Delft University of Technology

Fac. of Information Technology and Systems **Control Systems Engineering**

Technical report bds:99-11

State space modeling in multichannel active control systems*

G. Nijsse, M. Verhaegen, B. De Schutter, D. Westwick, and N. Doelman

If you want to cite this report, please use the following reference instead:

G. Nijsse, M. Verhaegen, B. De Schutter, D. Westwick, and N. Doelman, "State space modeling in multichannel active control systems," *Proceedings of the 1999 International Symposium on Active Control of Sound and Vibration (ACTIVE 99)* (S. Douglas, ed.), Fort Lauderdale, Florida, pp. 909–920, Dec. 1999.

Control Systems Engineering
Faculty of Information Technology and Systems
Delft University of Technology
Delft, The Netherlands
phone: +31-15-278.51.19 (secretary)
fax: +31-15-278.66.79
Current URL: `http://www.dcsc.tudelft.nl`

*This report can also be downloaded via `http://pub.deschutter.info/abs/99_11.html`

STATE SPACE MODELING IN MULTICHANNEL ACTIVE CONTROL SYSTEMS

G. Nijsse and M. Verhaegen
University of Twente, Faculty of Applied Physics
Measurement and System Technology Group
P.O. Box 217, 7500 AE ENSCHEDE, The Netherlands
E-mail: I.H.Appel-vanUlzen@TN.UTwente.NL

B. De Schutter and D. Westwick
Delft University of Technology
Faculty of Information Technology and Systems
Department of Electrical Engineering
P.O. Box 5031, 2600 GA DELFT, The Netherlands

N. Doelman
TNO Institute of Applied Physics
Department of Active Noise and Vibration Control
P.O. Box 155, 2600 AD DELFT, The Netherlands

Keywords: Active noise control, active vibration control, state space model, Filtered-X LMS, vibrating plate.

1 Introduction

In recent years, Active Control (AC) research and applications have gained much attention, e.g. Active Noise Control, Active Vibration Control and Active Acoustic Structural Control. Many universities and research institutes are actively participating in projects dealing with AC. However, despite the rapid increase in computational power and the development of fast and efficient algorithms, AC can still only be used for a limited number of applications such as exhaust pipes, air planes and air conditioners. Most of the time, even these applications are limited to narrow-band noise. Due to the complexity of the acoustics, large scale applications, such as actively canceling broad-band noise in a wide area, still pose serious computational problems when using hardware which is not too complex and expensive. For that reason we are looking for ways to reduce the computational burden in AC systems for reducing the price and the complexity of the hardware.

One of the methods we recently successfully investigated and which we deal with in this paper, is the use of state space models (SSMs) instead of finite impulse response models (FIRMs) in AC systems. While FIRMs are very commonly used in AC [7, 18], they can require a large number of coefficients, deteriorating the computational complexity, especially in poorly damped plants and/or broad-band noise control applications [7, 1]. On the other hand, one of the advantages of SSMs is

that they reduce the number of coefficients and the number of computations dramatically due to their "minimal" parameterization.

This paper is organized as follows. The next section gives a general description of SSMs, the derivation of SSMs and their benefits compared to FIRMs. The following section gives an explanation of SSMs applied to AC systems. More specifically, we will explain how SSMs are used in the Filtered-X LMS (FxLMS) algorithm and we will explain the reduction in the number of floating point operations per sample (flops — a flop is a multiplication or an addition [2]). A specific section is dedicated to the methods to obtain SSMs from impulse responses and measured Input/Output (I/O) data. These methods are illustrated for a vibrating plate model and an experimental vibrating plate structure. Finally, the conclusions and future research are given in the last section.

2 State space models

In this paper matrices are denoted by boldface uppercase letters ($\mathbf{X}$) and vectors are denoted by boldface lowercase letters ($\mathbf{x}$). Scalar signals will be denoted by normal lowercase letters (x). Linear convolution is denoted by the symbol "*" [8] and the Kronecker tensor product is denoted by the symbol "$\otimes$" [7].

2.1 What are state space models?

To introduce SSMs to the signal processing community and to make the relation between FIRMs and SSMs clear, we will give the basics of SSMs in this subsection (see e.g. [8, 9, 12]).

In state space form the relationship between the input and output signals is written as a system of first order difference or differential equations using a so called state vector $\mathbf{x}$, reflecting the state of the system [9]. A general form of a Multiple Input Multiple Output (MIMO) discrete time noiseless linear time invariant SSM with state dimension N , m inputs and p outputs is defined as:

$$\begin{aligned} \mathbf{x}(n+1) &= \mathbf{A}\mathbf{x}(n) + \mathbf{B}\mathbf{u}(n) \\ \mathbf{y}(n) &= \mathbf{C}\mathbf{x}(n) + \mathbf{D}\mathbf{u}(n) \end{aligned} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times N}$, $\mathbf{D} \in \mathbb{R}^{p\times m}$. $\mathbf{u}(n)$ is the input to the system, $\mathbf{x}(n)$ is the state of the system and $\mathbf{y}$ (n)is the output of the system. The transfer function of this system is equal to:

$$\mathbf{G}(z) = \mathbf{D} + \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$$

and can be determined with the z transform. Obviously, the structure of system (1) is totally different from the structure of FIRMs:

$$y(n) = g_0 + g_1u(n-1) + g_2u(n-2) + g_3u(n-3) + \cdots + g_ku(n-k)$$

with their accompanying transfer function:

$$G(z) = \sum_{n=0}^{k} g(n)z^{-n}$$

Note that due to the fact that in AC MIMO systems are modeled by a set of Single Input Single Output (SISO) FIRMs, we put the equations for the FIRMs in SISO and the equations for the SSM in MIMO form. This immediately reflects a benefit of SSMs: instead of having $p \times m$ FIRMs for a system

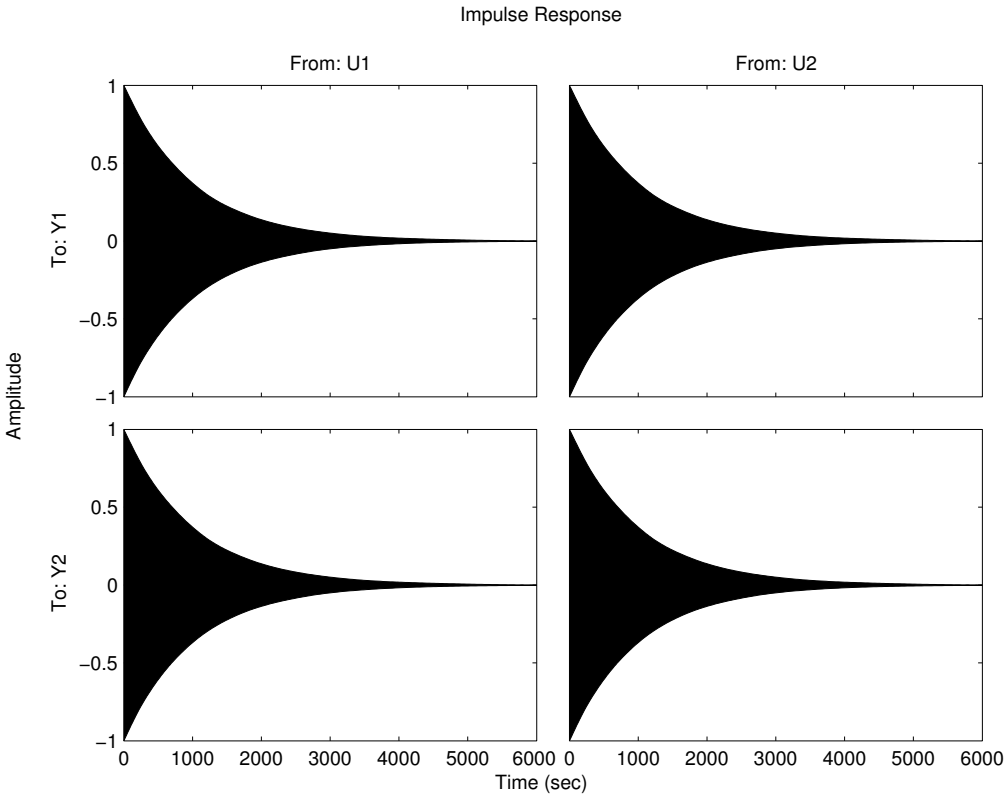

Figure 1: Impulse response for system (2). As can be seen from the picture, the system (2) should be modeled by 4 very long FIRMs.

with p inputs and m outputs, with SSMs it is possible to create one model *having a common state*. Furthermore, although SSMs absolutely differ from Infinite Impulse Response models, as far as the impulse response is concerned, a SSM can be compared with an Infinite Impulse Response model, which also has an infinite impulse response.

One of the great advantages of SSMs is that instead of having a large number of coefficients for FIRMs to reflect the I/O behavior, commonly SSMs only use a "minimal" number of coefficients and can reflect the I/O as well as FIRMs. As an example, consider the following SSM:

$$\mathbf{A} = \begin{bmatrix} -0.999 & 0 \\ 0 & -0.999 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, D = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \tag{2}$$

This SSM has 4 very long impulse responses (see Figure 1). It should be modeled by four very long FIRMs and is modeled here with a SSM with 12 coefficients only; the number of coefficients when fully parameterizing the system matrices **A**, **B**, **C**, **D** equals: (inputs $\times$ states) + (outputs $\times$ states) + (inputs $\times$ outputs). The reduction in the number of coefficients, as we will also see in the next section, gives rise to a dramatic reduction in the number of computations.

2.2 Obtaining state space models

Basically, SSMs can be obtained in two ways:

- From the estimated impulse response data [4, 6, 9]. Given the original FIRMs of the system, the corresponding SSM can be constructed easily. The key observation is that the relation between

the impulse response, the SSM, and the transfer functions of the systems, is given by:

$$\begin{aligned}\mathbf{G}(z) &= \sum_{k=0}^{\infty} \mathbf{G}(k) z^{-k} \\ &= \mathbf{D} + \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \\ &= \mathbf{D} + z^{-1}\mathbf{CB} + z^{-2}\mathbf{CAB} + z^{-3}\mathbf{CA}^2\mathbf{B} + \ldots\end{aligned}$$

Although the summation runs to ∞, only a finite portion of the impulse response is needed to obtain the SSM [14]. The Ho/Kalman/Kung algorithm [4, 6] makes use of this property and makes it possible to construct the **A**, **B**, **C**, **D** matrices of the SSM. Ho and Kalman [4] developed an algorithm to derive a minimal state space realization from non-perturbed data (no noise, no modeling errors etc.). Kung [6] extended this algorithm by using a singular value decomposition so that it can also be applied to perturbed data. Note that there is a direct relation between SSMs and FIRMs; finding a SSM from FIRMs is not an identification type of problem, but it represents the transformation of one model representation into another model representation.

- From the I/O data of the system [15, 17]. This approach is called the subspace model identification approach, which is a modern method for estimating SSMs from arbitrary, persistently exciting measured I/O data. The method has received wide attention in recent years and has proven to be very robust and easy to use in practice [3].

2.3 Benefits of state space models compared to FIR models

SSMs provide a degree of flexibility in model parameterization, which can enhance the performance enormously [13]. In contrast to polynomial or transfer function models, which requires polynomial fitting to get the final quantities, SSMs allow the use of numerically robust techniques such as eigenvalue decomposition and well-conditioned matrices to obtain the desired quantities [13]. Another advantage is that if the model order is not known a priori and is to be estimated from data, the state space approach provides estimates of the model order [13]. Also, SSMs provide a "minimal" parametrization which has the potential to be made robust to finite precision errors; this is completed by the availability of numerically reliable methods for estimating the parameters, such as the singular value decomposition [13]. This is not only advantageous for the reduction in the number of coefficients and the number of computations, but also the calculations are performed numerically more exact.

3 State space models applied to active control systems

In this paper we deal with an AC system that has J ($j = 1, 2, \ldots, J$) reference signals, K ($k = 1, 2, \ldots, K$) secondary sources, M ($m = 1, 2, \ldots, M$) error microphones and as a result $K \times J$ adaptive controller filters of length L, which may be represented by FIRMs.

3.1 Where can state space models be applied?

In almost all AC algorithms, estimates of paths are needed e.g. [7, 18]:

- the FxLMS algorithm requires an estimate of the secondary path.
- the Filtered-ε algorithm requires an estimate of the inverse secondary path.

- the Filtered-U algorithm requires a number of estimates, among which an estimate of the secondary path and an estimate of the feedback path.
- the Filtered-E algorithm requires a shaping function and an estimate of the secondary path.

Commonly, in all these algorithms, the shaping functions and estimates are modeled by FIRMs, e.g. in the Filtered-ε algorithm, FIRMs are used to model the inverse of the secondary path. Note that this is most of the time not even an exact inverse [18]. On the basis of the disadvantages of FIRMs as mentioned in the previous section and since FIRMs require a large number of computations, especially in poorly damped plant and/or broad-band noise control applications, we propose to use SSMs in these algorithms instead of FIRMs. This will not only give more numerical robustness, but it reduces the parameters and computations and it paves the way towards robust control design methods and inversion techniques, giving rise to new possibilities in acoustics e.g. in inversion (Filtered-ε LMS algorithm) [10, 11]. In the next subsection we highlight the application of SSMs to the FxLMS algorithm.

3.2 State space models in the Filtered-X LMS algorithm

Consider the Multiple Reference / Multiple Output Filtered-X LMS (MIMO FxLMS) algorithm of which a block diagram is given in Figure 2 [18, 7]. The filter update equation is given by:

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) + \mu \mathbf{X}'(n)\mathbf{e}(n) \\ &= \mathbf{w}(n) + \mu \Big(\mathbf{S}^T(n) \otimes \mathbf{x}(n)\Big)\mathbf{e}(n)\end{aligned} \tag{3}$$

where $\mathbf{x}(n) \in \mathbb{R}^{JL\times 1}$ represents the vector with reference signals, μ is the step size, $\mathbf{S}(n)$ is of size $M \times K$ and represents the secondary path impulse response matrix, $\mathbf{e}(n) \in \mathbb{R}^{M\times 1}$ denotes the vector with error signals and $\mathbf{w}(n) \in \mathbb{R}^{JKL\times 1}$ is a vector containing *all* filter weights. For details about the algorithm see [7]. Note that we do assume that the secondary path is linear time invariant, i.e. the impulse response matrix at time n is equal to the impulse response matrix at time $n-1$: $\mathbf{S}(n) = \mathbf{S}(n-1)$.

It is easy to derive the update equations per individual filter $\mathbf{w}_{kj}(n)$ [7, 5]:

$$\mathbf{w}_{kj}(n+1) = \mathbf{w}_{kj}(n) + \mu \sum_{m=1}^{M} [s_{mk}(n) * \mathbf{x}_j(n)] e_m(n)$$

where $s_{mk}(n)$ is the FIRM representing the secondary path from the kth secondary source to the mth error microphone. This update equation can also be written as:

$$\begin{bmatrix} w_{kj}^1(n+1) \\ w_{kj}^2(n+1) \\ \vdots \\ w_{kj}^L(n+1) \end{bmatrix} = \begin{bmatrix} w_{kj}^1(n) \\ w_{kj}^2(n) \\ \vdots \\ w_{kj}^L(n) \end{bmatrix} + \begin{bmatrix} \mu \sum_{m=1}^{M} [s_{mk}(n) * x_j(n)] e_m(n) \\ \mu \sum_{m=1}^{M} [s_{mk}(n) * x_j(n-1)] e_m(n) \\ \vdots \\ \mu \sum_{m=1}^{M} [s_{mk}(n) * x_j(n-L+1)] e_m(n) \end{bmatrix} \tag{4}$$

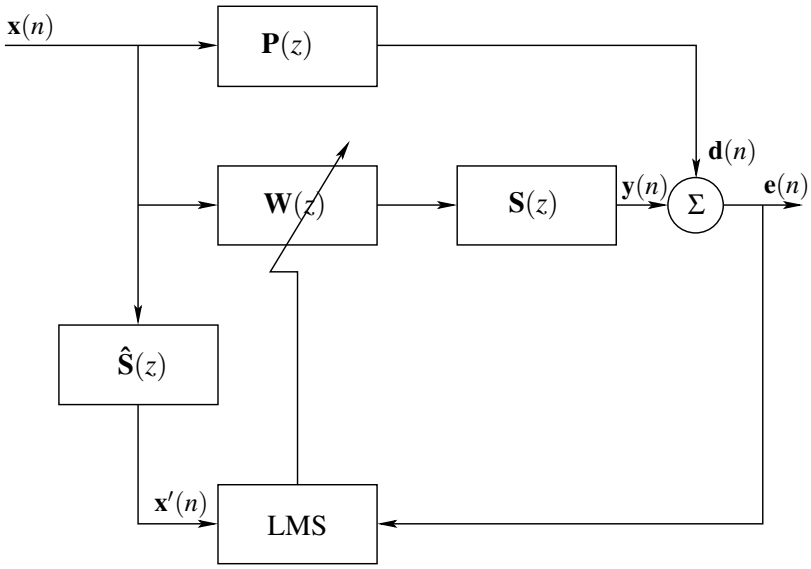

Figure 2: A block diagram of the MIMO FxLMS algorithm. The generated unwanted sound $\mathbf{x}(n)$ is filtered by an estimate of the secondary path before it is used by the LMS algorithm. The goal of this scheme is to cancel the sound $\mathbf{d}(n)$ by the signal $\mathbf{y}(n)$, which is driven by an adaptive filter $\mathbf{W}(z)$. $\mathbf{P}(z)$ is the primary path, $\mathbf{S}(z)$ is the secondary path and $\hat{\mathbf{S}}(z)$ is an estimate of the secondary path.

Let us now focus on the computationally complex part of (4), i.e. the summation and the convolution. More specifically, let us focus on the top element of the second part of (4), which corresponds to $w_{kj}^1(n+1)$ and which will be denoted by $p_{kj}(n)$:

$$p_{kj}(n) = \sum_{m=1}^{M} [s_{mk}(n) * x_j(n)] e_m(n)$$

This can be written as:

$$p_{kj}(n) = [s_{1k}(n) * x_j(n)] e_1(n) + [s_{2k}(n) * x_j(n)] e_2(n) + \cdots + [s_{Mk}(n) * x_j(n)] e_M(n)$$

or:

$$p_{kj}(n) = \underbrace{\begin{bmatrix} e_1(n) & e_2(n) & \dots & e_M(n) \end{bmatrix}}_{\text{Error: } \mathbf{e}^T(n)} \underbrace{\begin{bmatrix} s_{1k}(n) * x_j(n) \\ s_{2k}(n) * x_j(n) \\ \vdots \\ s_{Mk}(n) * x_j(n) \end{bmatrix}}_{\text{Convolutions: } \mathbf{y}_{kj}(n)} \tag{5}$$

Alternatively, the $\mathbf{y}_{kj}(n)$ vector which is generated by the M convolutions in (5), can be written as the output of a Single Input Multiple Output (SIMO) SSM:

$$\begin{aligned} \mathbf{z}(n+1) &= \mathbf{A}\mathbf{z}(n) + \mathbf{b}x_j(n) \\ \mathbf{y}_{kj}(n) &= \mathbf{C}\mathbf{z}(n) \end{aligned} \tag{6}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{b} \in \mathbb{R}^{N\times 1}$ and $\mathbf{C} \in \mathbb{R}^{M\times N}$, with N the state dimension. Since we have J inputs and K secondary sources, this means that for computing (5) for all filters, in total:

$$K \times J$$

SSMs have to be calculated. So, instead of using the conventional FIRMs with its convolutions, we now use SSMs to perform the filtering with the secondary path. As will be seen in the next subsection, this reduces the number of computations significantly.

3.3 Computational complexity: number of floating point operations per second

For calculating (4) using fully parameterized SSMs, the following number of flops have to be performed:

Operation	No. of Multiplications	No. of Additions
$\mathbf{A}\mathbf{z}(n)$	N^2	$N(N-1)$
$\mathbf{B}x_j(n)$	N	–
$\mathbf{A}\mathbf{z}(n)+\mathbf{B}x_j(n)$	–	N
$\mathbf{C}\mathbf{z}(n)$	NM	$M(N-1)$
$\mathbf{e}^T(n)\mathbf{y}_{kj}(n-i)\ \ (i:0\rightarrow L-1)$	LM	$L(M-1)$
Total	$N^2+(1+M)N+LM$	$N^2+M(N+L)-(M+L)$

Note that all convolutions $[s_{mk}(n) * x_j(n-j)]$, $j = 1,\ldots,L-1$ in (4) are shifted versions of the top convolution $[s_{mk}(n) * x_j(n)]$ at previous sample instants, and do not have to be calculated at sample instant n. Only the multiplication with $e_m(n)$ has to be calculated.

This means that for calculating all $\mathbf{w}_{kj}$ filters:

$$KJ\big\{2N^2+(1+2M)N+2LM-(M+L)\big\}\ \text{flops}$$

are needed. So, for calculating the complete MIMO FxLMS update equation (3) using SSMs:

$$2JKL+KJ\big\{2N^2+(1+2M)N+2LM-(M+L)\big\}\ \text{flops} \tag{7}$$

have to be performed. Remark: more compact (condensed) SSMs could be used to further reduce the computational complexity [16].

On the other hand, if the calculations are straightforwardly performed with convolutions and summations, i.e. with the FIRMs, then for every filter $\mathbf{w}_{kj}$, M convolutions of length F (= length of the FIR filter to represent the secondary path) have to be carried out. This results in MF multiplications and $M(F-1)$ additions. Every convolution outcome has to be multiplied by the error signal and the result has to be summed, which results in M extra multiplications and $M-1$ additions. For the complete summation (4) this gives $MF+LM$ multiplications and $M(F-1)+L(M-1)$ additions, in total $2MF+2LM-M-L$ flops. This gives for all $\mathbf{w}_{kj}$ filters:

$$KJ\big\{2MF+2LM-(M+L)\big\}\ \text{flops}$$

So, for calculating the complete MIMO FxLMS update equation (3) using FIRMs:

$$2JKL+KJ\big\{2MF+2LM-(M+L)\big\}\ \text{flops} \tag{8}$$

are needed. See also Figure 3 for results in the reduction of the number of flops with SSMs compared to FIRMs. From this picture can be concluded that the benefits of using SSMs are most pronounced if the secondary path originally is modeled by large order FIRMs and if the controller does not use many coefficients (L is small).

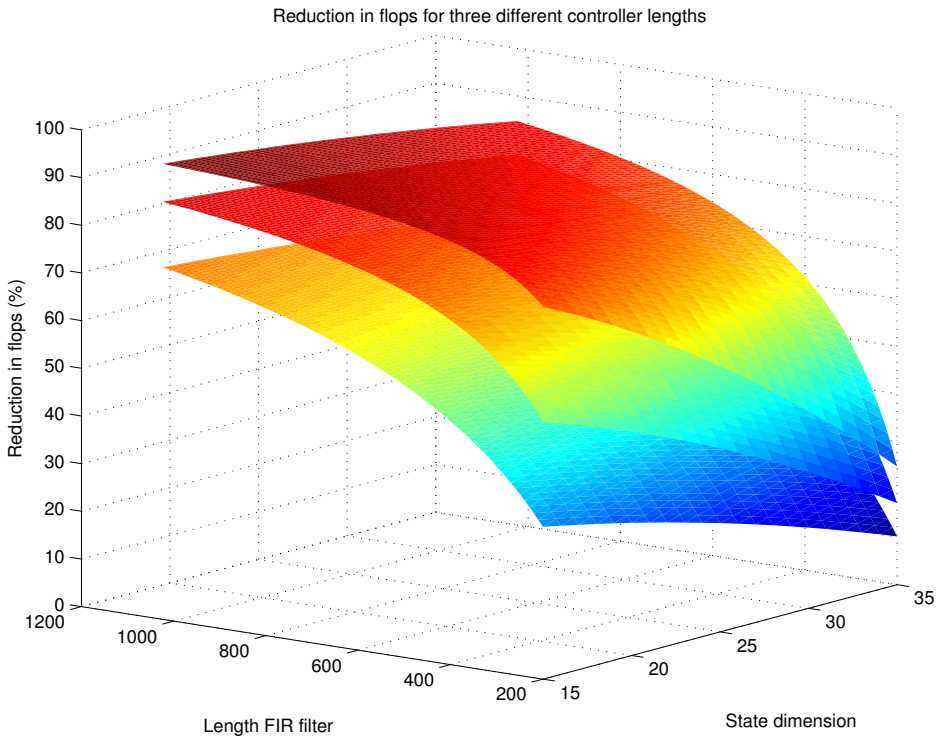

Figure 3: Reduction in the number of flops with SSMs, compared to FIRMs. The upper surface gives the results for a controller length $L = 10$. The middle surface gives results for controller length $L = 100$ and the lower surface gives the results for a controller length $L = 300$. As can be seen the benefits of using SSMs are most pronounced (+95 %) for a low order controller ($L = 10$) and in case the secondary path is originally modeled by large FIRMs but can be estimated by low order SSMs.

4 Experimental results

4.1 Vibrating plate model

A model of a vibrating plate in free space was chosen for testing the state space approach. A sound wave, incident from the left of the structure in Figure 4, is captured by the reference microphone, and applied to an adaptive filter, which drives the 6 piezo-electric actuators which create a region of silence on the right of the plate. The error signal is captured by 6 error microphones. The system is sampled at 1 kHz.

The goal of the simulation is to construct the $6 \times 6 = 36$ secondary path estimates, indicated in the figure by S_{ij}, by 6 SIMO SSMs (as explained in (6)) using the HKK method. Originally, the secondary path estimates were modeled by their impulse responses, leading to 36 FIRMs. Every FIRM has $F = 512$ elements, meaning that $36 \times 512 = 18432$ coefficients were needed to model the estimate of the secondary path transfer matrix $\mathbf{S}(z)$ (see Figure 3).

By using the Ho/Kalman/Kung algorithm we estimated the 6 SIMO SSMs. Figure 5 gives one of the impulse responses and its estimate with the residual. As can be seen, the SSM performs very well. For a further comparison of the estimated and real impulse responses, see Table 1 in which the Variance Accounted For (VAF) of the 36 signals is given together with the orders of the models. The

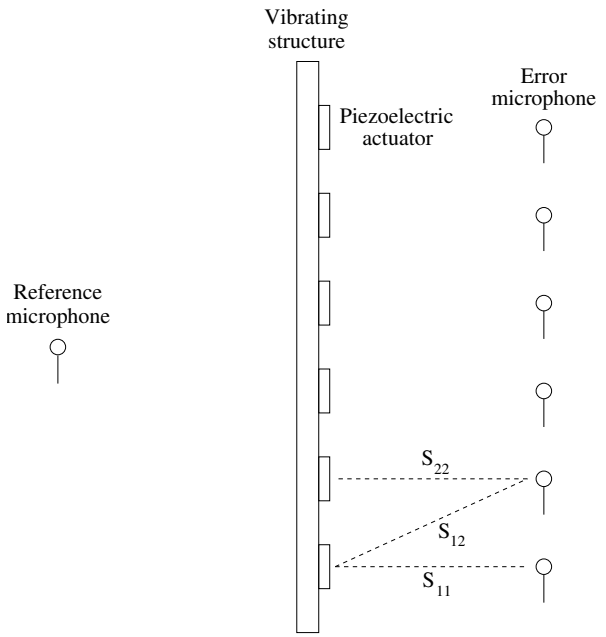

Figure 4: Vibrating structure with 6 piezoelectric actuators and 6 error microphones and 1 reference microphone. S_{ij} is the secondary path transfer function from the ith piezoelectric actuator to the jth error microphone.

VAF is defined as:

$$\text{VAF} = \left(1 - \frac{\text{Variance}(y - \hat{y})}{\text{Variance}(y)}\right) \times 100\,\%$$

where y and $\hat{y}$ are respectively the target signal and the measured output. If the VAF is 100 % the signals are identical. By using these models in the FxLMS algorithm (recall that $J = 1$, $K = M = 6$, $F = 512$), according to (7) and (8), a reduction in the number of flops of 58.4 % is accomplished if we assume a broad-band noise controller ($L = 256$), and 83.2 % if we assume a narrow-band noise controller ($L = 8$). Finally, by using SSMs, we only used 810 coefficients in total, which is a reduction in the number of *coefficients* with 95.6 %, compared to FIRMs.

4.2 Experimental vibrating plate

To test the state space approach even more, in this subsection the subspace modeling of an experimental vibrating plate structure in a reverberation room is described, as set up by the TNO Institute of Applied Physics, Delft, The Netherlands. The structure of the physical vibrating plate is the same as the structure of the model of the previous subsection. However, instead of the model having 6 actuators and 6 error microphones, this experimental vibrating plate has 7 actuators and 7 error microphones. Note that since we are dealing with a reverberation chamber, the response is more complicated.

The goal of this experiment is to find 7 SIMO models of the secondary path from the 7 actuators (inputs) to the 7 error microphones (outputs) on the basis of raw I/O data of length $7 \times 7 \times 200000$. Originally, the experimental vibrating plate setting was used in a FxLMS scheme, sampled at 2 kHz, with FIRMs of 160 coefficients.

Before we started the identification, first the data was preprocessed, i.e. the data was scaled, detrended and down-sampled from 10 kHz to 4 kHz. Note that we do not down-sample to 2 kHz, since

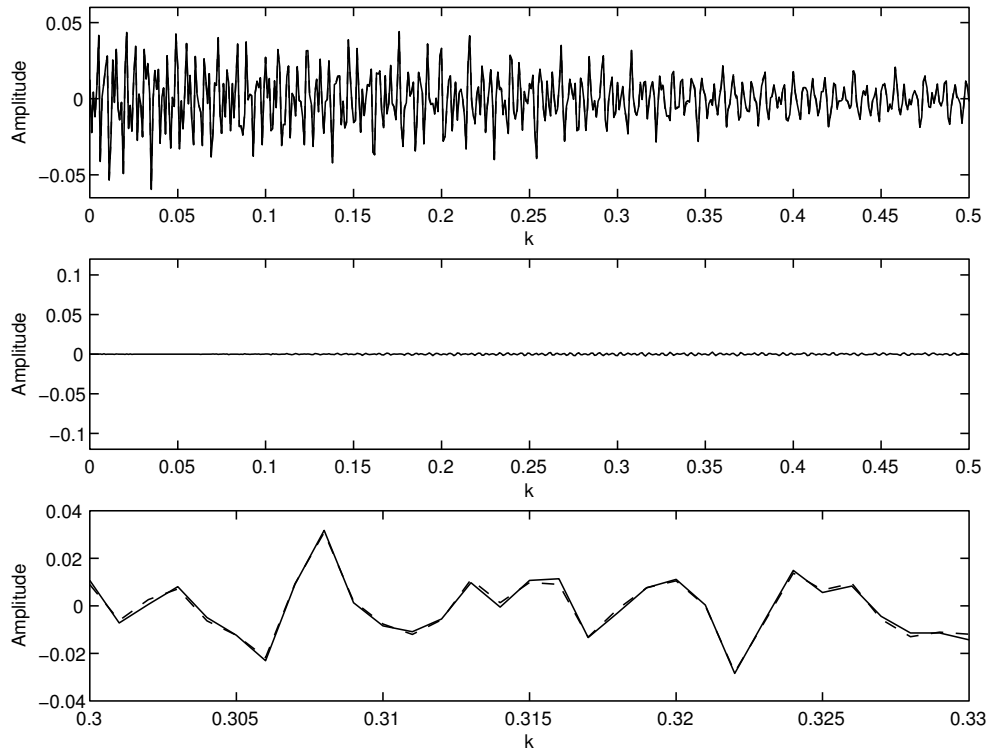

Figure 5: Upper plot: the real (solid) and estimated (dashed) impulse response S_{63}. Middle plot: the residual. Lower plot: zoomed in version of the real (solid) and estimated (dashed) impulse response.

Table 1: VAF of the 36 estimated and real impulse responses. The entry (i, j) corresponds to S_{ij}; the boldface typed entry corresponds to S_{63} and Figure 5. Note that we deal here with 6 models with each 6 outputs.

$i \setminus j$	1	2	3	4	5	6	Order N
1	99.88	99.87	99.63	99.74	99.86	99.93	21
2	99.87	99.88	99.74	99.63	99.93	99.86	21
3	99.45	99.74	99.41	99.71	99.47	99.73	14
4	99.74	99.47	99.71	99.41	99.73	99.47	14
5	99.86	99.93	99.63	99.73	99.88	99.87	21
6	99.93	99.86	***99.73***	99.63	99.87	99.88	21

this would sincerely penalize the identification procedure. As it appeared from the output spectra of the signals, most of the information was contained in the 1 kHz – 3 kHz region, so down-sampling to 2 kHz would discard a lot of valuable information. From this dataset we selected $7 \times 7 \times 10000$ data points for training purposes on which the models were estimated and $7 \times 7 \times 5000$ for cross-validation purposes. Finally, the delays were removed from the data, by estimating FIRMs on the data.

We performed several experiments on the data. These experiments led to the conclusion that we are dealing with a highly (block) diagonal system. Only two or three outputs from the seven outputs could be modeled for a specific input (we call these the dominant outputs). The other outputs could not be modeled very well, meaning that the specific input has minor influence on that specific output.

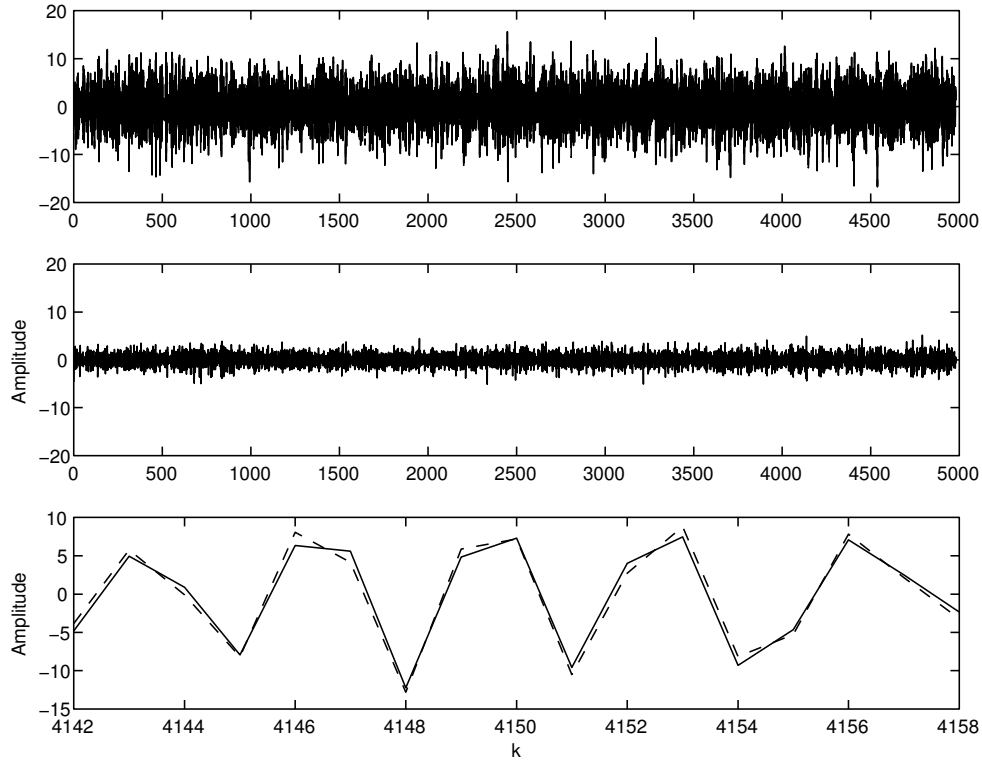

Figure 6: Upper: the real validation data (solid) and the output of the model (dashed) from input 3 to output 3. Middle plot: the residual. Lower plot: zoomed in version of the real validation data (solid) and the output of the model (dashed).

Or, non-dominant outputs can be regarded to be noise as far as modeling is concerned. The same conclusion was also drawn from correlation analysis methods.

We validated both the SSMs and the FIRMs on the validation I/O data. However, since we used a sampling frequency of 4 kHz we estimated a new set of FIRMs based on the data which was sampled at 4 kHz and used 320 coefficients instead of 160. See Figure 6 for a comparison between the validation data and the model output from actuator 3 to output 3 by our 3rd SIMO model. For a further comparison see Table 2 which gives the VAFs for all 49 signals. This table also clearly shows the dominant behavior of the plate: only two or three outputs could be modeled well. Note that although in some cases the FIRMs perform better than SSMs on the validation data, there is a trade-off between the number of parameters and the accuracy to fit the I/O data. The ultimate performance test would be to put these SSMs in the FxLMS algorithm and test the error measured at the error microphones during operation.

By using these models in the FxLMS algorithm (recall that $J = 1$, $K = M = 7$, $F = 320$), according to (7) and (8), a reduction in the number of flops of 53.1 % is accomplished if we assume a broadband noise controller ($L = 256$), and 86.9 % if we assume a narrow-band noise controller ($L = 8$). Finally, by using SSMs, we only used 577 coefficients in total, which is a reduction in the number of *coefficients* with 96.3 %, compared to FIRMs (15680 coefficients).

Table 2: VAF for the 49 I/O signals of the experimental vibrating plate, estimated by SSMs/FIRMs. Also the state space model order is given in this table. Note that we deal here with 7 models with each 7 outputs.

$i \setminus j$	1	2	3	4	5	6	7	N
1	89.6/94.0	13.3/43.0	64.5/76.0	0/39.6	25.5/45.4	1.8/33.1	6.9/32.6	3
2	11.8/37.2	72.1/93.5	12.9/37.4	66.8/79.9	4.2/32.7	25.8/45.0	0/29.5	25
3	61.2/73.4	2.27/39.2	91.3/94.4	18.2/42.7	71.6/80.8	7.8/31.5	24.2/51.5	4
4	12.8/36.7	64.1/76.9	12.4/39.9	85.8/94.9	9.6/35.9	71.6/80.4	8.8/33.4	24
5	21.1/41.7	4.1/32.0	63.0/72.0	3.7/38.7	90.6/94.0	6.3/42.7	72.0/80.4	3
6	12.6/38.1	11.2/35.8	27.8/51.0	5.9/39.0	69.0/79.7	8.5/37.1	87.8/93.4	5
7	3.7/32.2	5.6/38.3	0.6/32.1	18.4/50.0	2.7/34.0	73.0/84.3	5.7/43.9	2

5 Conclusions

5.1 Conclusions

We have investigated the application of state space models to Active Control applications. More specifically, we investigated the well-known Filtered-X LMS algorithm. It appeared that it is far more attractive to use state space models instead of finite impulse response models to model the secondary path. Finite impulse response models can require a large number of coefficients and for that reason a large number of computations, especially in poorly damped plants and/or broad-band noise control applications. State space models on the other hand, provide a "minimal" parameterization, which is beneficial for implementing the algorithm on DSPs. Furthermore, state space models provide more numerical robustness, leading to more exact computations, particularly in fixed precision arithmetic. For moderate problem sizes, calculations show coefficient reductions up to 96 % and computational reductions up to 86 %. Obtaining state space models can be done in two ways: from the corresponding impulse responses or with subspace identification techniques from input/output data (where the latter is more accurate).

5.2 Future research

The impact of state space models in the Filtered-X LMS algorithm has to be tested in a simulation environment, compared to FIR models. This way the number of flops can determined on a solid basis and the effect of modeling errors can be determined. Further, the use of state space models gives rise to a new area of research on inversion in Active Noise Control. As is well-known, with an inverse model of the secondary path it is possible to filter out its impact in e.g. in the Filtered-ε LMS algorithm. Also the number of computations can be reduced by using an inverse model. The state space inverse requires much less parameters and is more precise than the commonly used FIR inverses. This enables more computations per sampling instant. Note that currently we are working out the idea of the application of state space models to the Filtered-ε algorithm: instead of using an inverse based on FIR models, we use a state space model inverse. Finally, with the use of state space models a whole variety of control algorithms become available, such as robust control algorithms (e.g. H_∞ control).

Acknowledgement

This research was supported by the TNO Institute of Applied Physics, Delft, The Netherlands.

References

[1] S.C. Douglas. Fast implementations of the Filtered-X LMS and LMS algorithms for multichannel active noise control. Accepted for publication in *IEEE Transactions on Speech and Audio Processing*.

[2] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The John Hopkins University Press, 1996.

[3] B.R.J. Haverkamp, C.T. Chou, and M. Verhaegen. SMI toolbox: A matlab toolbox for state space model identification. *Journal A*, 38(3):34–37, September 1997.

[4] B.L. Ho and R.E. Kalman. Effective construction of linear state-variable models from input-output functions. In *Regelungstechnik*, volume 14, pages 545–548, 1966.

[5] X. Kong, P. Liu, and S.M. Kuo. Multiple channel hybrid active noise control systems. *IEEE Transactions on Control Systems Technology*, 6(6):719–729, November 1998.

[6] S. Kung. A new identification and model reduction algorithm via singular value decompositions. In *Proceedings 12th Asilomar Conference on Circuits, Systems and Computers*, pages 705–714, 1978.

[7] S.M. Kuo and D.R. Morgan. *Active Noise Control Systems*. John Wiley & Sons, Inc., 1996.

[8] H. Kwakernaak and R. Sivan. *Modern Signals and Systems*. Prentice-Hall, Inc., 1991.

[9] L. Ljung. *System Identification*. Prentice Hall, Inc., 1999.

[10] M. Miyoshi and Y. Kaneda. Inverse filtering of room acoustics. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(2):145–152, February 1988.

[11] P.A. Nelson, F. Orduña-Bustamante, and H. Hamada. Inverse filter design and equalization zones in multichannel sound reproduction. *IEEE Transactions on Speech and Audio Processing*, 3(3):185– 192, May 1995.

[12] C.L. Phillips and H.T. Nagle. *Digital Control System Analysis and Design*. Prentice Hall, Inc., 1995.

[13] B.D. Rao and K.S. Arun. Model based processing of signals: a state space approach. *Proceedings of the IEEE*, 80(2):283–309, 1992.

[14] A.J. Tether. Construction of minimal state-variable models from finite input-output data. *IEEE Transactions on Automatic Control*, 1970.

[15] P. Van Overschee and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic and stochastic systems. *Automatica*, 30(1):75–93, 1994.

[16] M. Verhaegen. *A new class of algorithms in linear system theory*. PhD thesis, Katholieke Universiteit Leuven, Departement Electrotechniek, November 1985.

[17] M. Verhaegen. Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica*, 30(1):61–74, 1994.

[18] B. Widrow and E. Walach. *Adaptive Inverse Control*. Prentice Hall, Inc., 1996.