 0$ $r'(k) = 0$, for $-n_w - 1 \leqslant k \leqslant -1$ (i.e. prewindowed data)
Initialization:	
$\begin{cases} \hat{\theta}_r(0) = \theta'_r(0) = 0_{n_s \times 1} \\ \hat{w}_r(0) = 0_{n_w \times 1} \end{cases}$	idem
$\begin{cases} r_{n_w}(-1) = [r(-1)\ \ r(-2)\ \ \cdots\ \ r(-n_w)]^{\mathrm{T}} \\ r'_{n_w}(-1) = 0_{n_w \times 1} \end{cases}$	$\begin{cases} \text{idem} \\ r'_{n_w+1}(-1) = 0_{n_w+1 \times 1} \end{cases}$
$P_0^r = \Pi_0^{ww}$	$\begin{cases} \tilde{L}^r_{-1} = \sqrt{\delta} \begin{bmatrix} 1 & 0 \\ 0_{n_w-1 \times 1} & 0_{n_w-1 \times 1} \\ 0 & \lambda^{n_w/2} \end{bmatrix} \\ \bar{K}^r_{-1} = 0 \\ R^r_{e,-1} 1/2 = R^{1/2} \end{cases}$
Iterate for $k \geqslant 0$:	
$\begin{cases} \theta'_r(k+1) = \begin{cases} A_s\theta'_r(k) + B_s r(k) & \text{(standard)} \\ \lambda^{1/2} A_s \theta'_r(k) + \lambda^{1/2} B_s r(k) & \text{(new)} \end{cases} \\ r'(k) = C_s \theta'_r(k) + D_s r(k) \end{cases}$	idem
$\begin{cases} r_{n_w}(k) = [r(k)\ \ r^{\mathrm{T}}_{n_w-1}(k-1)]^{\mathrm{T}} \\ r'_{n_w}(k) = [r'(k)\ \ r'^{\mathrm{T}}_{n_w-1}(k-1)]^{\mathrm{T}} \end{cases}$	$\begin{cases} \text{idem} \\ r'_{n_w+1}(k) = [r'(k)\ \ r'^{\mathrm{T}}_{n_w}(k-1)]^{\mathrm{T}} \end{cases}$
$\begin{cases} \hat{\theta}_r(k+1) = A_s\hat{\theta}_r(k) - B_s r^{\mathrm{T}}_{n_w}(k)\hat{w}_r(k) \\ \hat{y}(k) = C_s\hat{\theta}_r(k) - D_s r^{\mathrm{T}}_{n_w}(k)\hat{w}_r(k) \end{cases}$	idem
$\begin{cases} \tilde{y}(k) = -r'^{\mathrm{T}}_{n_w}(k)\hat{w}_r(k) \\ \varepsilon_r(k) = e(k) - \hat{y}(k) + \tilde{y}(k) \end{cases}$	idem
$\begin{cases} K_k^r = \lambda^{-1/2} P_k^r r'_{n_w}(k) \\ R^r_{e,k} = R^r + r'^{\mathrm{T}}_{n_w}(k) P_k^r r'_{n_w}(k) \\ P^r_{k+1} = \lambda^{-1} P_k^r - K_k^r R^r_{e,k-1} K_k^{r\mathrm{T}} \end{cases}$	$\begin{cases} \text{Perform } J\text{-unitary rotation to make the } 1-2 \text{ block in the post-array equal to zero, } J = (I_2 \oplus -1), \Theta_{k-1} J \Theta^{\mathrm{T}}_{k-1} = J \\ \begin{bmatrix} R^{r1/2}_{e,k-1} & r'^{\mathrm{T}}_{n_w+1}(k)\tilde{L}^r_{k-1} \\ \begin{bmatrix} 0 \\ \bar{K}^r_{k-1} \end{bmatrix} & \lambda^{-1/2}\tilde{L}^r_{k-1} \end{bmatrix} \Theta_{k-1} = \begin{bmatrix} R^{r1/2}_{e,k} & 0_{1\times 2} \\ \begin{bmatrix} \bar{K}^r_k \\ 0 \end{bmatrix} & \tilde{L}^r_k \end{bmatrix} \end{cases}$
$\hat{w}_r(k+1) = \lambda^{-1/2}\hat{w}_r(k) + K_k^r R^r_{e,k-1} \varepsilon_r(k)$	$\hat{w}_r(k+1) = \lambda^{-1/2}\hat{w}_r(k) + \bar{K}_k^r R^{r-1/2}_{e,k} \varepsilon_r(k)$

Proof

First, let us define

$$\bar{S}_{k+1} = \lambda^{1/2} A_s \bar{S}_k + \lambda^{1/2} B_s r^{\mathrm{T}}_{n_w}(k), \quad \bar{S}_0 = 0_{n_s \times n_w} \tag{85}$$

Using expression (76) and the expression for $r_{n_w}(k)$ from Table II it can be verified that

$$\bar{S}_k = [\theta_r'(k)\ \theta_r'(k-1)\cdots\theta_r'(k-n_w+1)] \tag{86}$$

Substituting this result into expressions (77), (82) and (84) yields

$$r_{n_w}'^{\mathrm{T}}(k) = C_s\bar{S}_k + D_s r_{n_w}^{\mathrm{T}}(k) \tag{87}$$

$$\tilde{y}(k) = -C_s\bar{S}_k\hat{w}_r(k) - D_s r_{n_w}^{\mathrm{T}}(k)\hat{w}_r(k) \tag{88}$$

and thus

$$\hat{y}(k) - \tilde{y}(k) = C_s\hat{\theta}_r(k) + C_s\bar{S}_k\hat{w}_r(k) \tag{89}$$

The relations just derived will be used in the sequel in the proof.

By induction the following relations can be verified:

$$\text{KALMAN}: \quad \text{RLS}:$$

$$\hat{w}(k) = \hat{w}_r(k) \tag{90}$$

$$\hat{\theta}(k) = \hat{\theta}_r(k) + \bar{S}_k\hat{w}_r(k) \tag{91}$$

$$R_{e,k} = R_{e,k}^r \tag{92}$$

$$K_k^w = K_k^r \tag{93}$$

$$K_k^\theta = \bar{S}_{k+1}K_k^r \tag{94}$$

$$P_k = \begin{bmatrix} P_k^{ww} & P_k^{w\theta} \\ P_k^{\theta w} & P_k^{\theta\theta} \end{bmatrix} = \begin{bmatrix} I_{n_w} \\ \bar{S}_k \end{bmatrix} P_k^r \begin{bmatrix} I_{n_w} & \bar{S}_k^{\mathrm{T}} \end{bmatrix} \tag{95}$$

The first step is to show that (90)–(95) hold for $k = 0$, which can be verified readily from the initialization of the algorithm from Tables I and II and the assumptions in the theorem. Note that if (91) holds, then also $\varepsilon(k) = \varepsilon_r(k)$ holds. Further, since $\bar{S}_0 = 0_{n_s\times n_w}$ equivalence (95) yields

$$P_0 = \begin{bmatrix} \Pi_0^{ww} & \Pi_0^{w\theta} \\ \Pi_0^{\theta w} & \Pi_0^{\theta\theta} \end{bmatrix} = \begin{bmatrix} \Pi_0^{ww} & 0_{n_w\times n_s} \\ 0_{n_s\times n_w} & 0_{n_s\times n_s} \end{bmatrix}$$

which is the reason to assume $\Pi_0^{\theta\theta}$, $\Pi_0^{w\theta}$ and $\Pi_0^{\theta w}$ to be zero in the theorem.

The second step is to show that if (90)–(95) hold for k, then (90)–(95) also hold for $k+1$. Assume (90)–(95) hold for k. That (90) holds for $k+1$ directly follows from (91) (i.e. $\varepsilon(k) = \varepsilon_r \times (k)$), (92) and (93) and the update rules of $w(k)$ and $w_r(k)$.

To show that (91) holds for $k+1$, we write

$$\begin{aligned}\hat{\theta}(k+1) &= A_s\hat{\theta}(k) + \bar{S}_{k+1}K_k^r R_{e,k}^{r-1}\varepsilon_r(k) \\ &= A_s(\hat{\theta}_r(k) + \bar{S}_k\hat{w}_r(k)) + \bar{S}_{k+1}K_k^r R_{e,k}^{r-1}\varepsilon_r(k) \\ &= A_s(\hat{\theta}_r(k) + \bar{S}_k\hat{w}_r(k)) + \bar{S}_{k+1}(\hat{w}_r(k+1) - \lambda^{-1/2}\hat{w}_r(k))\end{aligned}$$

On the other hand, we have

$$\begin{aligned}
&\hat{\theta}_r(k+1) + \bar{S}_{k+1}\hat{w}_r(k+1) \\
&= A_s\hat{\theta}_r(k) - B_s r_{n_w}^{\mathrm{T}}(k)\hat{w}_r(k) + \lambda^{1/2}A_s\bar{S}_k\hat{w}_r(k+1) + \lambda^{1/2}B_s r_{n_w}^{\mathrm{T}}(k)\hat{w}_r(k+1) \\
&= A_s(\hat{\theta}_r(k) + \bar{S}_k\hat{w}_r(k)) + (\lambda^{1/2}A_s\bar{S}_k + \lambda^{1/2}r_{n_w}^{\mathrm{T}}(k))(\hat{w}_r(k+1) - \lambda^{-1/2}\hat{w}_r(k)) \\
&= A_s(\hat{\theta}_r(k) + \bar{S}_k\hat{w}_r(k)) + \bar{S}_{k+1}(\hat{w}_r(k+1) - \lambda^{-1/2}\hat{w}_r(k))
\end{aligned}$$

and thus $\hat{\theta}(k+1) = \hat{\theta}_r(k+1) + \bar{S}_{k+1}\hat{w}_r(k+1)$.

Before showing that (92)–(94) hold for $k+1$, we show that (95) holds. Using the fact that (92)–(95) hold for k and the assumption that $Q = 0$, we can write

$$\begin{aligned}
P_{k+1} &= \begin{bmatrix} \lambda^{-1/2}I_{n_w} & 0_{n_w\times n_s} \\ B_s r_{n_w}^{\mathrm{T}}(k) & A_s \end{bmatrix} \begin{bmatrix} I_{n_w} \\ \bar{S}_k \end{bmatrix} P_k^r \begin{bmatrix} I_{n_w} & \bar{S}_k^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \lambda^{-1/2}I_{n_w} & r_{n_w}(k)B_s^{\mathrm{T}} \\ 0_{n_s\times n_w} & A_s^{\mathrm{T}} \end{bmatrix} \\
&\quad - \begin{bmatrix} I_{n_w} \\ \bar{S}_{k+1} \end{bmatrix} \lambda^{-1/2}P_k^r r_{n_w}'^{\mathrm{T}}(k)(R + r_{n_w}'^{\mathrm{T}}(k)P_k^r r_{n_w}'(k))^{-1} r_{n_w}'(k)P_k^r\lambda^{-1/2} \begin{bmatrix} I_{n_w} & \bar{S}_{k+1}^{\mathrm{T}} \end{bmatrix} \\
&= \begin{bmatrix} I_{n_w} \\ \bar{S}_{k+1} \end{bmatrix} \lambda^{-1}P_k^r \begin{bmatrix} I_{n_w} \\ \bar{S}_{k+1} \end{bmatrix}^{\mathrm{T}} - \begin{bmatrix} I_{n_w} \\ \bar{S}_{k+1} \end{bmatrix} K_k^r R_{c,k}^{-1} K_k^{r\mathrm{T}} \begin{bmatrix} I_{n_w} \\ \bar{S}_{k+1} \end{bmatrix}^{\mathrm{T}} \\
&= \begin{bmatrix} I_{n_w} \\ \bar{S}_{k+1} \end{bmatrix} P_{k+1}^r \begin{bmatrix} I_{n_w} \\ \bar{S}_{k+1} \end{bmatrix}^{\mathrm{T}}
\end{aligned}$$

Thus (95) holds for $k+1$.

Using this result, we can write

$$\begin{aligned}
R_{e,k+1} &= R + (D_s r_{n_w}^{\mathrm{T}}(k+1) + C_s\bar{S}_{k+1})P_{k+1}^r(D_s r_{n_w}^{\mathrm{T}}(k+1) + C_s\bar{S}_{k+1})^{\mathrm{T}} \\
&= R + r_{n_w}'^{\mathrm{T}}(k+1)P_{k+1}^r r_{n_w}'(k+1) \\
&= R_{e,k+1}^r
\end{aligned}$$

$$\begin{aligned}
K_{k+1} = \begin{bmatrix} K_{k+1}^w \\ K_{k+1}^\theta \end{bmatrix} &= \begin{bmatrix} I_{n_w} \\ \lambda^{1/2}(B_s r_{n_w}^{\mathrm{T}}(k+1) + A_s\bar{S}_{k+1}) \end{bmatrix} \lambda^{-1/2}P_{k+1}^r \begin{bmatrix} I_{n_w} \\ \bar{S}_{k+1} \end{bmatrix}^{\mathrm{T}} [D_s r_{n_w}^{\mathrm{T}}(k+1) \quad C_s]^{\mathrm{T}} \\
&= \begin{bmatrix} I_{n_w} \\ \bar{S}_{k+2} \end{bmatrix} \lambda^{-1/2}P_{k+1}^r(D_s r_{n_w}^{\mathrm{T}}(k+1) + C_s\bar{S}_{k+1})^{\mathrm{T}} \\
&= \begin{bmatrix} I_{n_w} \\ \bar{S}_{k+2} \end{bmatrix} K_{k+1}^r
\end{aligned}$$

Thus (92)–(94) hold for $k+1$.

Hence, (90)–(95) hold for all $k \geqslant 0$ and we conclude that the Kalman algorithm of Table I and the RLS algorithm of Table II (with $\theta_r'(k+1)$ given by (76)) are equivalent under the conditions given in the theorem. □

Table III. Computational load of the fast-array RLS (Table II) and fast-array Kalman (Table I) algorithms, with the secondary-path in full state–space and FIR parameterization in number floating point additions or multiplications (neglecting terms not depending on dimensions n_w and n_s).

	RLS		Kalman	
Action:	State–space	FIR	State–space	FIR
Filtered-reference	$2n_s^2 + 3n_s$	$2n_s$	—	—
Disturbance estimate	$2n_s^2 + 3n_s$	$2n_s$	—	—
Calculation innovation	$2n_w$	$2n_w$	$2n_s$	0
Construction pre-array	$6n_w$	$6n_w$	$6n_w + 4n_s^2$	$6n_w + 2n_s$
Performing rotations	$12n_w$	$12n_w$	$12n_w + 12n_s$	$12n_w + 12n_s$
Updating coefficients/state	$3n_w$	$3n_w$	$3n_w + n_s^2 + n_s$	$3n_w + 2n_s$
Calculating control	$2n_w$	$2n_w$	$2n_w$	$2n_w$
Total	$25n_w + 4n_s^2 + 6n_s$	$25n_w + 4n_s$	$23n_w + 5n_s^2 + 15n_s$	$23n_w + 16n_s$

The effect of λ in the expression for $\theta'(k+1)$ in (76) (see also Table II) yields

$$r'(k) = D_s r(k) + \lambda^{1/2} C_s B_s r(k-1) + \lambda C_s A_s B_s r(k-2) + \lambda^{3/2} C_a A_s^2 B_s r(k-3) + \cdots$$

and thus λ has the effect of *exponential forgetting* in the generation of the filtered reference signal $r'(k)$.

Finally, Table III compares the computational complexity of the fast-array implementations of the modified filtered-RLS and the Kalman algorithm proposed in this paper. From this table, we infer that the number of floating point operations are linearly increasing with n_w. The main computational step is the evaluation of the rotations. Each elementary rotation is of the form

$$x_{\text{new}} \leftarrow \alpha(x + \rho y)$$

$$y_{\text{new}} \leftarrow \rho x_{\text{new}} - \beta y$$

which takes six floating point operations. The rotations need to be evaluated for all rows in the pre-array and by operating on the elements in the column pair 1–2 and the column pair 1–3. Alternative implementations are also possible, see, e.g. Reference [14, Chapter 14].

5. SIMULATION RESULTS

To illustrate the method, simulations are performed on an $n_s = 19$th order discrete acoustic duct system, which has been obtained by physical modelling and discretized using a sampling rate of 1 KHz. Figures 4 and 5 show the impulse response coefficients of the disturbance path and the secondary path, respectively. The signal-to-noise ratio is chosen to be 30 dB and $r(k)$ is a zero-mean white-noise signal with unit variance. The number of filter coefficients was chosen to be $n_w = 150$. Only the *fast-array* implementations contained in Tables 1 and 2 are used, with $\lambda = 1$ (no exponential forgetting). The measurement noise variance was $R = 2.1 \times 10^{-5}$. The value of δ, which determines the magnitude of the initial state covariance P_{-1}, was set to $\delta = 10^{-3}$. In the (fast-array) Kalman filter algorithm Q has chosen to be $Q = 2 \times 10^{-3}$. For comparison, also the FxLMS and the preconditioned FxLMS as proposed in Reference [5] (where the secondary path system is preconditioned by its inverse outer factor) have been used with the normalized stepsize chosen to be 0.05, optimized by trial and error. All algorithms are turned on after 1000 samples.

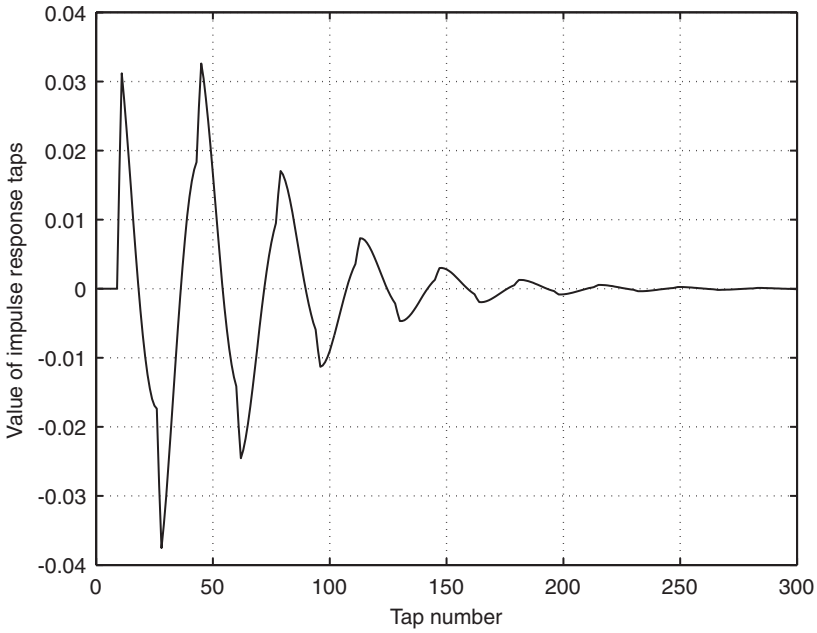

Figure 4. Impulse response of the disturbance path.

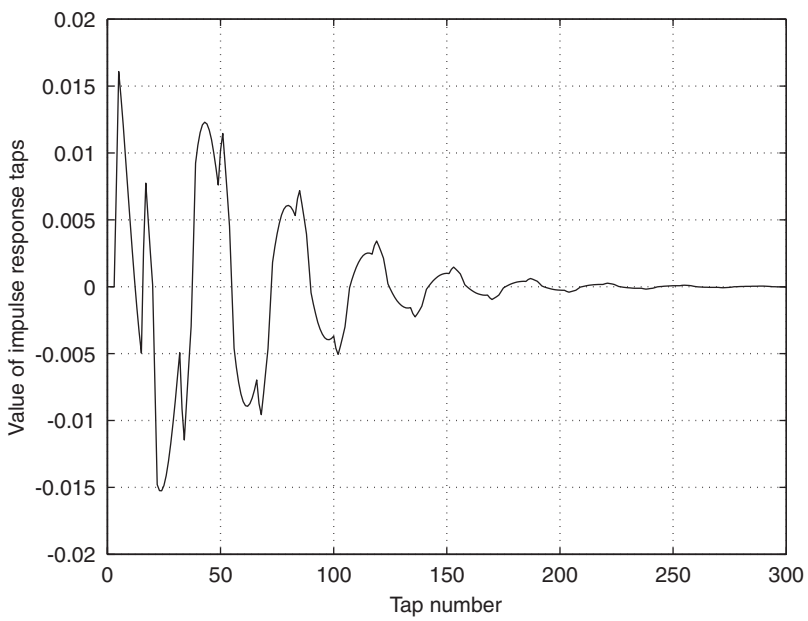

Figure 5. Impulse response of the secondary path.

Figure 6 shows the learning curves obtained by the algorithms, averaged over 50 experiments. From this figure, it can be concluded that both the RLS and the Kalman filter algorithm converge to (approximately) the same performance level. However, the RLS algorithm shows a significant overshoot directly after turning on the algorithm. This overshoot can be explained by the fact that the uncertainty in the secondary path state is not taken into account, contrary to the Kalman filter algorithm for $Q \neq 0$. The FxLMS and the preconditioned FxLMS are converging much slower, as is expected since they are based on an LMS estimated gradient update.

But, note that the computational complexity per iteration of FxLMS and preconditioned FxLMS is still lower than the computational complexity of the fast-array implementations of the RLS and Kalman filter algorithms.

The same experiment was performed by choosing δ in the RLS and the Kalman filter algorithm to be $\delta = 10^{-4}$, see Figure 7. From this figure, it is clear that the overshoot of the RLS algorithm with $\delta = 10^{-3}$ can be considerably reduced by lowering δ to $\delta = 10^{-4}$, but at the expense of convergence rate. Using $\delta = 10^{-4}$ in the Kalman algorithm, shows fast convergence at the first few hundred samples, but then its convergence rate slows down to the convergence of the RLS algorithm.

From these observations, we conclude that the overshoot or bad convergence of the RLS algorithm at startup can be prevented by the Kalman filter algorithm, since uncertainty in the secondary path state is accounted for.

To get insight in the robustness of the algorithms, the same experiment has been repeated for an erroneous secondary path model, which contains 1 sample pure delay in addition to the secondary path system. Figure 8 shows the learning curves obtained by the Kalman and the

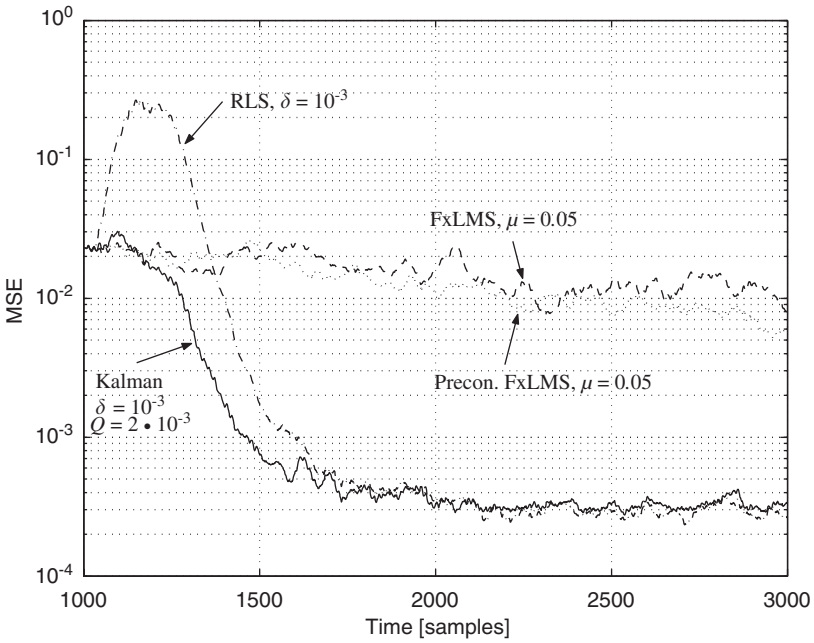

Figure 6. Learning curves obtained by the Kalman, RLS, FxLMS and the preconditioned FxLMS algorithm, averaged over 50 experiments.

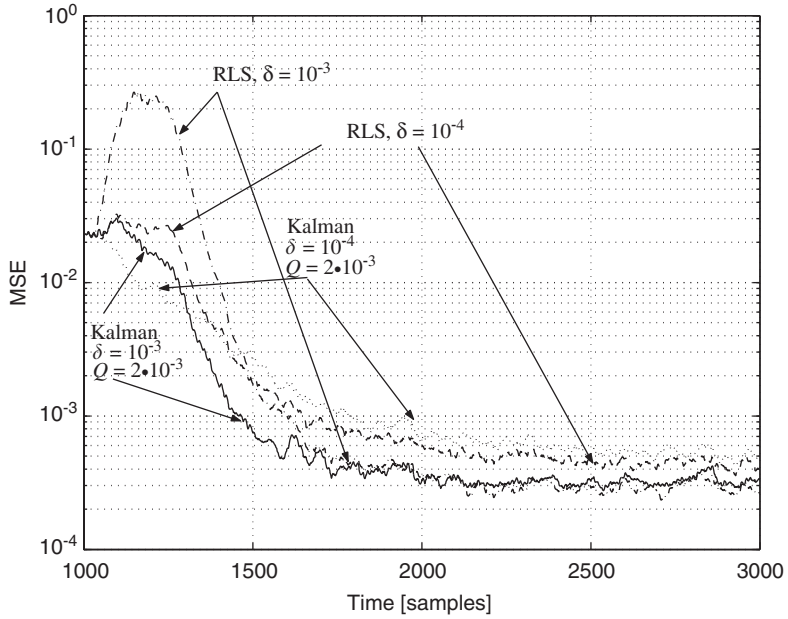

Figure 7. Learning curves obtained by the Kalman and the RLS algorithm for $\delta = 10^{-3}$ and for 10^{-4}, averaged over 50 experiments.

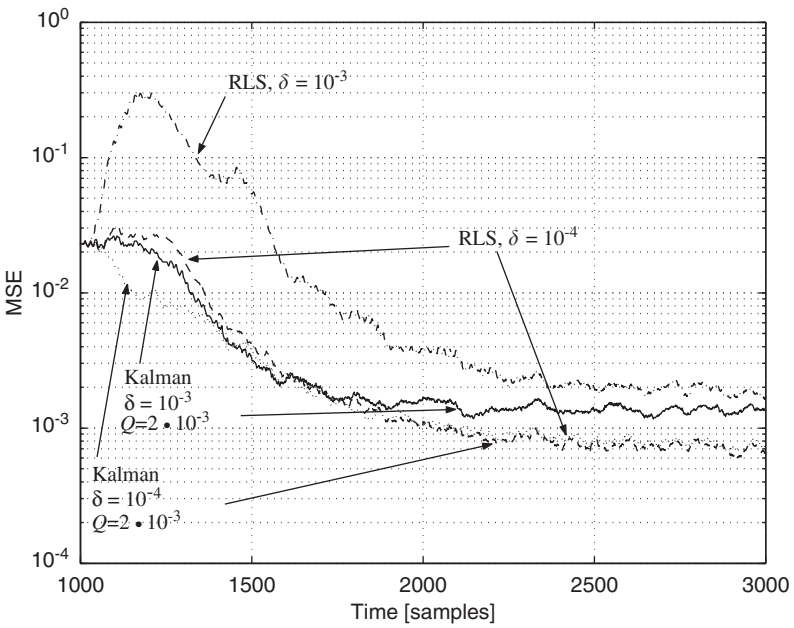

Figure 8. Learning curves obtained by the Kalman and the RLS algorithm for $\delta = 10^{-3}$ and for 10^{-4} with 1 sample delay uncertainty in the secondary path model, averaged over 50 experiments.

RLS algorithm for $\delta = 10^{-3}$ and 10^{-4}. From this figure, we observe that the algorithms are, to some extent, robust for the model uncertainty in the secondary path model, at least no divergence is obtained. However, all algorithms converge to a suboptimal solution (c.f. with the performance obtained in Figures 6 and 7), which is lower $\delta = 10^{-4}$ than for 10^{-3} for both the Kalman and the RLS algorithms. From this observation, we conclude, that the robustness w.r.t. model uncertainty (for this particular case) can be improved by lowering δ to $\delta = 10^{-4}$. But both, the Kalman and the RLS algorithm, converge to suboptimal solutions, and do not show significant different convergence behaviour for this model uncertainty (apart from the uncertainty in the secondary-path state).

The simulation examples merely demonstrate the potential of the (fast-array) Kalman filter solution. Further research will be devoted to analyse its robustness w.r.t. secondary path model errors. Such a robustness analysis is well known for the (modified) FxLMS type algorithms, for a recent contribution see Reference [16]. We will also investigate the potential of exploiting H_∞ state-estimation [17] and robust Kalman filtering [18].

6. CONCLUSIONS

The active control problem can be reformulated in state–space form, which overcomes formulating the control problem in terms of interchanging the adaptive filter and the secondary path. In this way, uncertainty due to initial-state and time-variations are taken into account explicitly. The state-estimation problem was solved by the Kalman filter and the structure in the state–space matrices was exploited to develop a fast-array implementation of the algorithm. Under the theoretical condition that there is no uncertainty in the secondary path state, it is proven that the Kalman algorithm is equivalent to the modified filtered-RLS algorithm. Hence, the Kalman algorithm can be seen as a generalization of the modified filtered-RLS algorithm. At the same time, conditions for optimality of the modified filtered-RLS algorithm are derived. When using exponential forgetting in the modified filtered-RLS algorithm, the forgetting factor should also be applied to the reference signal.

ACKNOWLEDGEMENTS

The work of A. H. Sayed was supported in part by NSF grants CCR-0208573 and ECS-0401188, the work of R. Fraanje, M. Verhaegen and N. J. Doelman was supported by the Knowledge Center 'Sound and Vibration UT-TNO', programme 'Robust Active Control', a joint initiative of TNO, Delft, The Netherlands and the University of Twente, Enschede, The Netherlands. The work of R. Fraanje and M. Verhaegen was also partially supported by the intelligent materials for Active Noise Reduction (InMAR) project funded under the 6th Framework Programme of the European Commission.

REFERENCES

1. Elliott S. *Signal Processing for Active Control*. Academic Press: London, U.K., 2001.
2. Flockton S. Fast adaptation algorithms in active noise control. In *Second Conference on Recent Advances in Active Noise Control of Sound and Vibration*, Virginia, April 1993; 802–810.
3. Rupp M, Sayed A. Robust FxLMS algorithms with improved convergence performance. *IEEE Transactions in Speech and Audio Processing* 1998; **6**(1):78–85.

4. Lopes P, Piedade M. The Kalman filter in active noise control. In *Proceedings of ACTIVE 99*, Ft. Lauderdale, FL, U.S.A., December 1999; 1111–1122.
5. Elliott S. Optimal controllers and adaptive controllers for multichannel feedforward control of stochastic disturbances. *IEEE Transactions on Signal Processing* 2000; **SP-48**(4):1053–1060.
6. Sayyar-Rodsari B, How J, Hassibi B, Carrier A. Estimation-based synthesis of H_∞-optimal adaptive FIR filters for filtered-LMS problems. *IEEE Transactions on Signal Processing* 2001; **SP-49**(1):164–178.
7. Meurers T, Veres S, Elliott S. Frequency selective feedback for active noise control. *IEEE Control Systems Magazine* 2002; **22**(4):32–41.
8. Bouchard M. Multichannel affine and fast affine projection algorithms for active noise control and acoustic equalization systems. *IEEE Transactions on Speech and Audio Processing* 2003; **SAP-11**(1):54–60.
9. Bjarnason E. Active noise cancellation using a modified form of the filtered-X LMS algorithm. In *Proceedings of the 6th European Signal Processing Conference*, vol. 2, Brussels, Belgium, August 1992; 1053–1056.
10. Doelman N. Control strategies for active noise reduction. In *Proceedings of Internoise 90*, Gothenburg, Sweden, August 1990; 1273–1276.
11. Sayed A, Kailath T. A state–space approach to adaptive RLS filtering. *IEEE Signal Processing Magazine* 1994; 18–60.
12. Sayed A, Kailath T. Extended Chandrasekhar recursions. *IEEE Transactions on Automatic Control* 1994; **AC-39**(3):619–623.
13. Rugh W. *Linear System Theory*. Prentice-Hall: Upper Saddle River, NJ, U.S.A., 1996.
14. Sayed A. *Fundamentals of Adaptive Filtering*. Wiley: New York, U.S.A., 2003.
15. Kailath T, Sayed A, Hassibi B. *Linear Estimation*. Prentice-Hall: Englewood Cliffs, NJ, U.S.A., 2000.
16. Lopes P, Piedade M. The behavior of the modified FX-LMS algorithm with secondary path modeling errors. *IEEE Signal Processing Letters* 2004; **11**(2):148–151.
17. Hassibi B, Kailath T, Sayed A. Array algorithms for H_∞-estimation. *IEEE Transactions on Automatic Control* 2000; **AC-45**(4):702–706.
18. Sayed A. A framework for state–space estimation with uncertain models. *IEEE Transactions on Automatic Control* 2001; **AC-46**(7):999–1013.

