Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SISTA/TR 05-14

Instrumental variable methods for acoustic feedback cancellation[1]

Toon van Waterschoot[2] [3], Koen Eneman and Marc Moonen[2]

November 2004

[1]This report is available by anonymous ftp from *ftp.esat.kuleuven.ac.be* in the directory *pub/sista/vanwaterschoot/reports/05-14.pdf*

[2]K.U.Leuven, Dept. of Electrical Engineering (ESAT), Research group SCD(SISTA), Kasteelpark Arenberg 10, 3001 Leuven, Belgium, Tel. +32 16 321927, Fax +32 16 321970, WWW: *http://www.esat.kuleuven.ac.be/sista-cosic-docarch.* E-mail: *toon.vanwaterschoot@esat.kuleuven.ac.be*.

[3]Toon van Waterschoot is a Research Assistant with the I.W.T. (Flemish Institute for Scientific and Technological Research in Industry). This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven, in the frame of the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office IUAP P5/22 ('Dynamical Systems and Control: Computation, Identification and Modelling'), the Concerted Research Action GOA-MEFISTO-666 (Mathematical Engineering for Information and Communication Systems Technology) of the Flemish Government and IWT project 020476: 'SMS4PA: Sound Management System for Public Address Systems'. The scientific responsibility is assumed by its authors.

Instrumental variable methods for acoustic feedback cancellation

Toon van Waterschoot, Koen Eneman and Marc Moonen

November 2004

1 Introduction: Adaptive feedback cancellation

1.1 Acoustic feedback problem

Consider the 1–microphone/1–loudspeaker system of figure 1. A desired near–end signal v is captured by a microphone, amplified (with a factor G_A), delayed (Δ time samples) and inputted into a loudspeaker. The main objective of the scheme is to produce a loudspeaker signal

$$u(k) = G_A v(k - \Delta) \tag{1}$$

that is a delayed and amplified copy of v. Possible applications of this setup are on the one hand hearing aids and on the other hand public address systems (live music performances, large auditoria, airports, ...).

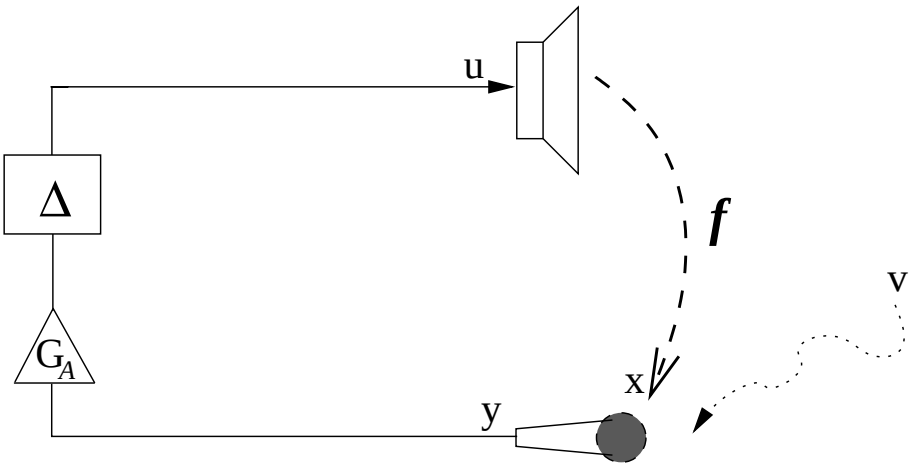

Figure 1: Acoustic feedback problem

Due to the inherent acoustic coupling f between loudspeaker and microphone, however, a closed–loop system is created, i.e. the loudspeaker signal is fed back to the microphone and in this way the desired signal is inputted again into the circuit. This closed–loop system may eventually become unstable and give rise to a harmful tonal (sinusoidal) feedback signal. Even when instability can be avoided, in the case of a small amplification for instance or if appropriate anti–feedback software is applied, the desired signal will nevertheless be distorted, i.e. $|u(k) - G_A v(k - \Delta)| \neq 0$, and a bad signal quality will result.
Several algorithmic approaches have been proposed to suppress acoustic feedback, an overview of which has been given in [16]. It was found that one of the more advanced and most promising classes of feedback cancellation techniques relies on adaptive filtering.

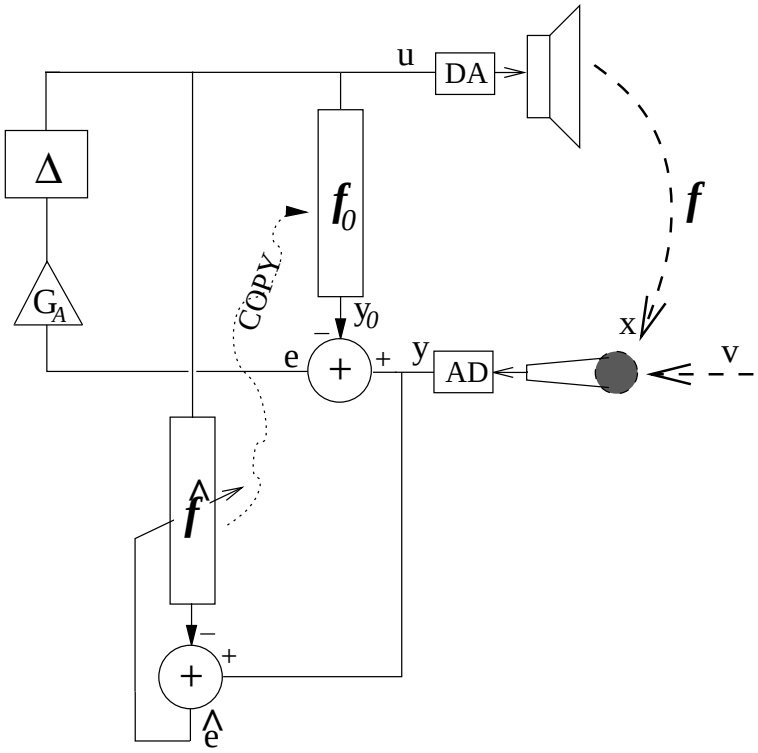

Figure 2: Adaptive filtering based feedback cancellation scheme

1.2 Standard adaptive feedback cancellation

In an adaptive feedback cancellation setup a digital filter $\hat{f}$ is added to the algorithmic scheme and is updated based on the error signal $\hat{e}$ so that it mimics the acoustic transfer function f (see figure 2). From time to time the acoustic model $\hat{f}$ is copied to the cancellation filter f_0 in order to effectively cancel out the feedback signal. In this respect, we basically considered two (extreme) modes of operation in our research :

- an off–line or batch mode, where we assess the performance of adaptive feedback cancellation algorithms in their ability to model the acoustic transfer function f and to reduce the estimated quality measure $|\hat{e}(k) - v(k)|$ in a (stable) closed–loop scheme with $f_0 = 0$.
- an on–line mode, where we assess the performance of adaptive feedback cancellation algorithms in their ability to model the acoustic transfer function f and to reduce the quality measure $|e(k) - v(k)|$ in a (stable) closed–loop scheme where $\hat{f}$ is copied to f_0 each sample instance.

In intermediate schemes the modelling filter $\hat{f}$ is copied to its cancellation equivalent f_0 every p samples, where $1/p$ is the update frequency.
Remark that the off–line mode corresponds to a standard adaptive identification process, as is used in acoustic echo cancellation for instance, however in this case operating in closed loop. As a consequence, the inputs to the adaptive filter, u and y, are highly correlated and a considerable bias will result (see also Eq. 26). This will lead to an erroneous acoustic model $\hat{f} \neq f$ and a quality measure $|\hat{e}(k) - v(k)|$ that is different from 0.
In the on–line mode another problem arises on top of that. The adaptive filter $\hat{f}$ has to cope with a highly time–varying acoustic environment now, even when the feedback path f is stationary, as the updated modelling filter $\hat{f}$ is continuously copied to f_0 and inserted in the circuit. In this

way the scheme acts as a complex system with complicated internal interconnections and feedback showing a nonlinear behaviour. However, thanks to the fact that the effects of a changing acoustic model $\hat{f}$ are immediately noticeable in the circuit, the on–line adaptive scheme shows, in general, a stable and good convergence behaviour.

1.3 Transfer functions and performance indices

The equations describing the standard adaptive feedback cancellation scheme of figure 2 can be formulated in the frequency (or z–) domain. It is observed that[1]

$$\begin{aligned} Y(f) &= V(f) + X(f) && (2)\\ E(f) &= Y(f) - Y_0(f) && (3)\\ X(f) &= F(f)U(f) && (4)\\ Y_0(f) &= F_0(f)U(f) && (5)\\ U(f) &= G(f)E(f) && (6) \end{aligned}$$

where $G(f)$ takes into account the amplifier gain G_A and the time delay Δ. If these equations are combined, the transfer function from v to, respectively, u, y and e can be computed :

$$\frac{U}{V} = \frac{G}{1-(F-F_0)G} \quad (7)$$

$$\frac{Y}{V} = \frac{1+F_0G}{1-(F-F_0)G} \quad (8)$$

$$\frac{E}{V} = \frac{1}{1-(F-F_0)G}. \quad (9)$$

Recall that the main objective of the scheme is to produce a loudspeaker signal $u(k) = G_A v(k-\Delta)$. Remark from Eq. 9 that this goal is reached if there is no bias, i.e. if $\hat{f} = f_0$, as then the quality measure $|\hat{e}(k) - v(k)| = 0$. Observe also that if the amplifier gain G_A is increased

$$\lim_{G_A \to \infty} \frac{U}{V} = \frac{1}{-(F-F_0)}, \quad (10)$$

which, in general, leads to an unstable system, as $F - F_0$ is typically non–minimum phase. Higher amplifier gains typically lead to more distortion and near instability. On the other hand, the higher the gain, the better the adaptation.

In order to assess the performance of the algorithms two performance measures have been considered :

1. A first performance index is the normalised bias between the real acoustic transfer function and the adaptive model. It is defined as

$$\boxed{\delta_1 = \frac{||\hat{\mathbf{f}} - \mathbf{f}||}{||\mathbf{f}||},} \quad (11)$$

   where $\mathbf{f}$ is an N–th order FIR filter representing the feedback path f :

$$\mathbf{f} = \begin{bmatrix} f(0) & \dots & f(N) \end{bmatrix}^T. \quad (12)$$

   Likewise, $\hat{\mathbf{f}}$ is defined as

$$\hat{\mathbf{f}} = \begin{bmatrix} \hat{f}(0) & \dots & \hat{f}(N) \end{bmatrix}^T, \quad (13)$$

[1] We define $S(f)$ as the frequency–domain equivalent of the time–domain signal $s(k)$.

representing the adaptive filter. If $\mathbf{f}$ and $\hat{\mathbf{f}}$ have unequal length, the shortest filter has to be padded with zeros. Observe that if $\delta_1 = 0$, the modelling filter $\hat{f}$ matches the feedback path f. As a consequence, the loudspeaker signal is an exact, delayed and amplified copy of the signal–of–interest v, as can be observed from Eq. 7. Hence, Eq. 1 is fulfilled. Remark however that a small δ_1 does not automatically imply a small error $u(k) - G_A e(k - \Delta)$.

2. Therefore, a second performance index is defined as

$$\boxed{\delta_2 = |e(k) - v(k)|,} \tag{14}$$

which really measures the quality of the loudspeaker signal as a function of time. Indeed, signal e goes to v if δ_2 approaches zero and as $u(k)$ equals $G_A e(k - \Delta)$, Eq. 1 is fulfilled. Hence, δ_2 verifies to which extent u approaches $G_A v(k - \Delta)$. In an off–line experiment ($f_0 = 0$) we define $\delta_2 = |\hat{e}(k) - v(k)|$ assuming that if the modelling filter $\hat{f}$ is copied to f_0, the error e goes to $\hat{e}$.
Remark that Eq. 14 can be expressed in the frequency domain as

$$\delta_2(f) = |E - V|. \tag{15}$$

Taking into account Eq. 9, equation 15 can be rewritten as

$$\delta_2(f) = \left| \frac{(F - F_0)G}{1 - (F - F_0)G} V \right|. \tag{16}$$

It is observed that in general minimising

$$\min_{F_0} \left| \frac{(F - F_0)G}{1 - (F - F_0)G} V \right| \tag{17}$$

is far from trivial, due to the structure of the object function, an its dependence on environmental parameters like F, G_A and Δ. Furthermore, $\delta_2(f)$ depends on the near–end signal v. Hence, a minimisation of Eq. 16 is not straightforward.

Both performance measures have been computed to evaluate our experiments (see section 4). In order to clearly observe the evolution of δ_2 as a function of time, $|e(k) - v(k)|$ was averaged by a lowpass filter with a window length of typically 500 samples. Time–averaging on δ_1 was, in general, not necessary. It was found that the observations and conclusions drawn from time evolution plots of δ_1 and δ_2 were typically consistent with each other. Hence, both performance indices seem reliable. They are furthermore easy to compute in a simulation environment.

1.4 Standard adaptive filtering approach : bias

In a standard adaptive filtering approach the following cost is minimised :

$$\sum_{i=0}^{L} \left(y(k - i) - \sum_{n=0}^{N} u(k - i - n)\hat{f}(n) \right)^2. \tag{18}$$

Finding the solution to Eq. 18 is equivalent to solving

$$\mathbf{U}\hat{\mathbf{f}} \overset{LS}{=} \mathbf{y} \tag{19}$$

in least–squares sense, where

$$\mathbf{y} = \mathbf{U}\mathbf{f} + \mathbf{v}, \tag{20}$$

$$\mathbf{y} = \begin{bmatrix} y(k - L) \\ \vdots \\ y(k) \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v(k - L) \\ \vdots \\ v(k) \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f(0) \\ \vdots \\ f(N) \end{bmatrix}, \quad \hat{\mathbf{f}} = \begin{bmatrix} \hat{f}(0) \\ \vdots \\ \hat{f}(N) \end{bmatrix} \tag{21}$$

and

$$\mathbf{U} = \begin{bmatrix} u(k-L) & u(k-L-1) & \dots & u(k-L-N) \\ u(k-L+1) & u(k-L) & \dots & u(k-L-N+1) \\ \vdots & & \ddots & \vdots \\ u(k) & u(k-1) & \dots & u(k-N) \end{bmatrix} \tag{22}$$

$$= \begin{bmatrix} \mathbf{u}^T(k-L) \\ \mathbf{u}^T(k-L+1) \\ \vdots \\ \mathbf{u}^T(k) \end{bmatrix}, \qquad \mathbf{u}(k) = \begin{bmatrix} u(k) \\ \vdots \\ u(k-N) \end{bmatrix}. \tag{23}$$

If $L \geqslant N$, the solution to Eq. 19 can be written as

$$\hat{\mathbf{f}}_{\text{LS}} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{y} \tag{24}$$

$$= (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T(\mathbf{U}\mathbf{f}+\mathbf{v}) \tag{25}$$

$$= \mathbf{f} + \underbrace{(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{v}}_{\text{bias}} \tag{26}$$

If v is correlated with u, a bias term appears in Eq. 26 and hence $\hat{\mathbf{f}}_{\text{LS}}$ differs from the ideal solution $\mathbf{f}$ that annihilates the feedback signal. Whereas in acoustic echo cancellation v is typically uncorrelated with u, in feedback cancellation, due to the nature of the setup, there is a strong correlation between the amplified signal u and the signal to be amplified v. In order to reduce the correlation nonlinearities or a delay (Δ) can be inserted in the signal path [16].
In our experiments we also did simulations with two standard adaptive algorithms, which serve as a reference point. The algorithms we applied, were the [5]

- normalised least–mean–squares algorithm
- exponentially–weighted recursive least–squares algorithm (standard implementation)

2 Linear prediction based approach

2.1 Filtered–X scheme : prewhitened least–squares

If now v is the output of an autoregressive (AR) process, then $V(z)$ can be written as

$$V(z) = \frac{1}{H(z)}W(z), \tag{27}$$

where $W(z)$ is a white time sequence and $1/H(z)$ is a stable all–pole filter of order P :

$$H(z) = 1 + z^{-1}h_1 + \cdots + z^{-P}h_P. \tag{28}$$

This is a realistic assumption for the application we have in mind, i.e. a hearing aid device. It is known that speech signals can be accurately described by autoregressive models, which leads to the so–called source–filter model for speech [1].
Under these assumptions, w can be written as

$$\mathbf{w} = \mathbf{H}\mathbf{v}, \tag{29}$$

with

$$\mathbf{w} = \begin{bmatrix} w(k-L+P) \\ \vdots \\ w(k) \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v(k-L) \\ \vdots \\ v(k) \end{bmatrix} \tag{30}$$

and

$$\mathbf{H} = \begin{bmatrix} h_P & \dots & h_1 & 1 & 0 & 0 & \dots & 0 \\ 0 & h_P & \dots & h_1 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & & \ddots & & & & \vdots \\ 0 & & \dots & & 0 & h_P & \dots & 1 \end{bmatrix}. \tag{31}$$

Hence, Eqs. 19 and 20 can be reformulated as

$$\mathbf{HU}\hat{\mathbf{f}} \stackrel{LS}{=} \mathbf{Hy}, \tag{32}$$

$$\stackrel{LS}{=} \mathbf{HUf} + \underbrace{\mathbf{Hv}}_{\mathbf{w}}, \tag{33}$$

which leads to

$$\hat{\mathbf{f}} = (\mathbf{U}^T\mathbf{H}^T\mathbf{HU})^{-1}\mathbf{U}^T\mathbf{H}^T\mathbf{Hy} \tag{34}$$

$$= (\mathbf{U}^T\mathbf{H}^T\mathbf{HU})^{-1}\mathbf{U}^T\mathbf{H}^T(\mathbf{HUf} + \mathbf{w}) \tag{35}$$

$$= \mathbf{f} + \underbrace{(\mathbf{U}^T\mathbf{H}^T\mathbf{HU})^{-1}\mathbf{U}^T\mathbf{H}^T\mathbf{w}}_{0} \tag{36}$$

Observe that the bias term in Eq. 36 goes to zero if $L \to \infty$, due to the nature of the autoregressive process which lies on the basis of signal v : the residual (innovation) w is white and uncorrelated with the system input u, which is a delayed amplified replica of v. For finite L a residual bias will remain, which, in general, will be lower than in the case when no prewhitening is applied.

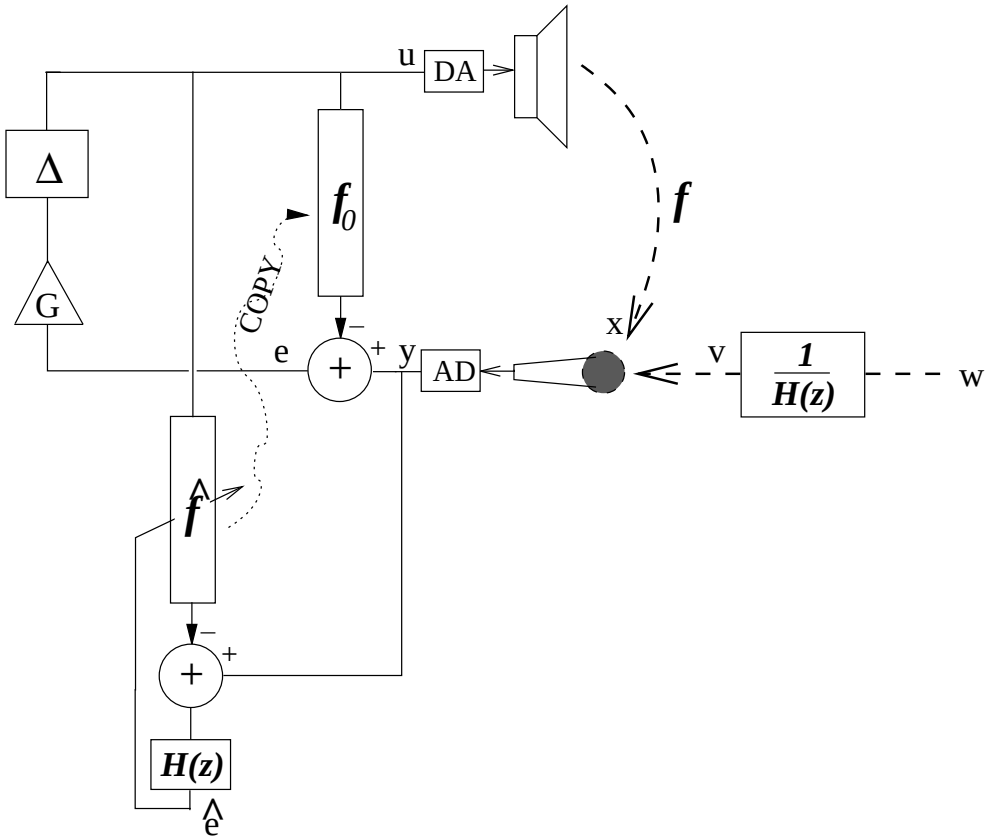

Figure 3: Filtered–X scheme

An adaptive filtered–X based scheme can be derived for the prewhitening based algorithm that has been presented in Eqs. 27–36. The scheme is schematically represented in figure 3. The adaptive filter will converge until the bias level is reached. From then on, it will move around its Wiener solution on the rhythm of the near–end disturbance v. It is furthermore shown in [6] and [10] that this approach exhibits many desirable properties.

2.2 2–channel linear prediction based identification scheme

A fully automatic filtered–X based scheme has been derived in [10]. In this scheme two filters **a** and **b** are adapted (see figure 4). It can be proven that under certain conditions filter **a** converges

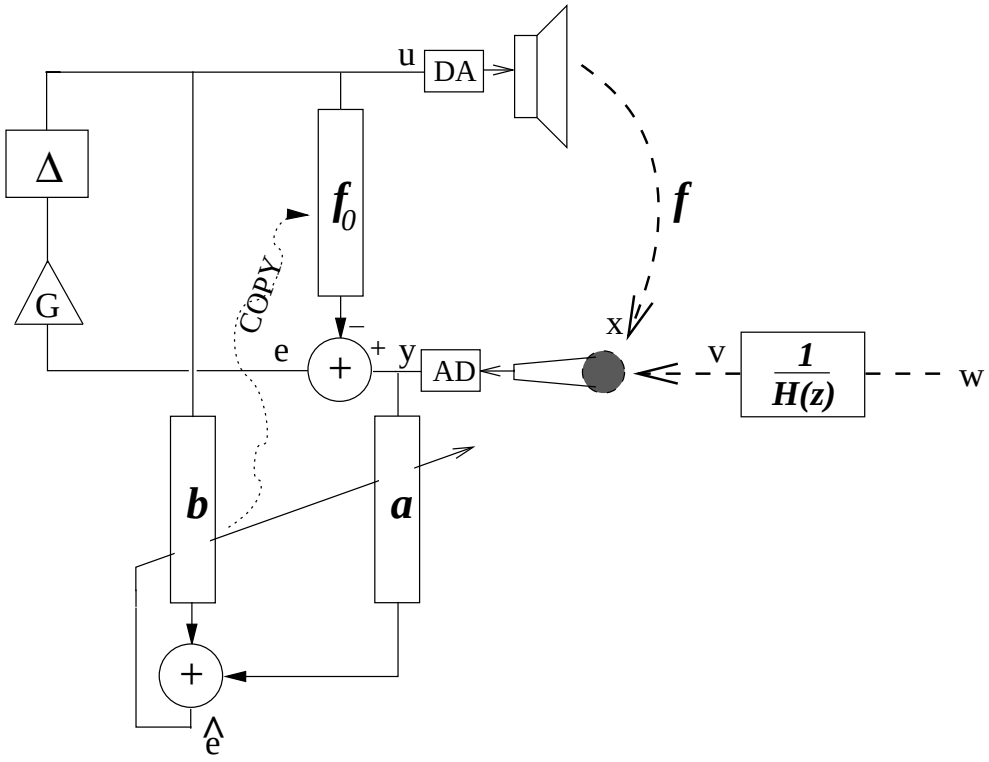

Figure 4: Fully adaptive filtered–X scheme

to the AR filter h and filter **b** approaches $-h \star f$. A sufficient condition is that the delay Δ that is inserted in the forward path is equal to or larger than the length of **a**. Further, **a** and **b** should be sufficiently long to model h and $-h \star f$.
An exponentially weighted standard RLS scheme has been derived for this algorithm. The algorithmic equations are :

$$\bar{\mathbf{X}}_{AA}^{-1}(k) = \frac{1}{\lambda^2}\mathbf{X}_{AA}^{-1} \tag{37}$$

$$\mathbf{X}_{AA}^{-1}(k+1) = \bar{\mathbf{X}}_{AA}^{-1}(k) - \frac{\bar{\mathbf{X}}_{AA}^{-1}(k)\mathbf{v}_{k+1}\mathbf{v}_{k+1}^T\bar{\mathbf{X}}_{AA}^{-1}(k)}{1+\mathbf{v}_{k+1}^T\bar{\mathbf{X}}_{AA}^{-1}(k)\mathbf{v}_{k+1}} \tag{38}$$

$$\begin{bmatrix}\mathbf{b}_{k+1}\\ \mathbf{a}_{k+1}\end{bmatrix} = \begin{bmatrix}\mathbf{b}_k\\ \mathbf{a}_k\end{bmatrix} + \mathbf{X}_{AA}^{-1}(k+1)\mathbf{v}_{k+1}\left(y(k+1) - \mathbf{v}_{k+1}^T\begin{bmatrix}\mathbf{b}_k\\ \mathbf{a}_k\end{bmatrix}\right). \tag{39}$$

Vector $\mathbf{v}_{k+1}$ is defined as

$$\mathbf{v}_{k+1} = \begin{bmatrix} \mathbf{u}_{k+1} \\ \mathbf{y}_k \end{bmatrix}, \qquad \text{with } \mathbf{u}_k = \begin{bmatrix} u(k) \\ \vdots \\ u(k-N_b) \end{bmatrix}, \quad \mathbf{y}_k = \begin{bmatrix} y(k) \\ \vdots \\ y(k-N_a) \end{bmatrix}. \tag{40}$$

Further, $\mathbf{a}_k$ and $\mathbf{b}_k$ are vectors of length $N_a + 1$ and $N_b + 1$, respectively.
Also an LMS updating scheme can be derived, defined by

$$\begin{bmatrix} \mathbf{b}_{k+1} \\ \mathbf{a}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_k \\ \mathbf{a}_k \end{bmatrix} + \mu \mathbf{v}_{k+1} \left(y(k+1) - \mathbf{v}_{k+1}^T \begin{bmatrix} \mathbf{b}_k \\ \mathbf{a}_k \end{bmatrix} \right). \tag{41}$$

The stepsize μ can be normalised in order to cope with signals having a high dynamic range such as speech.

3 Instrumental variable methods

Whereas a standard adaptive filtering based feedback cancellation scheme as presented in figure 2 is relatively easy to implement as adaptive filtering algorithms are well known in the literature [5], the bias term in Eq. 26 gives rise to wrong system estimates and hence reduced system performance.
Over the past two decades instrumental variable (IV) methods have been proposed as a bias–free alternative to standard least–squares techniques for parameter estimation.

3.1 Derivation and properties

Observe that Eq. 24 can be written as

$$\hat{\mathbf{f}}_{\text{LS}} = \left(\sum_{i=0}^{L} \mathbf{u}(k-i)\mathbf{u}^T(k-i) \right)^{-1} \sum_{i=0}^{L} \mathbf{u}(k-i)y(k-i). \tag{42}$$

Hence, the least–squares estimate $\hat{\mathbf{f}}_{\text{LS}}$ can be found as the solution to the following set of equations

$$\sum_{i=0}^{L} \mathbf{u}(k-i) \left(y(k-i) - \mathbf{u}^T(k-i)\hat{\mathbf{f}} \right) = \mathbf{0}, \tag{43}$$

where it is assumed that y can be written as

$$y(k-i) = \mathbf{u}^T(k-i)\mathbf{f} + v(k-i). \tag{44}$$

If now a general correlation vector $\mathbf{c}(k-i)$ is used instead of $\mathbf{u}(k-i)$, leading to

$$\sum_{i=0}^{L} \mathbf{c}(k-i) \left(y(k-i) - \mathbf{u}^T(k-i)\hat{\mathbf{f}} \right) = \mathbf{0}, \tag{45}$$

an instrumental variable method is obtained. The elements of $\mathbf{c}$ are called the instruments or the instrumental variables. Equation 42 can now be replaced by

$$\hat{\mathbf{f}}_{\text{IV}} = \left(\sum_{i=0}^{L} \mathbf{c}(k-i)\mathbf{u}^T(k-i) \right)^{-1} \sum_{i=0}^{L} \mathbf{c}(k-i)y(k-i). \tag{46}$$

If now $\hat{\mathbf{f}}_{\text{IV}}$ goes to $\mathbf{f}$ for large L, it is observed from Eqs. 44 and 45 that

$$\sum_{i=0}^{L} \mathbf{c}(k-i)v(k-i)$$

should tend to zero. Hence, in order to successfully apply Eq. 45 to the system represented by Eq. 44 the following properties are required for the instrumental variable $\mathbf{c}(k)$:

$$\boxed{\mathcal{E}\{\mathbf{c}(k)\mathbf{u}^T(k)\} \text{ is nonsingular}} \tag{47}$$

$$\boxed{\mathcal{E}\{\mathbf{c}(k)v(k)\} = \mathbf{0}.} \tag{48}$$

In other words, the instruments must be correlated with the regression variables, but be uncorrelated with the noise.
Similarly to Eqs. 24 and 26 we can write

$$\hat{\mathbf{f}}_{\text{IV}} = (\mathbf{C}^T\mathbf{U})^{-1}\mathbf{C}^T\mathbf{y} \tag{49}$$

$$= \mathbf{f} + \underbrace{(\mathbf{C}^T\mathbf{U})^{-1}\mathbf{C}^T\mathbf{v}}_{\text{bias}} \tag{50}$$

where

$$\mathbf{C}^T = \begin{bmatrix}\mathbf{c}(k-L) & \dots & \mathbf{c}(k)\end{bmatrix} \qquad \mathbf{c}(k) = \begin{bmatrix} c(k) \\ \vdots \\ c(k-N)\end{bmatrix}. \tag{51}$$

Observe from Eq. 50 that if Eq. 48 is true, an unbiased system estimate is obtained.

3.2 Instrumental variables used in systems and control

Many variants have been derived for this algorithm, mainly tailored, unfortunately, towards the application field of system identification and control. A lot of references have been found in the literature : [2] [3] [4] [7] [8] [9] [11] [12] [13] [14] [15] [17]. Most of these advanced techniques rely on specific assumptions that lead to elegant solutions in the field of systems and control. They are however not straightforwardly applicable to the acoustic feedback scheme. Instrumental variable methods for systems and control application assume e.g. that

- there is no correlation between certain variables
- the designers have the possibility to insert noisy test signals in the circuit
- the designers have full access to all signals in the scheme

We implemented one of the most promising IV–schemes for control [15], but failed to obtain good results for the acoustic feedback scenario, as some crucial assumptions could not be guaranteed.

3.3 Choice of the instruments

The big art when applying an instrumental variable method is the choice of a good instrument. Typically, the instruments are generated similarly to the underlying system model, as suggested in [7]. Furthermore, a "good" instrument is said to be data independent. Contrary to control applications, where basically every part and every signal in the (relatively) slowly time–varying system can be isolated and measured, the system under consideration in our application, however, is a high order, highly time–varying acoustic model, which is difficult to identify and to track in real time.
As shown before, the instruments should be consistent with the following two conditions :

$$\mathcal{E}\{\mathbf{c}(k)\mathbf{u}^T(k)\} \text{ is nonsingular} \tag{52}$$

$$\mathcal{E}\{\mathbf{c}(k)v(k)\} = \mathbf{0}. \tag{53}$$

In this way an unbiased estimate of the feedback path f is obtained if the number of data samples L goes to infinity. These conditions are often contradictory, and hence difficult to be fulfilled at

the same time. In the acoustic feedback problem for instance, v is correlated with u, hence Eqs. 52 and 53 tend to be incompatible : the instruments should nullify $\mathcal{E}\{\mathbf{c}(k)v(k)\}$, whilst preserving a full rank $\mathcal{E}\{\mathbf{c}(k)\mathbf{u}^T(k)\}$–matrix.
Referring to Eq. 50 the "optimal" instrument can be found as

$$\mathbf{c}_{\text{opt}}(k) = \underset{\mathbf{c}}{\text{argmin}} \left|\left|(\mathbf{C}^T\mathbf{U})^{-1}\mathbf{C}^T\mathbf{v}\right|\right|^2 \tag{54}$$

$$= \underset{\mathbf{c}}{\text{argmin}} \left|\left|\left(\sum_{i=0}^{L}\mathbf{c}(k-i)\mathbf{u}^T(k-i)\right)^{-1}\sum_{i=0}^{L}\mathbf{c}(k-i)v(k-i)\right|\right|^2. \tag{55}$$

It appears, unfortunately, that minimising Eq. 55 is far from trivial if we want the instrument to be independent of the unknown near–end signal v. Although a general solution to Eq. 55 has not been found, a number of candidate instruments have been checked :

1. **the instruments are double–filtered versions of the loudspeaker signal $\boldsymbol{u}$**, i.e. $\mathbf{C} = \mathbf{G}^T\mathbf{G}\mathbf{U}$, where the filtering matrix $\mathbf{G}$ has a Toeplitz structure as in Eq. 31, performing a filtering operation with FIR filter g. The multiplication with $\mathbf{G}^T$ performs a time–reversed filtering operation with g, which is a non–causal filtering operation. Typically, in a feedback scheme a delay Δ is inserted in the forward path. If Δ is larger than or equal to the order of g the non–causal operation can be performed. Observe that $\mathbf{G}^T\mathbf{G}$ performs a filtering operation with the auto–correlation sequence of g.
   In order to find the optimal filter g,

$$g_{\text{opt}} = \underset{g}{\text{argmin}} \left|\left|(\mathbf{U}^T\mathbf{G}^T\mathbf{G}\mathbf{U})^{-1}\mathbf{U}^T\mathbf{G}^T\mathbf{G}\mathbf{v}\right|\right|^2 \tag{56}$$

   has to be minimised. Finding a solution to Eq. 56 that is independent of v is far from easy. However, from Eq. 34 it is observed that in the case of an autoregressive near–end signal v the prewhitening approach presented in section 2.1 can be applied. This approach can be considered as an instrumental variable method where the instruments are equal to

$$\mathbf{C} = \mathbf{H}^T\mathbf{H}\mathbf{U}. \tag{57}$$

   Matrix $\mathbf{H}$ is Toeplitz and performs a filtering operation with the AR model h, where $\mathbf{H}\mathbf{v} = \mathbf{w}$. In this way, Eq. 34 becomes

$$\hat{\mathbf{f}}_{\text{IV}} = (\mathbf{C}^T\mathbf{U})^{-1}\mathbf{C}^T\mathbf{y}, \tag{58}$$

   similar to Eq. 49. Observe from Eq. 36 that the two IV–properties are fulfilled. Hence, Eq. 57 is a good solution to Eq. 55 if the time horizon L goes to infinity. Recall that the residual (innovation) w is white and hence uncorrelated with u, which is a delayed and amplified version of the the system input v. By selecting $\mathbf{C} = \mathbf{H}^T\mathbf{H}\mathbf{U}$ based on the AR model as an instrument, an IV–algorithm is obtained that is strongly related to the filtered–X scheme of section 2.1. Initial experiments have been performed with this IV-H^{T}H–algorithm. It is observed that a small bias can be obtained if L is large.

2. **the instruments are single reverse filtered versions of the loudspeaker signal $\boldsymbol{u}$**, i.e. $\mathbf{C} = \mathbf{G}^T\mathbf{U}$. Again we have to take care of the non–causality, which is easily solved by inserting an appropriate delay in the forward path. Also here the optimal filter g can be found according to

$$g_{\text{opt}} = \underset{g}{\text{argmin}} \left|\left|(\mathbf{U}^T\mathbf{G}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{G}^T\mathbf{v}\right|\right|^2, \tag{59}$$

   to which a trivial solution does not exist. If again the AR model is proposed as a prefilter, the following (more simple, compared to Eq. 57) instrument

$$\mathbf{C} = \mathbf{H}^T\mathbf{U} \tag{60}$$

   is obtained, which also meets conditions 47 and 48 (see Eq. 36). Initial experiments showed that a small bias can be obtained with this IV-H^T–algorithm if L is large.

3. **the instruments are a delayed version of the loudspeaker signal $\boldsymbol{u}$**. It is known that signals can be (partly) decorrelated by inserting a delay. This is certainly the case if the AR model is time–varying. Unfortunately, initial experiments showed that this algorithmic variant had serious convergence problems.

4. **the instruments are halve wave rectified version of the loudspeaker signal $\boldsymbol{u}$**. It is known that be inserting a nonlinearity the signals can be (partly) decorrelated. Different nonlinear operation can be considered. A very simple, yet popular one, commonly used in multi–channel acoustic echo cancellation, is a halve wave rectifier. The instruments are generated as follows :

$$c(k) = u(k) + \alpha(u(k) + |u(k)|). \tag{61}$$

Parameter α determines the amount of nonlinear distortion that is inserted. Initial experiments showed that this algorithm shows good convergence.

5. **the instruments are phase modulated version of the loudspeaker signal $\boldsymbol{u}$**. Another decorrelation technique is based on phase modulation. In this case the instruments are defined as

$$c(k) = u(k)\cos(k_m \sin(2\pi f_m k)). \tag{62}$$

Parameter k_m is the modulation depth and f_m is the modulation frequency. Initial experiments showed that this algorithm shows good convergence.

6. **the instruments are linearly frequency shifted version of the loudspeaker signal $\boldsymbol{u}$**. By linearly frequency shifting the loudspeaker signal u a decorrelated instrument $c(k)$ is obtained :

$$\begin{aligned} u_\delta(k) &= u(k - M/2) \\ u_h(k) &= \sum_{m=0}^{M-1} h_f(m) u(k-m) \\ c(k) &= u_\delta(k)\cos(2\pi\Delta_f k) - u_h(k)\sin(2\pi\Delta_f k) \end{aligned}$$

where $h_f(m)$ is an FIR filter of length M approximating a Hilbert filter, as shown in figure 5 and Δ_f is the desired frequency shift. By taking for instance the 100 most central (largest) filter coefficients of $h_f(m)$ the quality of the linear frequency shift operation is good, the audible artifacts are minimal. Figure 6 shows the frequency response of a 100–taps Hilbert filter. Observe that due to the non–causal characteristic of the Hilbert filter an extra delay has to be inserted in the forward path if $\Delta < M/2$. Unfortunately, initial experiments showed that this algorithmic variant had serious convergence problems.

3.4 Implementation schemes

Different implementation schemes have been derived for the instrumental variable approach. Starting from Eq. 49, which defines an IV estimation of the feedback path in "batch", an exponentially–weighted recursive IV updating scheme has been derived (see section 3.4.1). Also a QRD–based recursive updating algorithm has been developed, which will be presented in section section 3.4.2. Finally, a least–means squares IV variant was proposed (section 3.4.3). It was verified that the algorithms are consistent with Eq. 49 and that they reach the expected steady–state value (bias), which can be computed from

$$\hat{\mathbf{f}}_{\text{IV}} - \mathbf{f} = (\mathbf{C}^T\mathbf{U})^{-1}\mathbf{C}^T\mathbf{v}. \tag{63}$$

3.4.1 Exponentially–weighted recursive IV updating

Starting from Eq. 49

$$\hat{\mathbf{f}}_{\text{IV}}(k) = \underbrace{\left(\mathbf{C}^T(k)\mathbf{U}(k)\right)^{-1}}_{\mathbf{x}_{CU}^{-1}(k)} \underbrace{\mathbf{C}^T(k)\mathbf{y}(k)}_{\mathbf{x}_{Cy}(k)}, \tag{64}$$

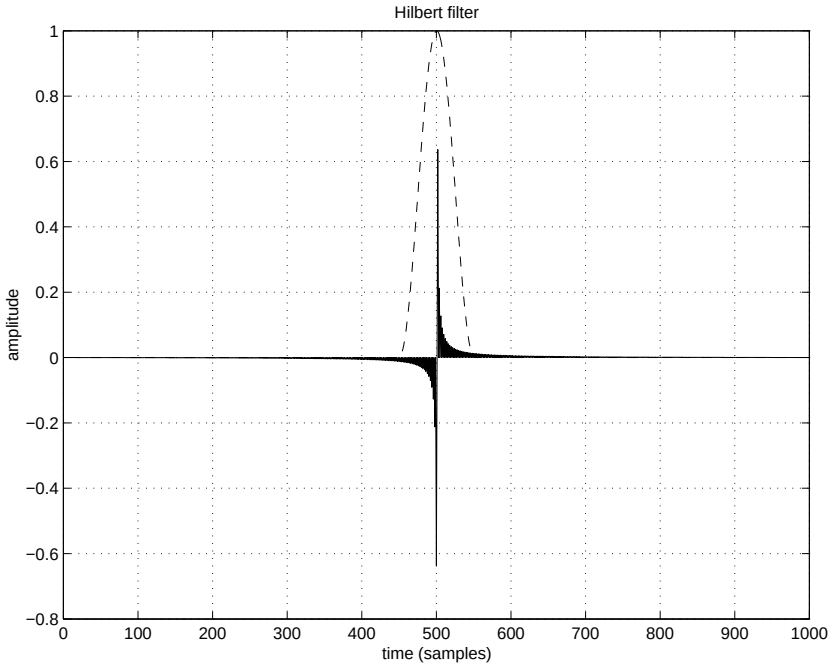

Figure 5: Hilbert filter approximation : 1000 taps

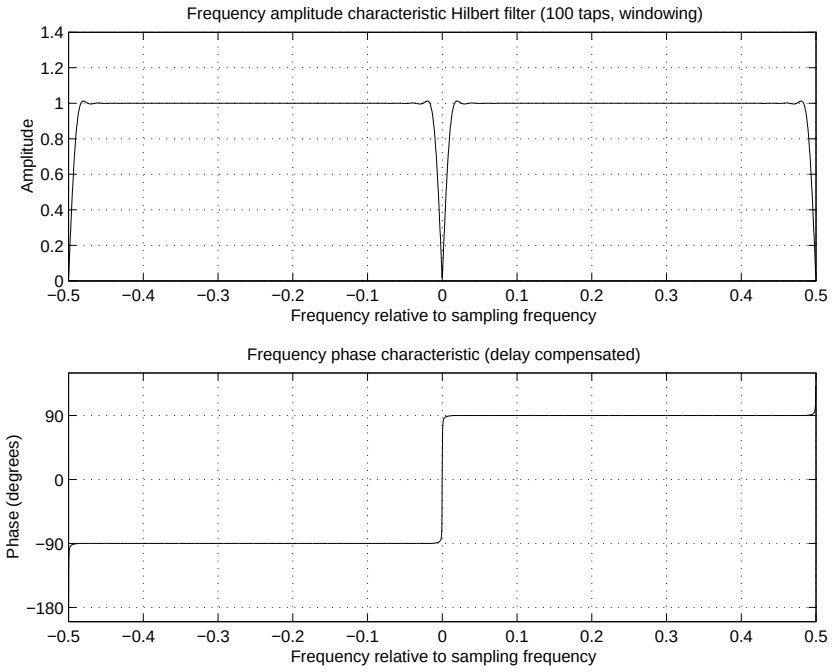

Figure 6: Hilbert filter approximation : 100 taps, frequency–domain representation

we obtain

$$\hat{\mathbf{f}}_{\text{IV}}(k+1) = \underbrace{\Big(\mathbf{C}^T(k+1)\mathbf{U}(k+1)\Big)^{-1}}_{\mathbf{X}_{CU}^{-1}(k+1)} \underbrace{\mathbf{C}^T(k+1)\mathbf{y}(k+1)}_{\mathbf{X}_{Cy}(k+1)} \tag{65}$$

with

$$\mathbf{U}(k) = \begin{bmatrix} \lambda^{k-1}\mathbf{u}^T(1) \\ \vdots \\ \mathbf{u}^T(k) \end{bmatrix}, \qquad \mathbf{C}(k) = \begin{bmatrix} \lambda^{k-1}\mathbf{c}^T(1) \\ \vdots \\ \mathbf{c}^T(k) \end{bmatrix}, \qquad \mathbf{y}(k) = \begin{bmatrix} \lambda^{k-1}y(1) \\ \vdots \\ y(k) \end{bmatrix}. \tag{66}$$

It is observed that

$$\mathbf{X}_{CU}(k+1) \;=\; \lambda^2\mathbf{X}_{CU}(k) + \mathbf{c}_{k+1}\mathbf{u}_{k+1}^T \tag{67}$$

$$\mathbf{X}_{Cy}(k+1) \;=\; \lambda^2\mathbf{X}_{Cy}(k) + \mathbf{c}_{k+1}y(k+1) \tag{68}$$

From Eq. 67 and the matrix inversion lemma it follows that

$$\boxed{\mathbf{X}_{CU}^{-1}(k+1) = \bar{\mathbf{X}}_{CU}^{-1}(k) - \frac{\bar{\mathbf{X}}_{CU}^{-1}(k)\mathbf{c}_{k+1}\mathbf{u}_{k+1}^T\bar{\mathbf{X}}_{CU}^{-1}(k)}{1+\mathbf{u}_{k+1}^T\bar{\mathbf{X}}_{CU}^{-1}(k)\mathbf{c}_{k+1}}} \tag{69}$$

with

$$\boxed{\bar{\mathbf{X}}_{CU}^{-1}(k) = \frac{1}{\lambda^2}\mathbf{X}_{CU}^{-1}.} \tag{70}$$

From Eqs. 64, 65, 67 and 68 it finally follows that

$$\boxed{\hat{\mathbf{f}}_{\text{IV}}(k+1) = \hat{\mathbf{f}}_{\text{IV}}(k) + \mathbf{X}_{CU}^{-1}(k+1)\mathbf{c}_{k+1}\left(y(k+1) - \mathbf{u}_{k+1}^T\mathbf{f}_{\text{IV}}(k)\right).} \tag{71}$$

3.4.2 QRD–based recursive IV updating

A numerically more stable QRD–based variant has been derived for the algorithm of section 3.4.1. Starting from Eq. 64 and by taking the QR decomposition of

$$\mathbf{C}(k) = \mathbf{Q}(k)\mathbf{R}(k), \qquad \mathbf{Q}(k)^T\mathbf{Q}(k) = \mathbf{I}, \tag{72}$$

it follows that

$$\hat{\mathbf{f}}_{\text{IV}}(k) = \left(\mathbf{R}^T(k)\underbrace{\mathbf{Q}^T(k)\mathbf{U}(k)}_{\mathbf{M}(k)}\right)^{-1}\underbrace{\mathbf{R}^T(k)\mathbf{Q}^T(k)\mathbf{y}(k)}_{\mathbf{z}(k)} \tag{73}$$

and hence $\hat{\mathbf{f}}_{\text{IV}}(k)$ can be found by solving the linear set of equations

$$\mathbf{M}(k)\hat{\mathbf{f}}_{\text{IV}}(k) = \mathbf{z}(k). \tag{74}$$

If we define

$$\begin{bmatrix}\mathbf{C}(k) & \mathbf{U}(k) & \mathbf{y}(k)\end{bmatrix} = \begin{bmatrix}\mathbf{Q}(k) & \mathbf{q}(k)\end{bmatrix}\begin{bmatrix}\mathbf{R}(k) & \mathbf{M}(k) & \mathbf{z}(k) \\ \mathbf{0} & \boldsymbol{\rho}^T(k) & \xi(k)\end{bmatrix} \tag{75}$$

it follows that

$$\begin{bmatrix}\mathbf{C}(k+1) & \mathbf{U}(k+1) & \mathbf{y}(k+1)\end{bmatrix} = \begin{bmatrix}\lambda\mathbf{C}(k) & \lambda\mathbf{U}(k) & \lambda\mathbf{y}(k) \\ \mathbf{c}_{k+1}^T & \mathbf{u}_{k+1}^T & y(k+1)\end{bmatrix} \tag{76}$$

$$= \begin{bmatrix}\mathbf{Q}(k) & \mathbf{q}(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\lambda\mathbf{R}(k) & \lambda\mathbf{M}(k) & \lambda\mathbf{z}(k) \\ \mathbf{0} & \lambda\boldsymbol{\rho}^T(k) & \lambda\xi(k) \\ \mathbf{c}_{k+1}^T & \mathbf{u}_{k+1}^T & y(k+1)\end{bmatrix} \tag{77}$$

which can be updated with the appropriate Givens rotation, such that

$$\boxed{\begin{bmatrix} \mathbf{R}(k+1) & \mathbf{M}(k+1) & \mathbf{z}(k+1) \\ \mathbf{0} & \star & \star \end{bmatrix} \overleftarrow{Q}_{(k+1)} \begin{bmatrix} \lambda\mathbf{R}(k) & \lambda\mathbf{M}(k) & \lambda\mathbf{z}(k) \\ \mathbf{c}_{k+1}^T & \mathbf{u}_{k+1}^T & y(k+1) \end{bmatrix}} \tag{78}$$

is obtained. A graphical representation of the algorithm is shown in figure 7. It appears that residue extraction with this scheme is not so easy and nice as with a classical QRD-SRI updating scheme. It is cheaper to compute the a–priori error explicitly as $e(k+1) = y(k+1) - \mathbf{u}_{k+1}^T \mathbf{f}_{\text{IV}}(k)$.

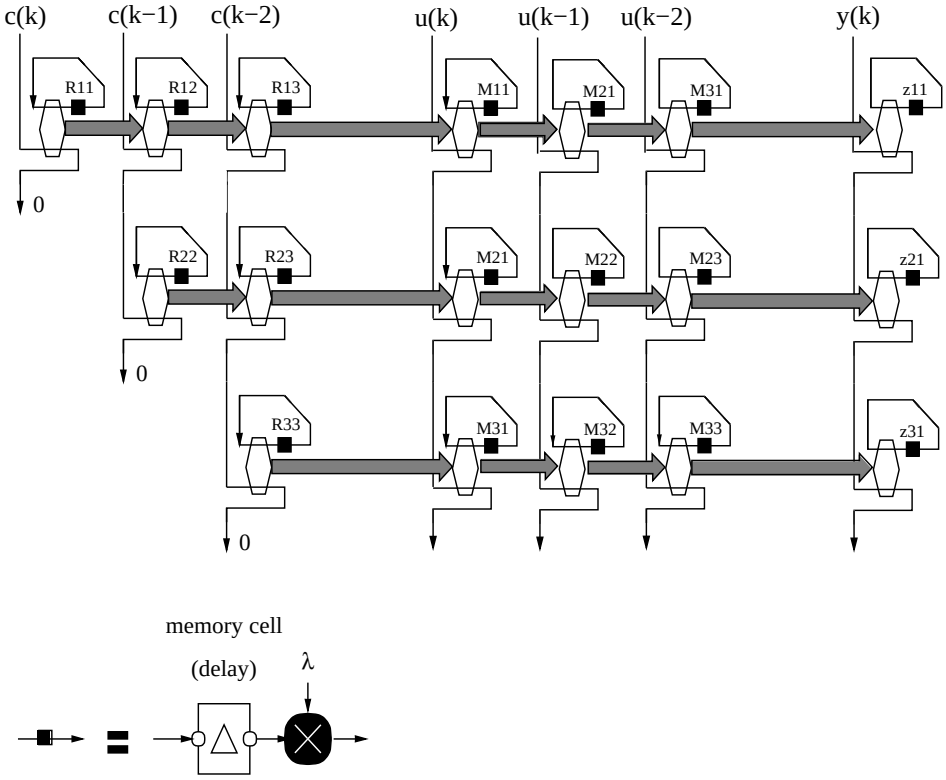

Figure 7: Square–root information recursive IV updating, $L = 2$

3.4.3 Least–means squares IV algorithm

Based on Eqs. 69–71 an LMS updating scheme can be derived for the IV–approach. The algorithmic equations are given by

$$\boxed{\hat{\mathbf{f}}_{\text{IV}}(k+1) = \hat{\mathbf{f}}_{\text{IV}}(k) + \mu\mathbf{c}_{k+1}\left(y(k+1) - \mathbf{u}_{k+1}^T\mathbf{f}_{\text{IV}}(k)\right).} \tag{79}$$

If signals with a high dynamic range such as speech are processed, the stepsize μ should be normalised. This leads to the following NLMS–IV updating scheme :

$$\boxed{\hat{\mathbf{f}}_{\text{IV}}(k+1) = \hat{\mathbf{f}}_{\text{IV}}(k) + \frac{\mu}{\alpha^2 + \mathbf{u}_{k+1}^T\mathbf{c}_{k+1}}\mathbf{c}_{k+1}\left(y(k+1) - \mathbf{u}_{k+1}^T\mathbf{f}_{\text{IV}}(k)\right).} \tag{80}$$

It can be verified that the NLMS–IV updating algorithms are not guaranteed to be stable realisations. This was observed during the simulations. In most cases however, the NLMS–IV algorithms worked fine.

3.4.4 Estimation of the AR parameters

So far it was assumed that the AR model is known. In practice, the noise colouring filter h has to be estimated in order to make the IV-H^T(H)-technique practically implementable. From Eq. 9 it follows that

$$E = \frac{1}{1-(F-F_0)G}\,\frac{1}{H}\,W. \tag{81}$$

if it is assumed that the input is an AR signal with $h \star v = w$. Now, the filter $H(z)$ can be written as $H(z) = 1 - A(z)$ where the highest power of z in $A(z)$ is z^{-1}. Hence,

$$E = \frac{1}{1-A-(F-F_0)G+A(F-F_0)G}\,W. \tag{82}$$

Observe that the error signal e is an AR signal of order $P+N+\Delta-1$. If $\Delta > P$, the first $P+1$ coefficients of

$$\hat{H}_e = 1 - A - (F-F_0)G + A(F-F_0)G \tag{83}$$

will converge to a bias–free estimate of $H(z)$ if the number of data samples is large enough. Two algorithmic variants are proposed

- identification of $H(z)$ based on a short (order P) model $\hat{H}(z)$
- identification of $H(z)$ based on a large (order $P+N+\Delta-1$) model $\hat{H}_e(z)$

As far as the online adaptation is concerned, the following cost criterion is minimised :

$$J = \min_{\mathbf{a}_k} \left(\mathbf{a}_k^T \mathbf{y}_k\right) \tag{84}$$

$$= \min_{\boldsymbol{\alpha}_k} \left(y(k) - \boldsymbol{\alpha}_k^T \mathbf{y}_{k-1}\right) \tag{85}$$

with

$$\mathbf{a}_k = \begin{bmatrix} 1 \\ \boldsymbol{\alpha}_k \end{bmatrix}. \tag{86}$$

This leads to a standard NLMS or RLS updating scheme. An advantage of this approach is that we can easily work with two adaptation stepsizes : one for the update of the AR model, which will be fast typically, and another for the update of the feedback path model, where we average over a larger time window.
It appeared in some simulations that identification of the AR model based on the microphone signal y rather than on the error e gave better results. Therefore we simulated with two versions of the algorithm : versions based on y and versions based on e.
We also derived both an exponentially–weighted standard and QRD–based RLS update scheme for a combined AR modelling and IV feedback path identification, by minimising the cost

$$\min_{\mathbf{a}_k, \mathbf{b}_k} \left(\mathbf{b}_k^T \mathbf{u}_k + \mathbf{a}_k^T \mathbf{y}_k\right), \qquad \mathbf{a}_k = \begin{bmatrix} 1 \\ \boldsymbol{\alpha}_k \end{bmatrix} \tag{87}$$

in an IV approach. This algorithm can be thought of as a mixture between the IV–schemes presented in section 3.4.1 3.4.2 and the 2–channel linear prediction algorithm of section 2.2. Unfortunately, this algorithm showed a bad performance and it was not clear how the instruments should be chosen.

4 Experiments

A large number of experiments have been carried out to compare the different IV–variants with standard algorithms and with linear prediction techniques. Only a selected number of the many experiments we did are presented in the report. The most relevant simulation results have been selected and they are displayed at the end of the report. The conclusions that could be drawn based on the experiments are summarised in this section.

4.1 Algorithms

The algorithms that have been used and compared in the experiments are summarised in table 1. For more information about the algorithms, we refer to sections 1.4, 2.1, 2.2 and 3.4.

a colour code	algorithm
RLS	exponentially–weighted standard RLS
RLS prewhitened	exponentially–weighted RLS with prewhitening (filtered–X algorithm), H is exactly known
RLS–IV H^T	exponentially–weighted RLS–IV with H^T as instruments, H is exactly known
RLS–IV $H^T H$	exponentially–weighted RLS–IV with $H^T H$ as instruments, H is exactly known
RLS 2–channel	2–channel exponentially–weighted RLS–based linear prediction algorithm, H is estimated on–line
RLS–IV H^T estim	exponentially–weighted RLS–IV with H^T as instruments, H is estimated on–line
RLS–IV $H^T H$ estim	exponentially–weighted RLS–IV with $H^T H$ as instruments, H is estimated on–line
RLS–IV halve wave	exponentially–weighted RLS–IV using a halve wave rectifier to generate the instruments
RLS–IV phase mod	exponentially–weighted RLS–IV using a phase modulator to generate the instruments
NLMS	standard NLMS
NLMS prewhitened	NLMS with prewhitening (filtered–X algorithm), H is exactly known
NLMS–IV H^T	NLMS–IV with H^T as instruments, H is exactly known
NLMS–IV $H^T H$	NLMS–IV with $H^T H$ as instruments, H is exactly known
NLMS 2–channel	2–channel NLMS–based linear prediction algorithm, H is estimated on–line
NLMS–IV H^T estim	NLMS–IV with H^T as instruments, H is estimated on–line
NLMS–IV $H^T H$ estim	NLMS–IV with $H^T H$ as instruments, H is estimated on–line
NLMS–IV halve wave	NLMS–IV using a halve wave rectifier to generate the instruments
NLMS–IV phase mod	NLMS–IV using a phase modulator to generate the instruments

Table 1: Feedback cancellation algorithms used in the experiments

The algorithmic parameters were set such that the algorithms operate properly and such that a fair comparison can be made between them. The colour code corresponds to the label which is assigned to the algorithm and which is used in the figures 8–40 containing the simulation results. Remark that the algorithms can be divided in different categories :

- RLS– versus LMS–like algorithms
- standard algorithms versus linear prediction algorithms versus IV–algorithms
- algorithms that make use of the exact AR–model H versus algorithms that estimate H on–line

4.2 Simulation environment

At this stage of the project we have evaluated the feedback cancellation algorithms of table 1 in a controlled simulated, yet realistic testing environment. So far, the algorithms have been tested

1. **in a standard adaptive filtering setup** (no feedback scenario), basically as a first test to verify the stability of the algorithms and to initially tune some algorithmic parameters

2. **in a stationary closed–loop setup as presented in figure 2 with a stationary AR input signal** v, with w a Gaussian distributed white noise signal. The AR model was selected such that the largest pole of $H^{-1}(z)$ lay between 0.95 and 0.995. The order of the AR model was set to $P = 9$. The feedback path is a fixed, randomly generated FIR filter of order 29 with decreasing amplitude such that the variance of the 30th filter coefficient is about 10% of the variance of the first coefficient. The adaptive filter is an FIR filter with 30 filter coefficients (sufficient order). The forgetting factor for the RLS–like algorithms was set to 1, both for the adaptive filter as for the AR estimation. The stepsize of the LMS–variants was fixed to a realistic value. The amplifier gain was tuned such that it stayed (just) below the edge of instability. A delay $\Delta = 10$, equal to the length of the AR modelling filter was inserted in the forward path. The algorithms that estimate $H(z)$ use the exact AR model order 9 (short model) or a long combined AR–feedback path model of $\Delta + N + P - 1 = 48$.

3. **in a stationary closed–loop setup as presented in figure 2, but with a time–varying AR input signal**. Again, w is a Gaussian distributed white noise signal. New AR coefficients were computed based on the old coefficients by adding a small random disturbance and preserving stability (largest pole smaller than 0.995). The speed at which the AR coefficients vary, was controlled by

   - the amplitude of the added random disturbance
   - the frequency at which the AR coefficients were updated.

   In our experiments the amplitude was fixed to 0.01. The frequency was changed between 1/100 and 1. An update frequency 1/100 means that every 100 samples a new AR process is computed. The order of the AR model was set to $P = 9$. The feedback path is a fixed, randomly generated FIR filter of order 29 with decreasing amplitude such that the variance of the 30th filter coefficient is about 10% of the variance of the first coefficient. The adaptive filter is an FIR filter with 30 filter coefficients (sufficient order). The forgetting factor for the RLS–like algorithms was set to 0.999 for the adaptive filter and to 0.995 for the AR estimation. The stepsize of the LMS–variants were also fixed to realistic values, taking the stepsize for the AR estimation about twice as large as the stepsize for the update of the feedback model. Also in this case, the amplifier gain was tuned such that it stayed (just) below the edge of instability. A delay $\Delta = 10$, equal to the length of the AR modelling filter was inserted in the forward path. The algorithms that estimate $H(z)$ use the exact AR model order 9 (short model) or a long combined AR–feedback path model of $\Delta + N + P - 1 = 48$.

The rest of this section is dedicated to a discussion of the simulation results. Important to note is that the simulation results (performance indices) and hence the conclusions (apparently) strongly dependent on the environmental parameters, unfortunately. We tried to reduce the variation on δ_1 and δ_2 by fixing some parameters, but found that basically all parameters (feedback path, AR model, input signal w, ...) had a considerable impact on the performance indices. This is probably due to the inherent feedback structure of the scheme and the definition of the performance measures, as can be seen in Eqs. 11 and 14 for instance. Making strong assessments and a quality ranking of the algorithms based on a single experiment would lead to unreliable statements. Therefore, based on the many experiments that have been carried out, we selected typical plots and included them in this report. Sometimes different plots are included with the same environmental settings, just to show the variability of convergence curves (compare for instance figures 13–20). Based on these typical plots, with "mean" behaviour our conclusions have been drawn.

4.3 Estimation of the AR parameters

As far as the estimation of the AR parameters in NLMS/RLS-IV-H^T(H) algorithms is concerned, the following conclusions can be drawn :

- in an off–line experiment (open–loop) all algorithms work fine : the larger the time window (number of data samples) the smaller the bias. Signal w acts as a near–end disturbance and

hence the bias does not go to zero for finite length buffers. A forgetting factor $\lambda = 1$ gives good results for large time windows. If $\lambda < 1$, the algorithm behaves like a batch process with length $1/(1-\lambda)$, and has hence worse performance.

- if the AR parameters are estimated on a AR signal using a small model (with the model order equal to the order of the true AR process), then there is a significant bias, but the results are acceptable. The bias is more or less the same for RLS and NLMS and is not reduced if larger batches (larger time window) are processed. If the model order equals to the order of the true AR process an RLS implementation is feasible in terms of computational complexity.

- if the AR parameters are estimated on a microphone signal with using a large model (with the model order equal to $P + \Delta + N - 1$), then the bias goes to 0 is large windows are applied. Both NLMS and RLS show a good performance, the RLS algorithm outperforming the NLMS. If the model order equals $P + \Delta + N - 1$ only the NLMS algorithm is feasible in terms of computational complexity. The NLMS algorithm with model order equal to $P + \Delta + N - 1$ gives a better performance than the RLS implementation with order=P. On the other hand, if time–varying AR models have to be estimated a small order may be sufficient, as large models cannot track the time variations anyway.

- from a complexity/performance point of view it is preferable

  – to estimate the AR model with an NLMS algorithm.

    It appears that an NLMS algorithm with $P + \Delta + N$ coefficients, in general, leads to a slightly worse overall performance than if only $P + 1$ coefficients are computed. A possible explanation could be that although a large filter order $P + \Delta + N - 1$ leads to a bias–free estimate of H (in theory only for infinite length data buffers !), the presence of the near–end disturbance (signal v) diverts the coefficients from their Wiener solution. This effect seems to be larger than in the case of a short NLMS filter with $P + 1$ coefficients.

  OR

  – to use an RLS algorithm to estimate the AR coefficients with P coefficients. The RLS algorithm with $P + \Delta + N$ coefficients should be rejected from a computational complexity point of view.

- It was furthermore found that the NLMS algorithm with $P+1$ coefficients is most appropriate to estimate the AR model if we take H^T as instruments. For the instruments $H^T H$ the RLS algorithm with $P + 1$ coefficients gave the best performance.

- A disadvantage of the NLMS approach to estimate the AR coefficients is that it may be slower than RLS as far as tracking of highly time–varying AR models is concerned.

4.4 Closed–loop identification : stationary AR input

NLMS– and RLS–variants of basically all algorithms (see table 1) have been considered for this configuration. Both off–line and on–line identification procedures have been considered on a closed–loop scheme with a stationary AR input. The results are shown in figures 8–20 on pages 24–36. The main conclusions are :

- standard algorithms (NLMS/RLS) on the average have the highest bias

- RLS–variants are better than their LMS–counterpart

- algorithms that use exact knowledge of the AR model show the best performance

- the performance of fully adaptive IV– and LP–algorithms lies in between that of the standard algorithms (NLMS/RLS) and the algorithms that use exact knowledge of the AR model

- IV–algorithms based on H^T or $H^T H$ have a similar performance as their linear prediction algorithm equivalents, but are in general somewhat worse. This is certainly the case if short AR modelling filters are used. This similar performance is not unexpected, given the instruments that are being used.
- in general RLS-IV-H^{T}H is better than RLS-IV-H^T, although the difference is small.
- experiments carried out in on–line mode show similar results as could be obtained in off–line mode. However, apparently there is less variability in the performance of the algorithms : the different curves lie closer to each other.
- NLMS-IV-H^T with estimated AR model using RLS gets sometimes unstable when the order of the AR estimation filter equals the order P of the true AR model.
- for standard NLMS and RLS the bias in steady–state is almost the same. For the other algorithms (that use or estimate H) RLS–equivalents lead to a smaller bias.
- both performance indices give, in general, consistent results
- in general the IV–algorithms score less good on the $||e - n||$–plot
- observe that using different data leads to different plots. Compare for instance figures 13–20 : sometimes NLMS/RLS 2-channel is better than NLMS/RLS-IV-H^T(H), sometimes it is worse.
- it appeared that in some simulations identification of the AR model based on the microphone signal y rather than on the error e gave better results, especially for the RLS type of algorithms (compare figure 13 with 14 for instance).
- in an off–line experiment RLS can converge to a filter $\hat{f}$ that leads to an unstable closed loop if at a certain moment it is copied to f_0

Also two "nonlinear" instrumental variable algorithms have been taken into consideration (see also section 3.3) :

- an instrumental variable technique that generates instruments by halve wave rectifying the loudspeaker signal u.
- an instrumental variable technique that generates instruments by phase modulating the loudspeaker signal u.

An NLMS– and an RLS–variant of both algorithms have been compared with standard NLMS/RLS and with the filtered–X scheme that uses the exact AR model. Both off–line and on–line identification have been considered on a closed–loop scheme with a stationary AR input. The results are shown in figures 21–24 on pages 37–40.
It is observed that with this kind of algorithm the bias can only be marginally improved : for both performance indices the algorithms hardly score better than standard NLMS/RLS, leaving the curves for the filtered–X scheme far below. The algorithm with the phase modulated instruments is typically somewhat better than the algorithm that uses halve wave rectification.

4.5 Closed–loop identification : non–stationary AR input

To this configuration all algorithms of table 1 have been applied, except the "nonlinear" IV–algorithms due to the marginal improvement that is expected with this techniques (see also section 4.4). Both off–line and on–line identification procedures have been implemented. The results are shown in figures 25–41 on pages 41–57. Note that

- standard algorithms (NLMS/RLS) have the highest bias only if the AR modelling coefficients update frequency is low. Standard algorithms perform relatively well in a highly non–stationary environment. A reason for this may be that the loudspeaker and the microphone signal get uncorrelated due to the fact that the AR coefficients change continuously and that there is a delay in the forward path.
- it was observed that NLMS–IV algorithms sometimes get unstable typically when short (order P) AR modelling filters are used and if the AR modelling coefficients update frequency is increased.
- RLS–variants are better than their LMS–counterpart although the difference is relatively small
- the filtered–X algorithms that use exact knowledge of the AR model typically show the best performance
- NLMS/RLS–IV algorithms that use the exact AR model have a worse performance than under stationary AR conditions. Sometimes the algorithms using the exact AR model are even inferior to their fully adaptive NLMS/RLS–IV equivalents. This is probably due to the fact that in a time–varying AR environment with delay the different signal samples have been generated by different AR models.
- fully adaptive IV–algorithms based on H^T or $H^T H$ have a similar performance as their linear prediction algorithm equivalent, but are now in general somewhat better.
- in general RLS-IV-H^TH is better than RLS-IV-H^T, although the difference is typically small.
- experiments carried out in on–line mode show similar results as in off–line mode.
- both performance indices give, in general, consistent results
- typically, the algorithms using short (order P) AR modelling score better on δ_1, whereas the algorithms using long (order $P + \Delta + N - 1$) AR modelling score better on δ_2
- as the AR modelling coefficients update frequency is increased, there is less spreading in the different performance curves
- also in this case different plots are obtained if other data sets are used. Compare for instance figure 26 with 29 and figure 27 with 30.

5 Open issues and future research

A number of open issues are formulated in this section :

- so far (see section 4.2) the algorithms have been evaluated in a simulated stationary feedback scenario only, both with stationary as well as with time–varying AR inputs. It should verified which problems occur when near–AR signals or true speech is applied.
- still it is not been completely clear how to choose appropriate instruments. A more fundamental and theoretical analysis is lacking.
- so far we only considered stationary feedback paths of small order. It is not clear how the algorithm will behave on more realistic or time–varying acoustic paths.
- RLS algorithmic variants show the best performance, but demand a high computational load. It is not clear whether fast RLS schemes can be derived for the instrumental variable or linear prediction approach.

- it was observed that if the AR input characteristics strongly change with time standard algorithmic schemes such the NLMS and the RLS algorithm perform remarkably well. Do advanced algorithms such IV–techniques and linear prediction schemes still make sense in a highly non–stationary acoustic environment ?
- NLMS-IV-H^T(H) algorithms are not guaranteed to be stable realisations. It should be analysed more in detail when and if stability can be obtained.

In the next workpackage linear prediction algorithms will be investigated. As can be observed from the experimental results presented in section 4, linear prediction schemes have been included in our simulations and some initial results are already available. As the most promising IV–schemes that came out of this research are strongly related to linear prediction algorithms it can be stated that both research paths rather coincide. Future research activities will hence be focused on the evaluation of both linear prediction and IV algorithms in more realistic/real–life scenarios.

6 Conclusions

Standard adaptive filtering techniques such as the NLMS and the RLS algorithm have been successfully applied to cancel acoustic feedback in public address systems and hearing aids. However, due to the inherent correlation between the input signals of the adaptive filter the acoustic model is biased and hence reduced system performance is observed.
Over the past two decades instrumental variable methods have been proposed as a bias–free alternative to standard least–squares parameter estimation. The proposed techniques however are mainly focused towards the application field of system identification and control and typically rely on assumptions that are not fulfilled in the acoustic feedback context.
In this report instrumental variable methods have been studied that can be used to cancel out acoustic feedback. It appeared that the behaviour of IV–algorithms strongly depends on the quality of the instruments. Different instruments have therefore been considered. It is was found that in the case of AR input signals such as speech an inverse filtering with the AR model is the most appropriate choice for the instruments. It was shown that in this way the two bias–free conditions specified in Eqs. 47 and 48 can be fulfilled for large time–window lengths. It was indicated that with this choice for the instruments in fact an alternative implementation scheme is obtained for the filtered–X algorithm, which forms the base of the linear prediction based approach that will be studied in the next workpackage.
Differrent implementation schemes have been presented for the instrumental variable approach. An exponentially–weighted standard RLS variant has been derived, as well as a square–root information algorithm. Also an LMS updating scheme has been proposed. For the IV–algorithm to work in a fully adaptive way an AR model estimation procedure has been provided. Both an LMS and an RLS algorithm have been derived.
A large number of experiments have been carried out to compare the different IV–variants with standard algorithms. The algorithms were tested in a simulated stationary feedback context, both with stationary and time–varying AR inputs. The main conclusions that came out of the experiments are :

- standard algorithms (NLMS/RLS) on the average have the highest bias
- RLS–variants are better than their LMS–counterpart
- algorithms that use exact knowledge of the AR model show the best performance
- IV–algorithms based on H^T or H^TH have a similar performance as their linear prediction algorithm equivalents, but are in general somewhat worse
- IV–algorithms based on H^TH are typically slightly better and have a less nervous behaviour than IV–algorithms based on H^T

- the performance of fully adaptive IV– and LP–algorithms lies in between that of the standard algorithms (NLMS/RLS) and the algorithms that use exact knowledge of the AR model
- IV–algorithms based on a "nonlinear" instrument hardly give extra bias reduction
- instability sometimes occurs with NLMS–IV algorithms
- the performance of the algorithms in an on–line scenario is comparable to that in an off–line scenario
- AR model estimation based on a short (order P) model does not significantly differ from AR model estimation using a long (order $P + \Delta + N - 1$) model.

As the most promising IV–schemes that came out of this research are strongly related to linear prediction algorithms future research activities will focus on the evaluation of both linear prediction and IV algorithms in more realistic/real–life scenarios.

References

[1] J. Deller, J. Proakis, and J. Hansen. *Discrete–Time Processing of Speech Signals*. Prentice Hall, Englewood Cliffs, New Jersey, 1987.

[2] U. Forssell. *Closed-Loop Identification: Methods, Theory, and Applications*. PhD thesis, Linköping University, SE-581 83 Linköping, Sweden, 1999.

[3] P. Guo Zou and L. Shi Du. An Optimal Instrumental Variable Method for ARMA Spectral Estimation. *IEEE Transactions on Signal Processing*, 39(12):2728–2733, December 1991.

[4] T. Gustafsson. An Instrumental Variable Based Subspace Tracking Algorithm Based on Subspace Averaging. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP98)*, pages 1721–1724, Seattle, WA, USA, May 1998.

[5] S. Haykin. *Adaptive Filter Theory, 3th ed.* Prentice Hall, Englewood Cliffs, New Jersey, 1996.

[6] J. Hellgren and U. Forssell. Bias of feedback cancellation algorithms in hearing aids based on direct closed loop identification. *IEEE Trans. Speech Audio Processing*, 9(7):906–913, November 2001.

[7] L. Ljung. *System identification: theory for the user*. Prentice Hall, Englewood Cliffs, New Jersey, 1987.

[8] T. Söderström and P. Stoica. On Some System Identification Techniques for Adaptive Filtering. *IEEE Transactions on Circuits and Systems*, 35(4):457–461, April 1988.

[9] T. Söderström, P. Stoica, and E. Trulsson. Instrumental variable methods for closed loop systems. In *Proc. 10th Congress of IFAC*, volume 14.5-6, pages 364–369, Munich, Germany, 1987.

[10] A. Spriet, M. Moonen, and I. Proudler. Feedback cancellation in hearing aids: an unbiased modelling approach. In *Proceedings of the European Signal Processing Conference (EUSIPCO02)*, pages 531–534, Toulouse, France, September 2002.

[11] P. Stoica and M. Jansson. A Transfer Function Approach to MIMO System Identification. In *Proceedings of the 38th Conference on Decision & Control (CDC99)*, pages 2400–2405, Phoenix, AR, USA, December 1999.

[12] P. Stoica and M. Jansson. MIMO System Identification: State-Space and Subspace Approximations versus Transfer Function and Instrumental Variables. *IEEE Transactions on Signal Processing*, 48(11):3087–3099, November 2000.

[13] P. Stoica, V. Šimonytė, and T. Söderström. Study of a bias-free least-squares parameter estimator, 16 pp. March 1994.

[14] P. Stoica, M. Viberg, M. Wong, and Q. Wu. Optimal IV-SSF approach to array signal processing in colored noise fields. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP95)*, pages 2088–2091, Detroit, MI, USA, May 1995.

[15] P. Van den Hof and M. Gilson. Closed–loop system identification via a tailor–made IV method. In *Proceedings of the 40th IEEE Conference on Decision and Control (CDC01)*, pages 4314–4319, Orlando, FL, USA, December 2001.

[16] T. van Waterschoot, G. Rombouts, K. Eneman, and M. Moonen. Feedback Control : State-of-the-art, AHAS–Report 1, 2003.

[17] Y. Zhang, T. Lie, and C. Soh. Consistent parameter estimation of systems disturbed by correlated noise. *IEE Proc.-Control Theory Appl.*, 144(1):40–44, January 1997.

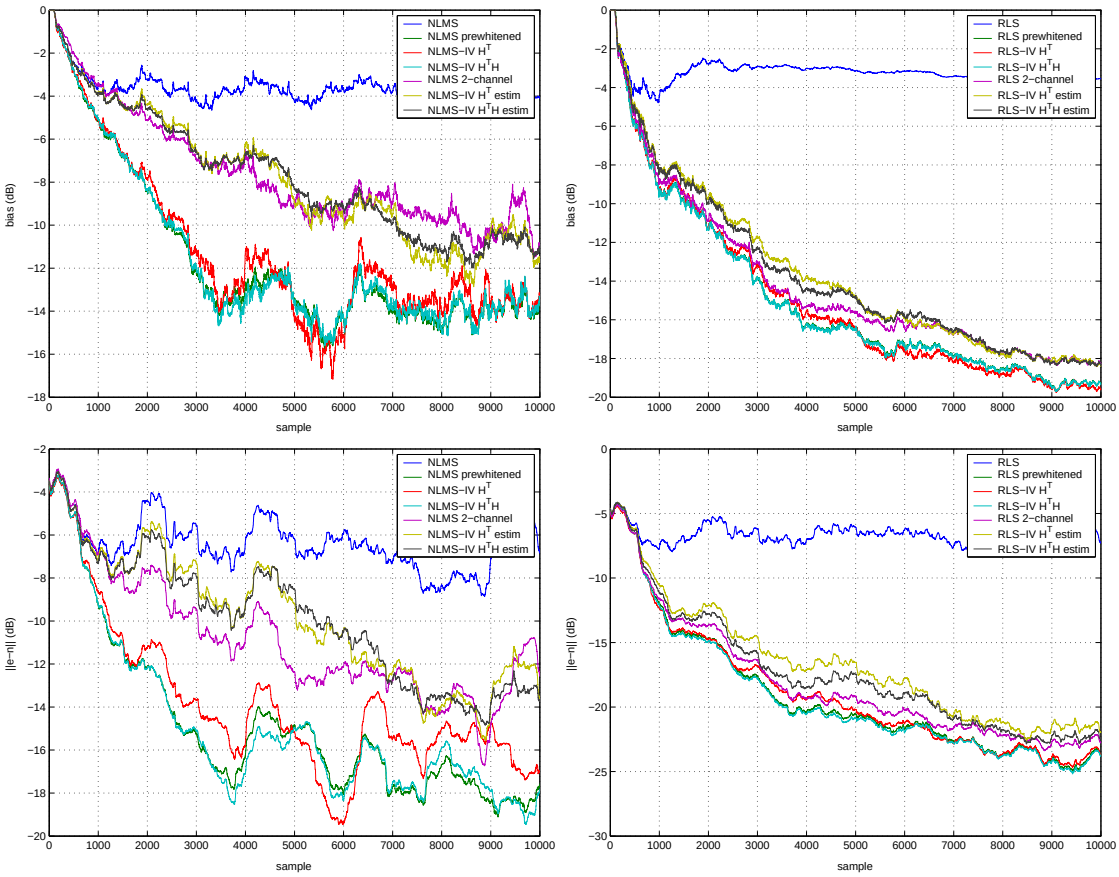

Figure 8: Typical off–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model.

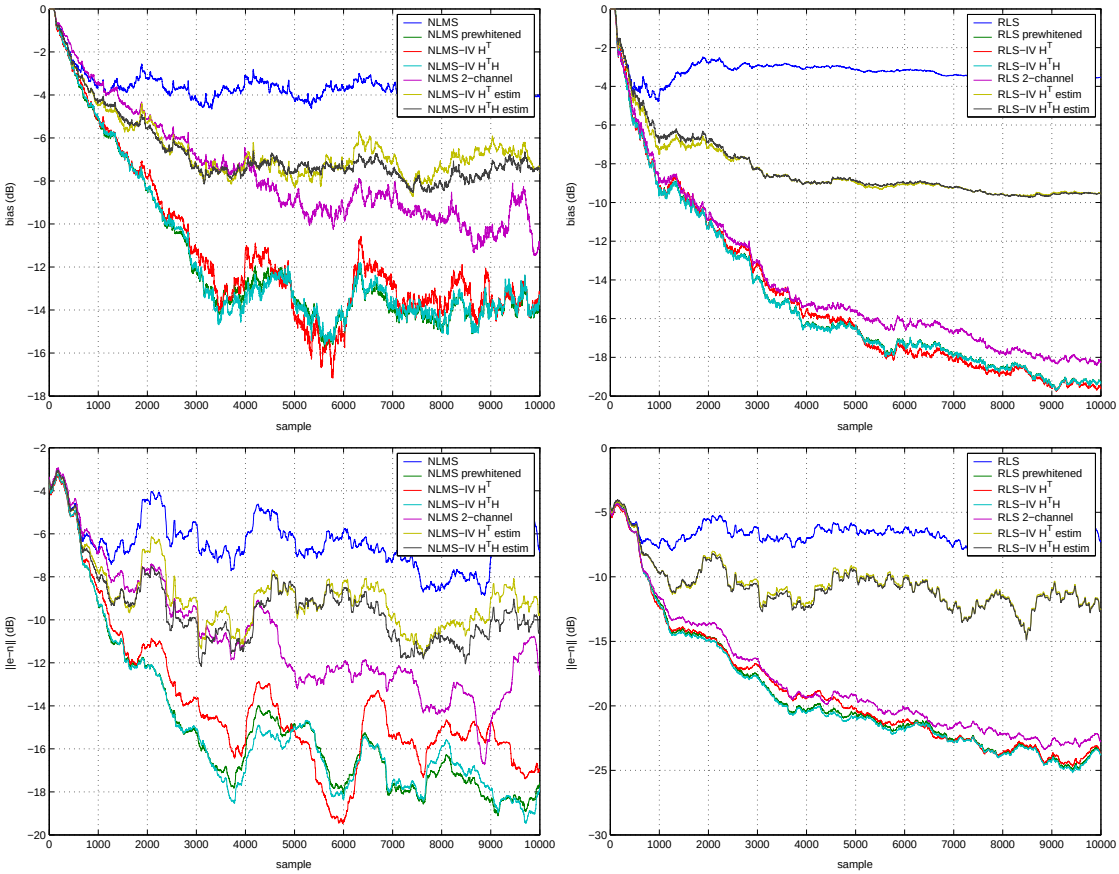

Figure 9: Typical off–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model.

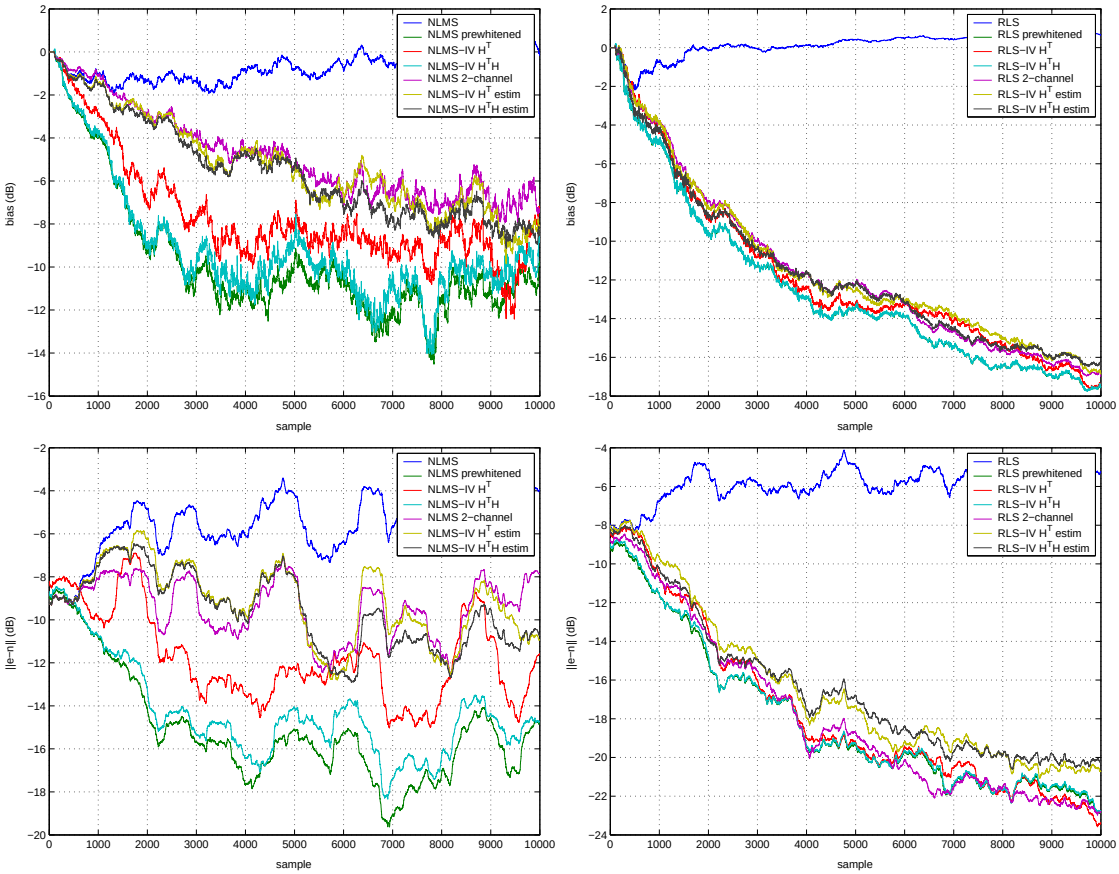

Figure 10: Another example of off–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model.

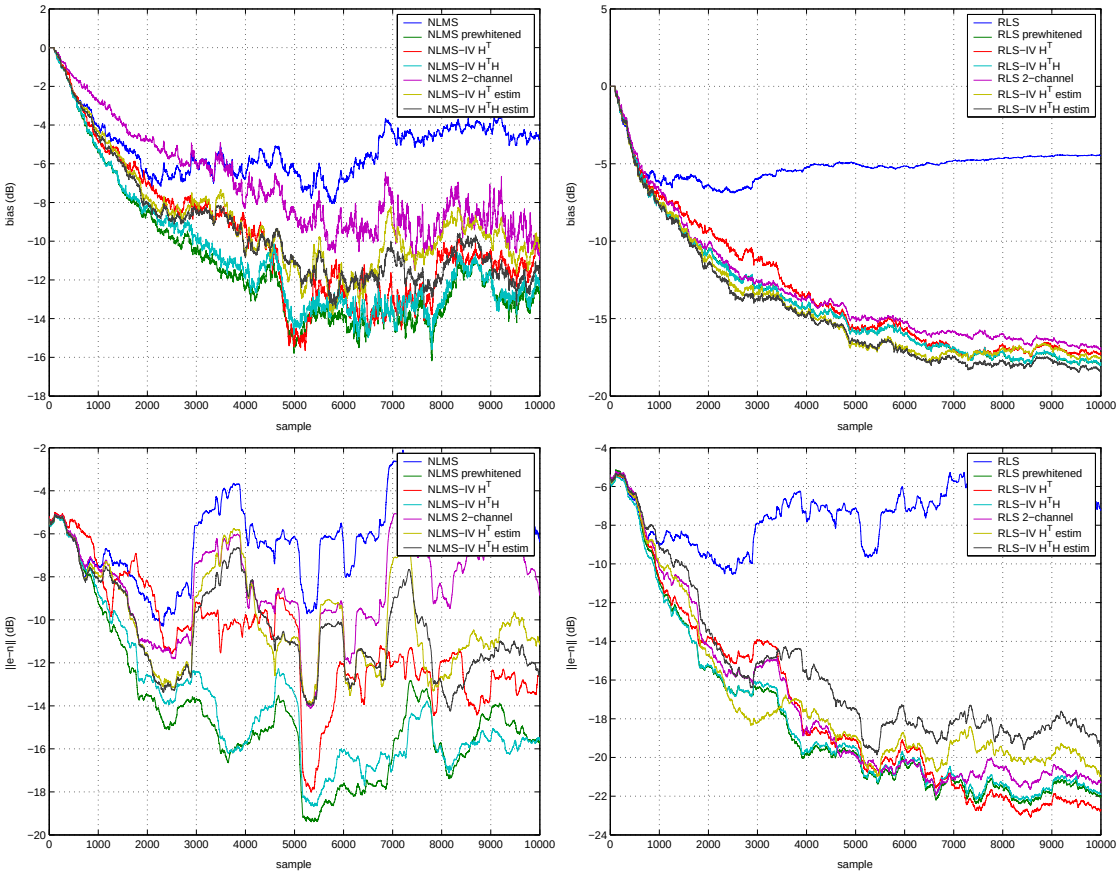

Figure 11: Yet another example of off–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model.

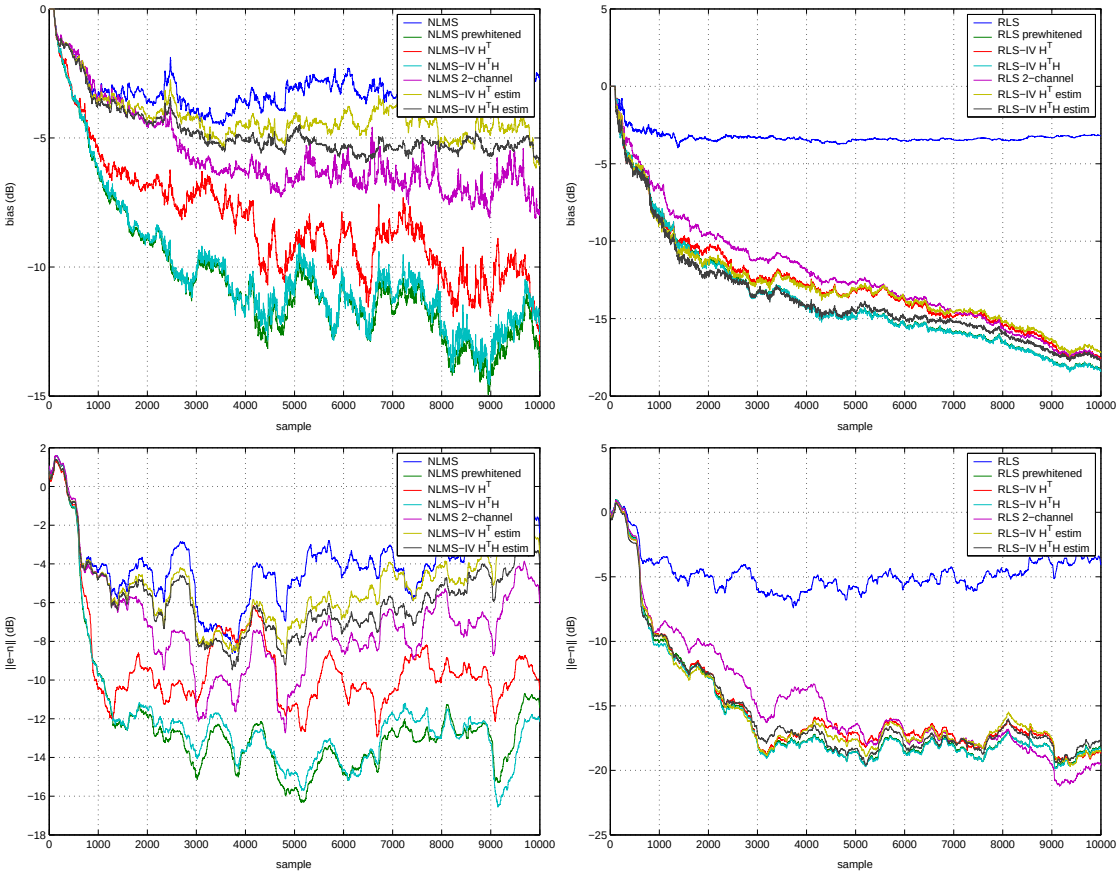

Figure 12: Yet another example of off–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model.

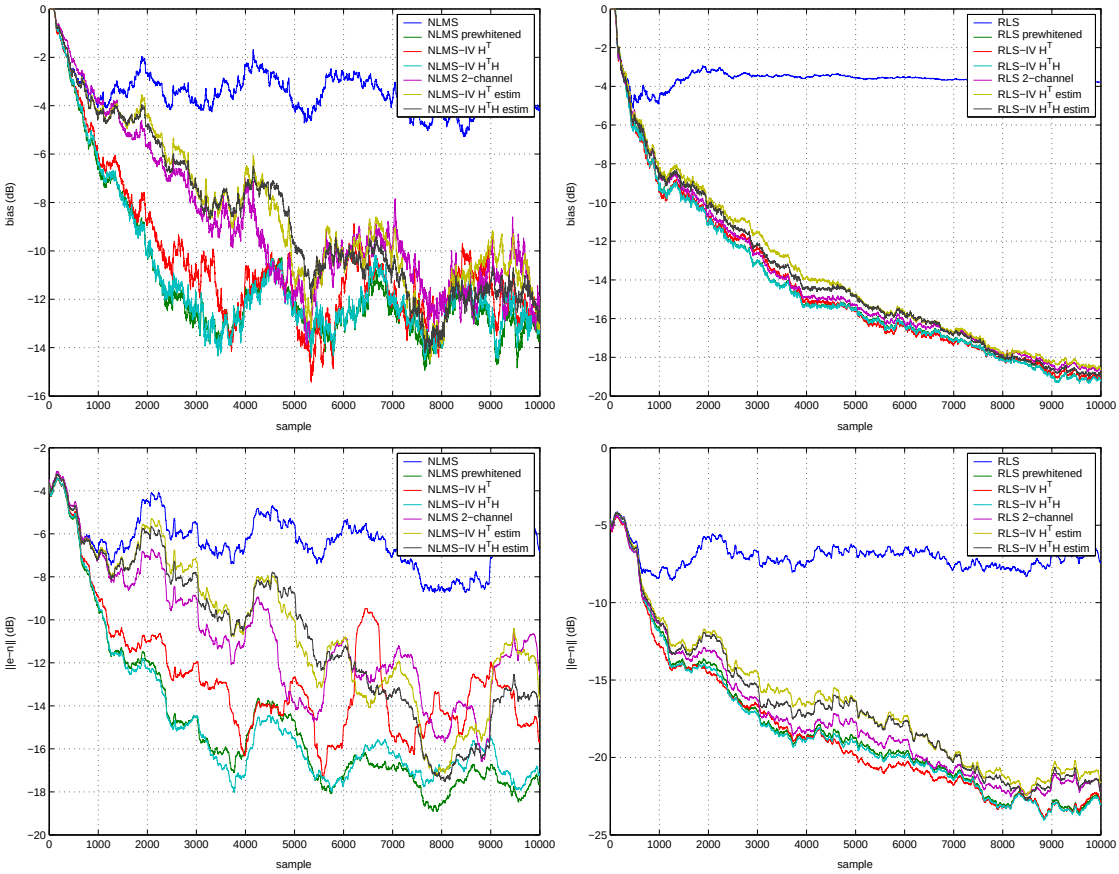

Figure 13: Typical on–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model on signal y.

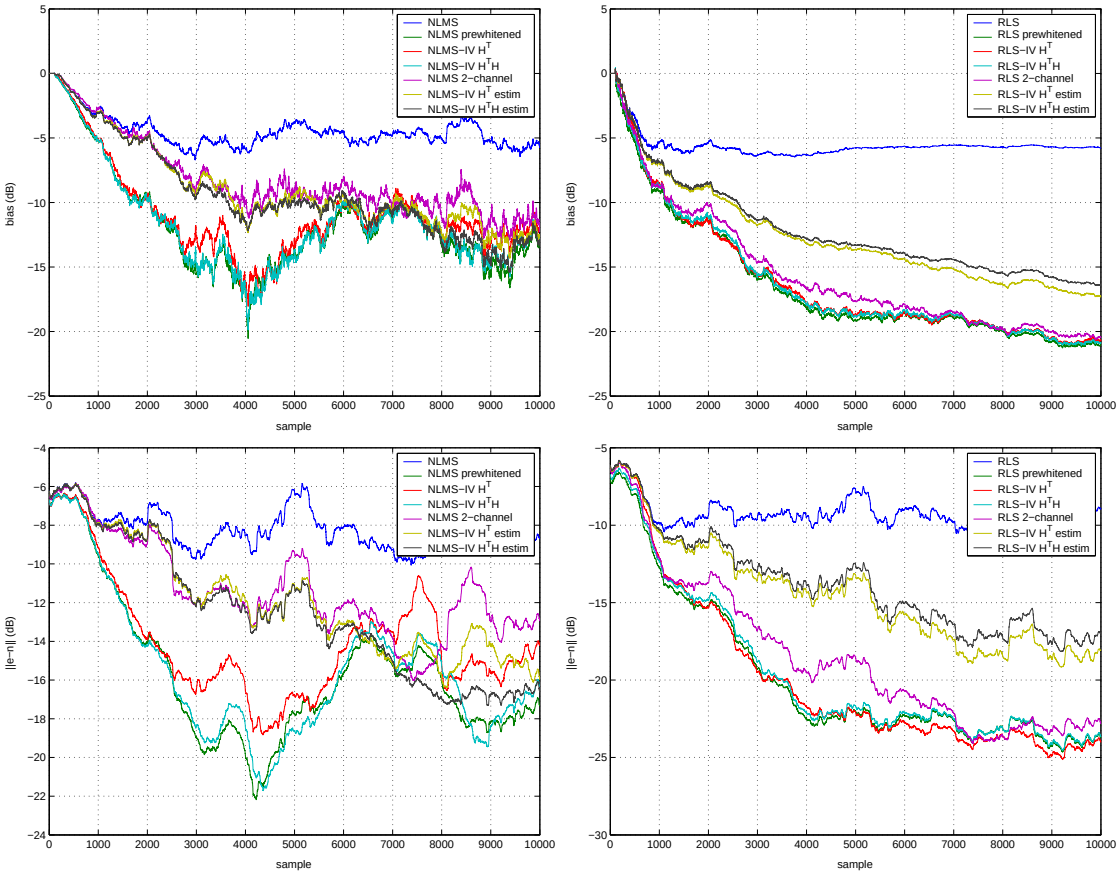

Figure 14: Typical on–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model on signal e.

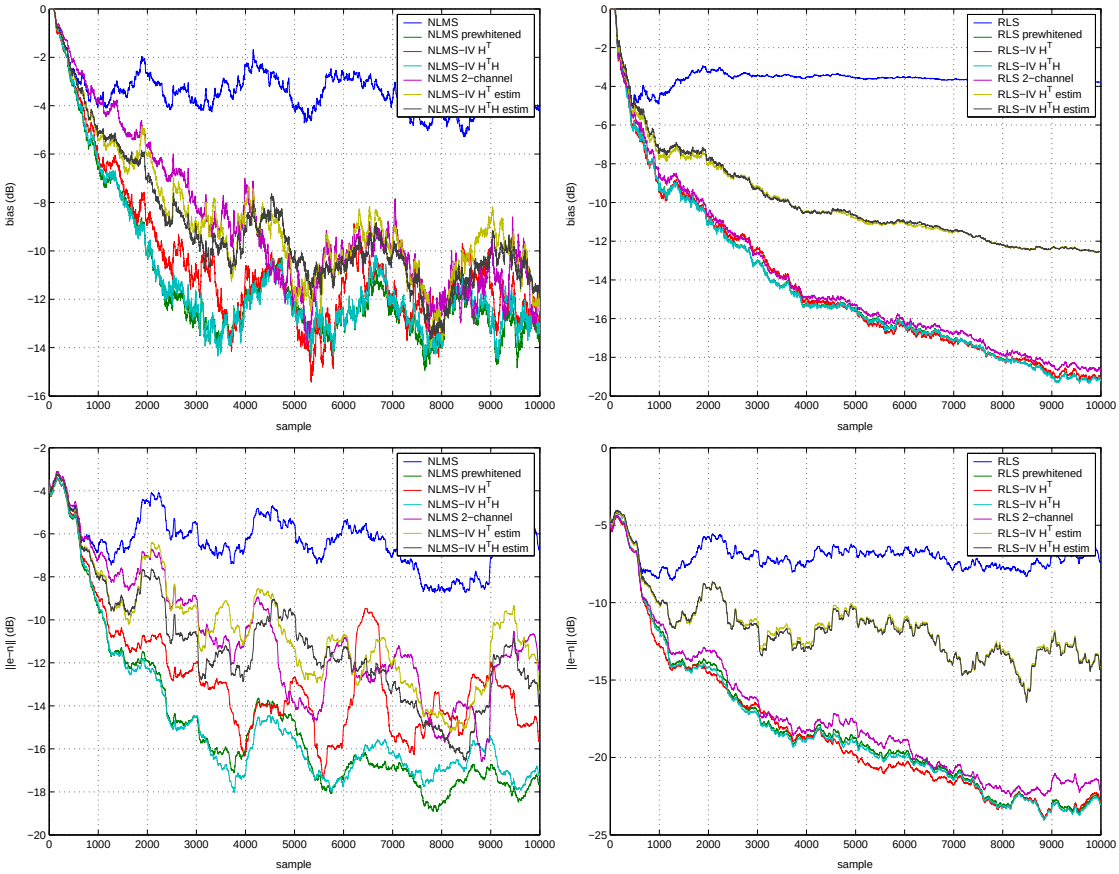

Figure 15: Typical on–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model on signal y.

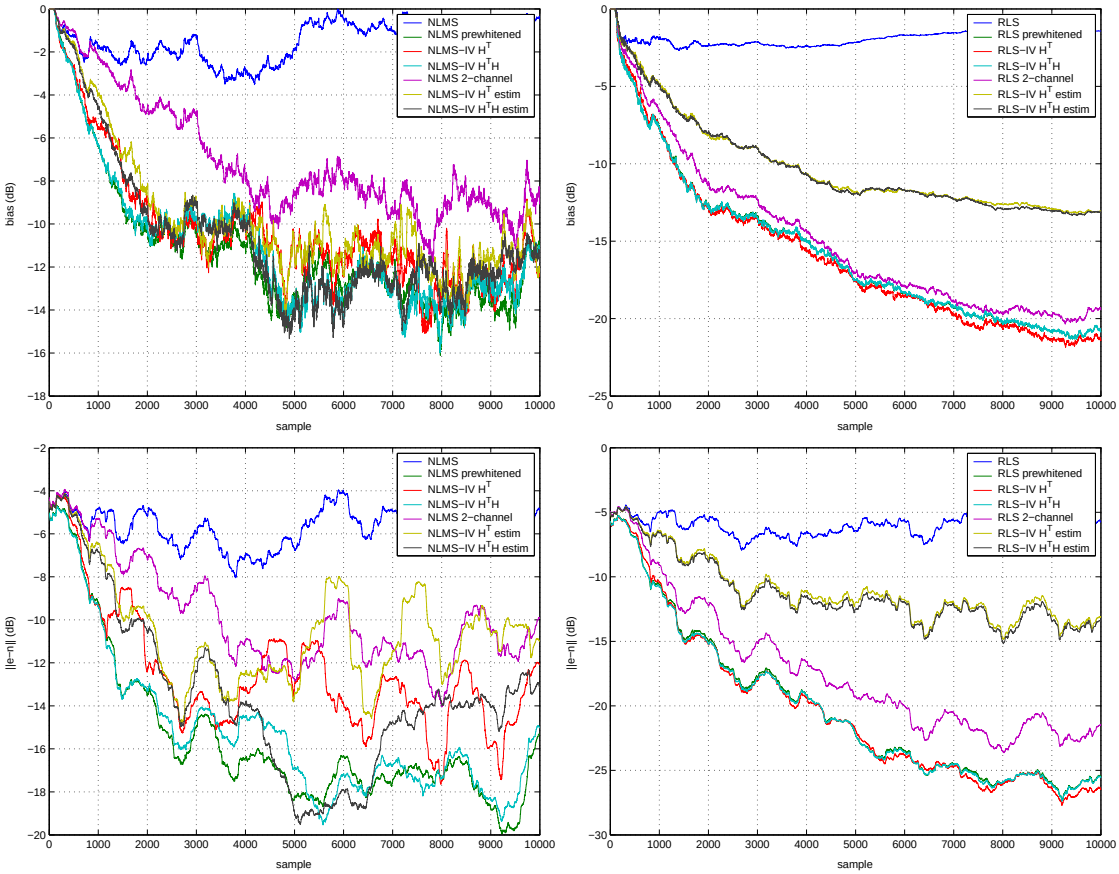

Figure 16: Typical on–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model on signal e.

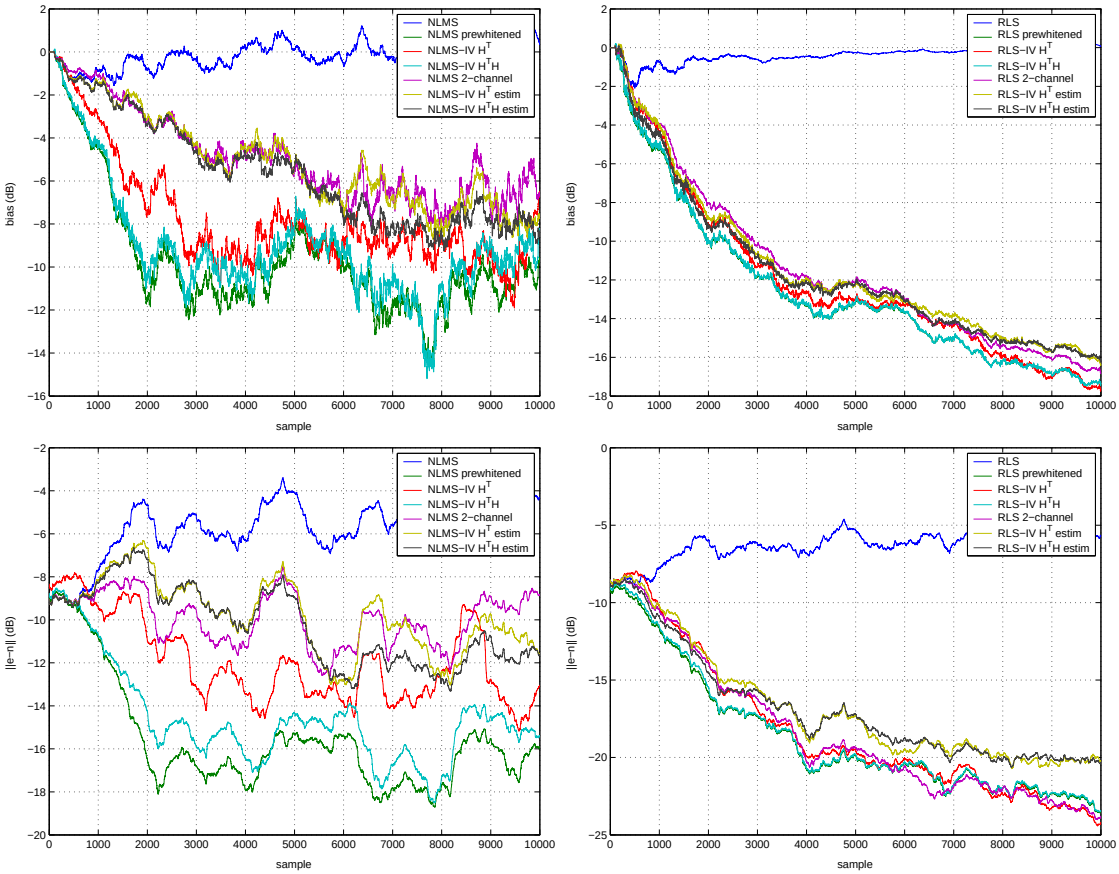

Figure 17: Another example of on–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model on signal y.

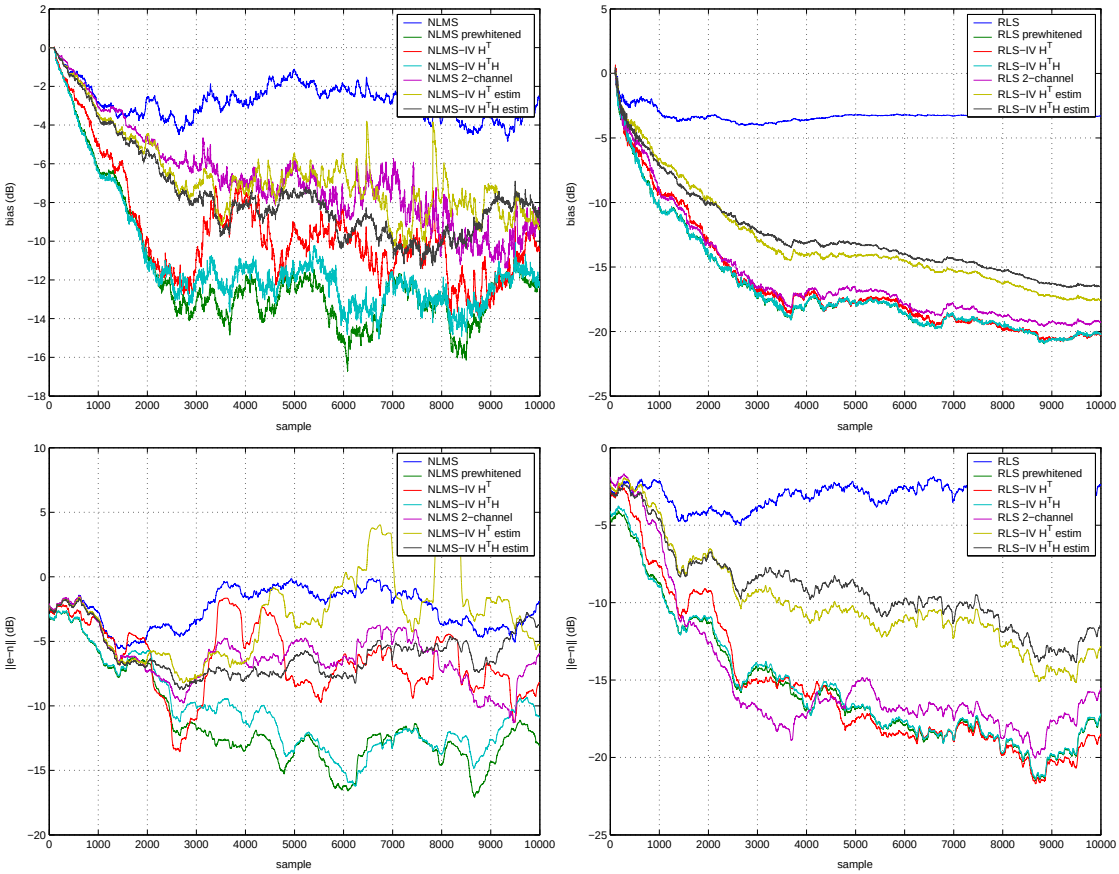

Figure 18: Another example of on–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model on signal e.

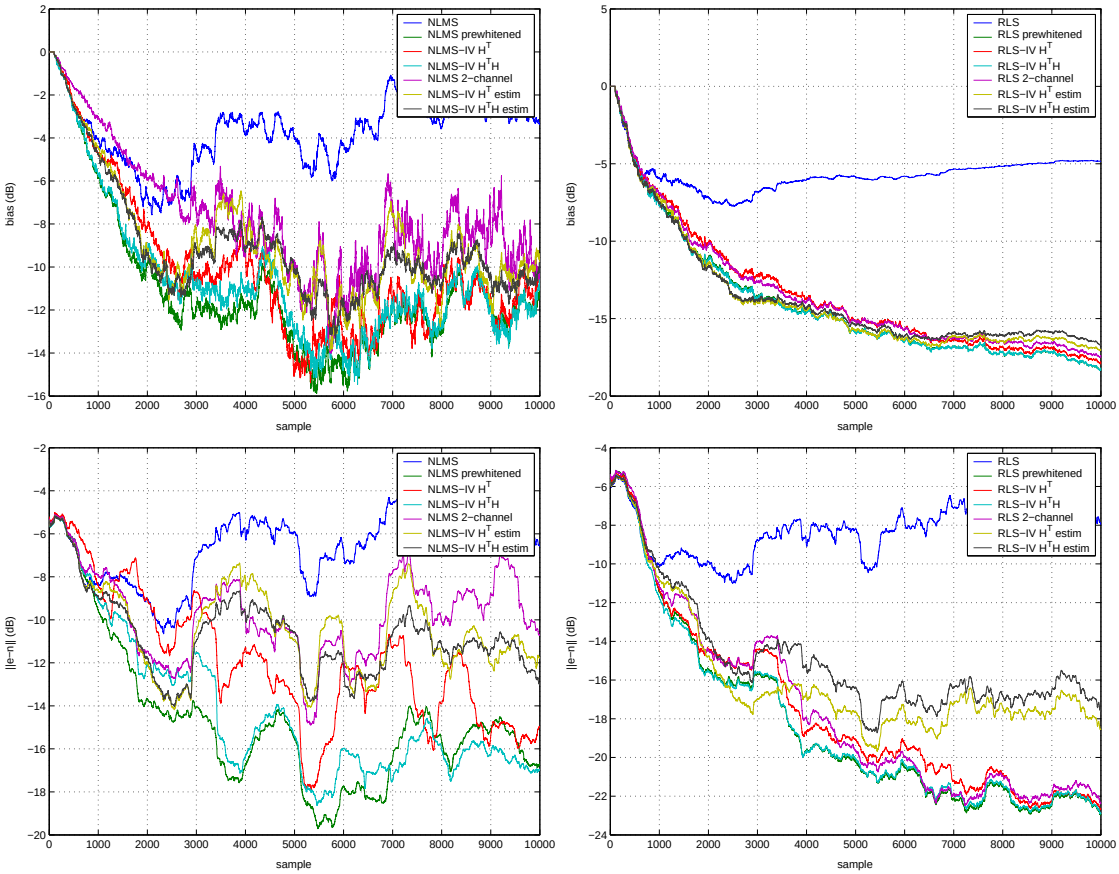

Figure 19: Yet another example of on–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model on signal y.

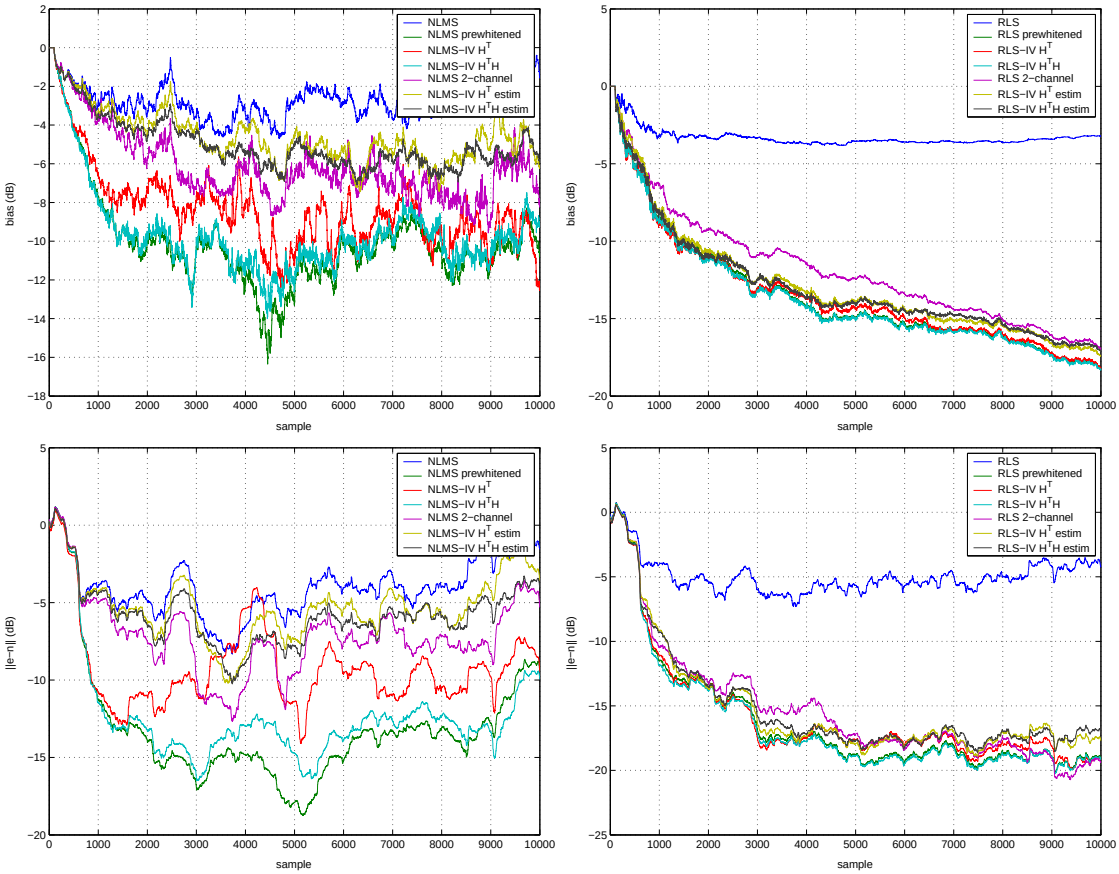

Figure 20: Yet another example of on–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model on signal y.

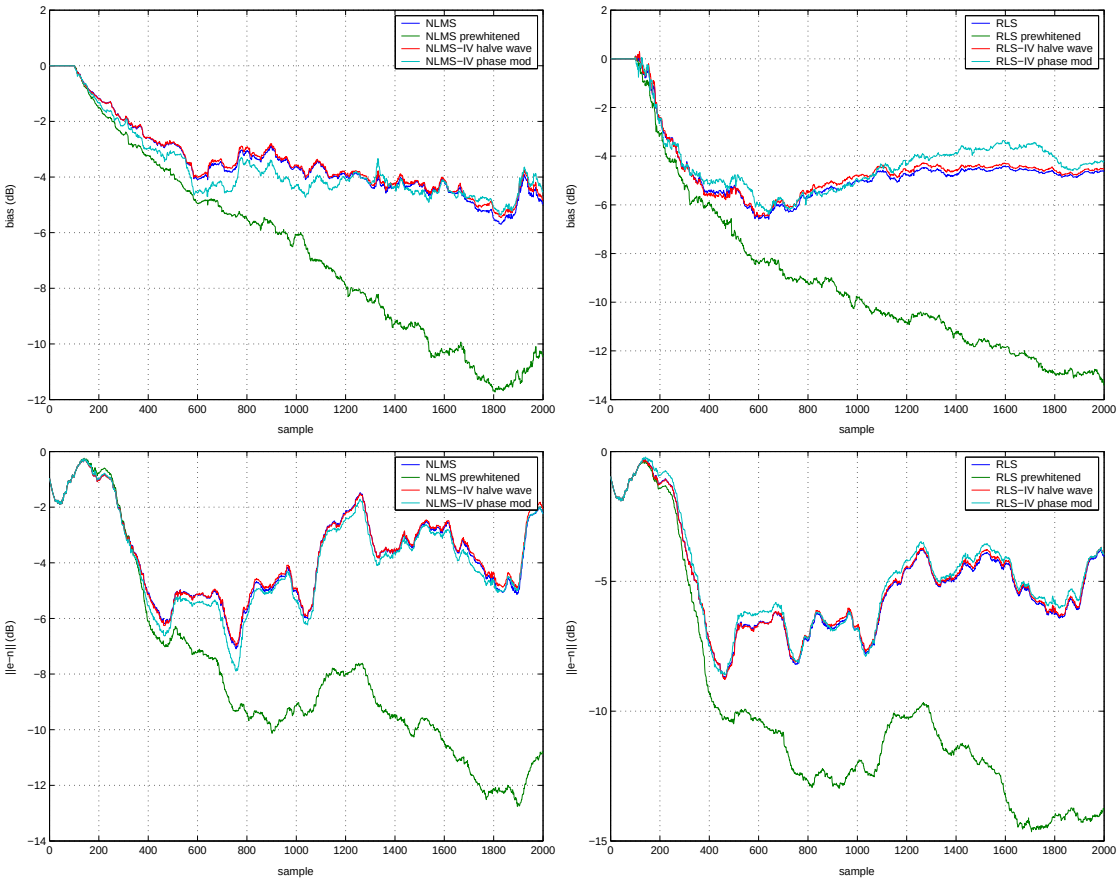

Figure 21: Typical off–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom.

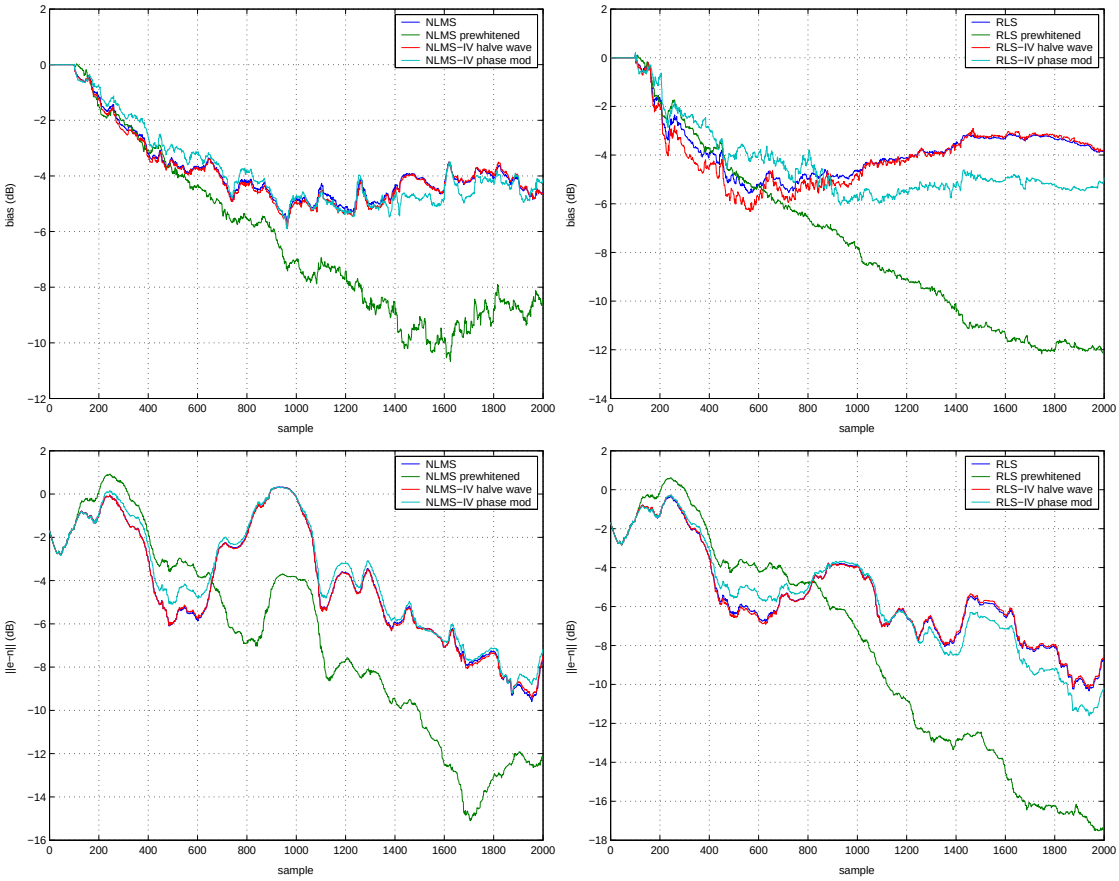

Figure 22: Another example of off–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom.

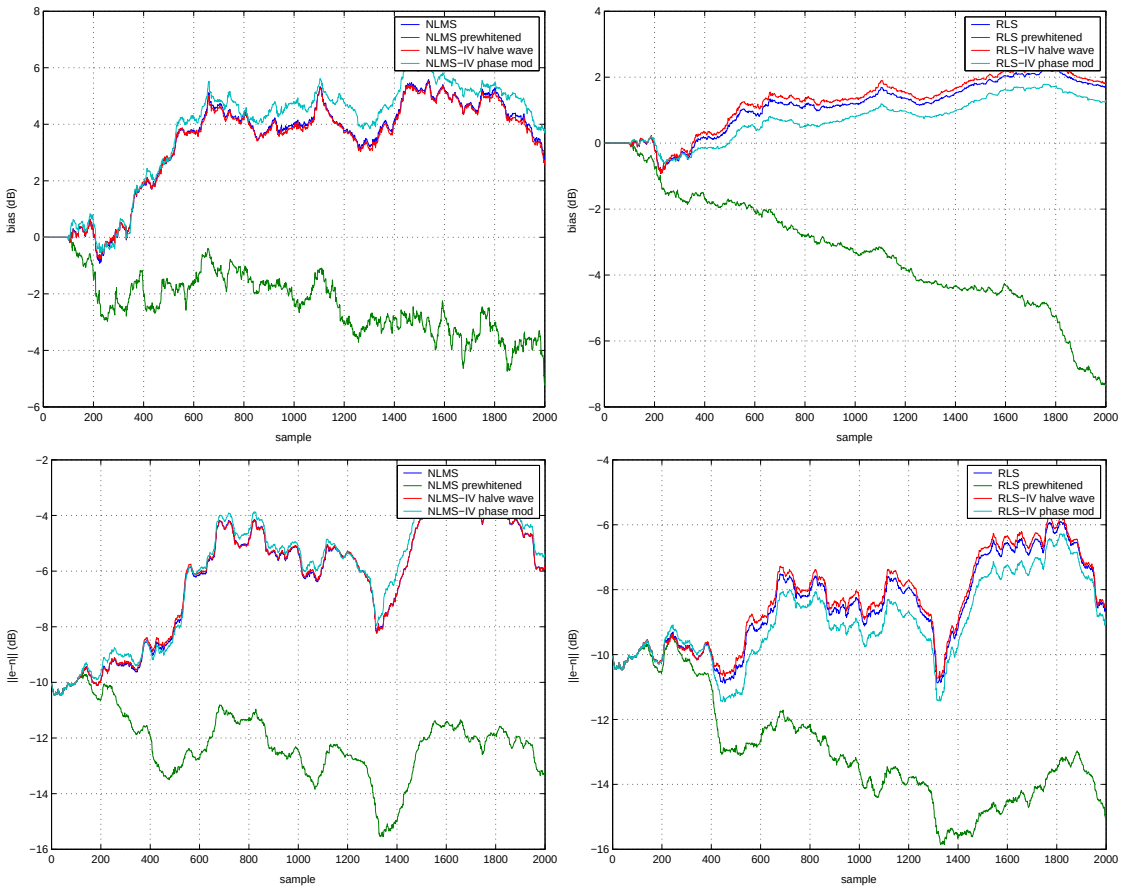

Figure 23: Typical on–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom.

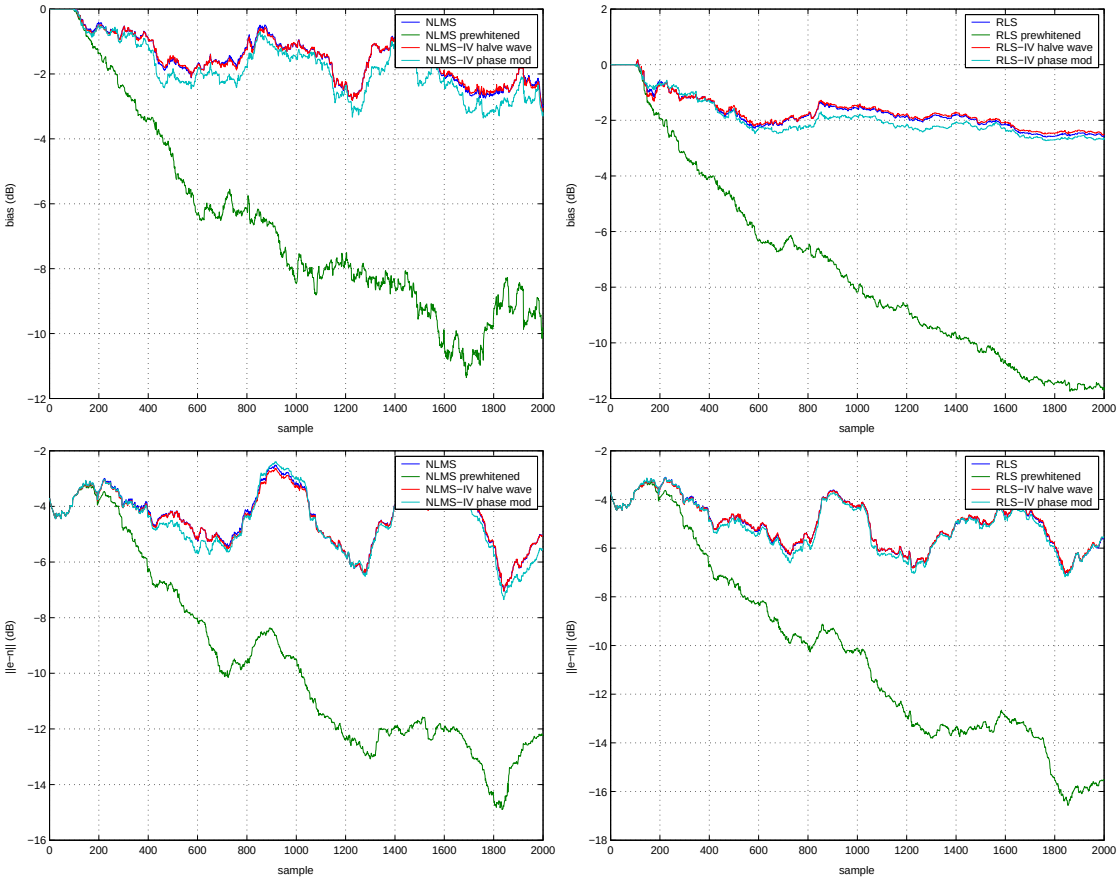

Figure 24: Another example of on–line performance on a stationary AR process : δ_1 is shown at the top, δ_2 is shown at the bottom.

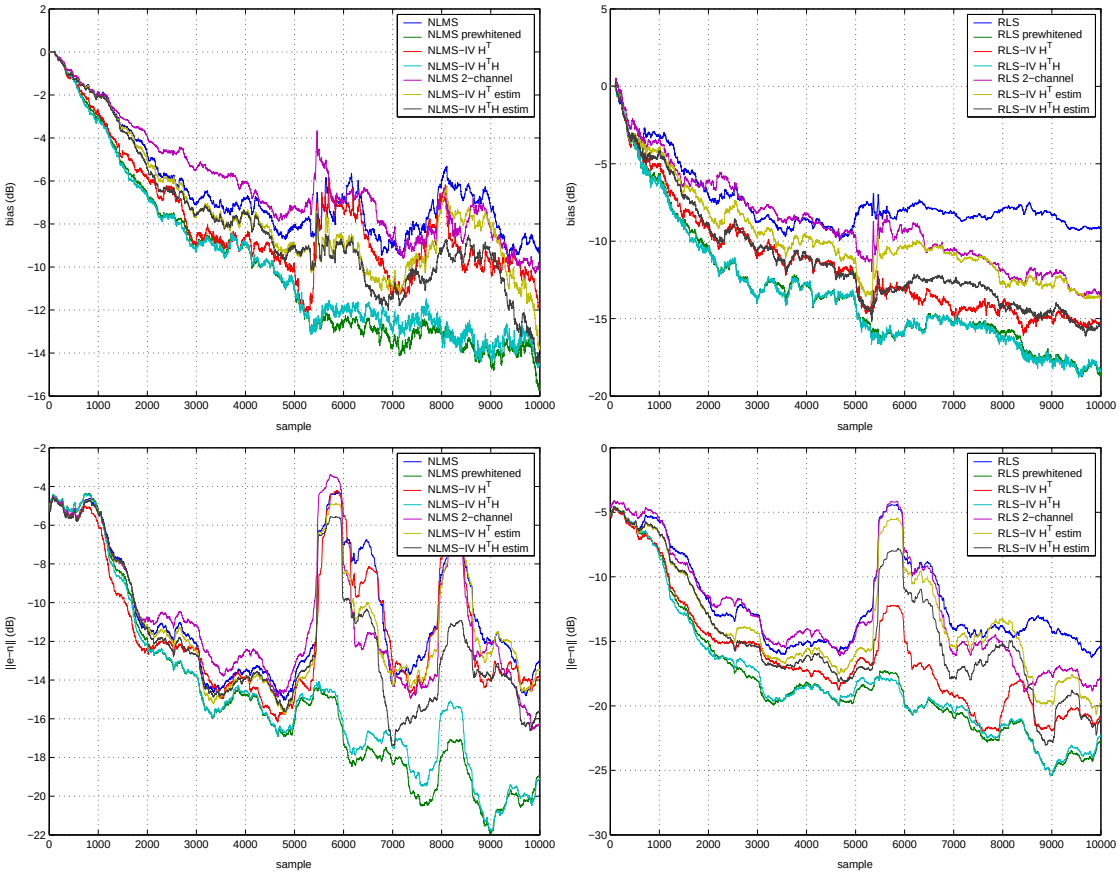

Figure 25: Typical off–line performance on a non–stationary AR process that is changed every 100 samples : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model.

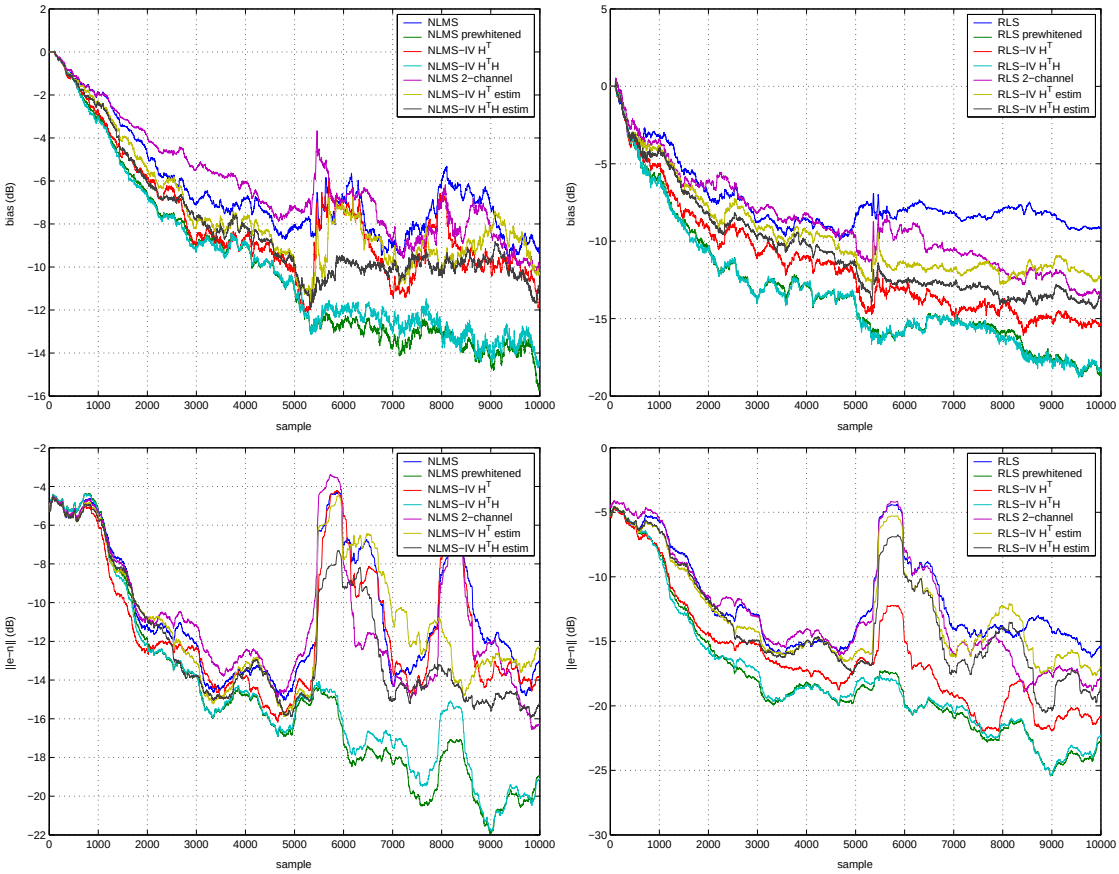

Figure 26: Typical off–line performance on a non–stationary AR process that is changed every 100 samples : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model.

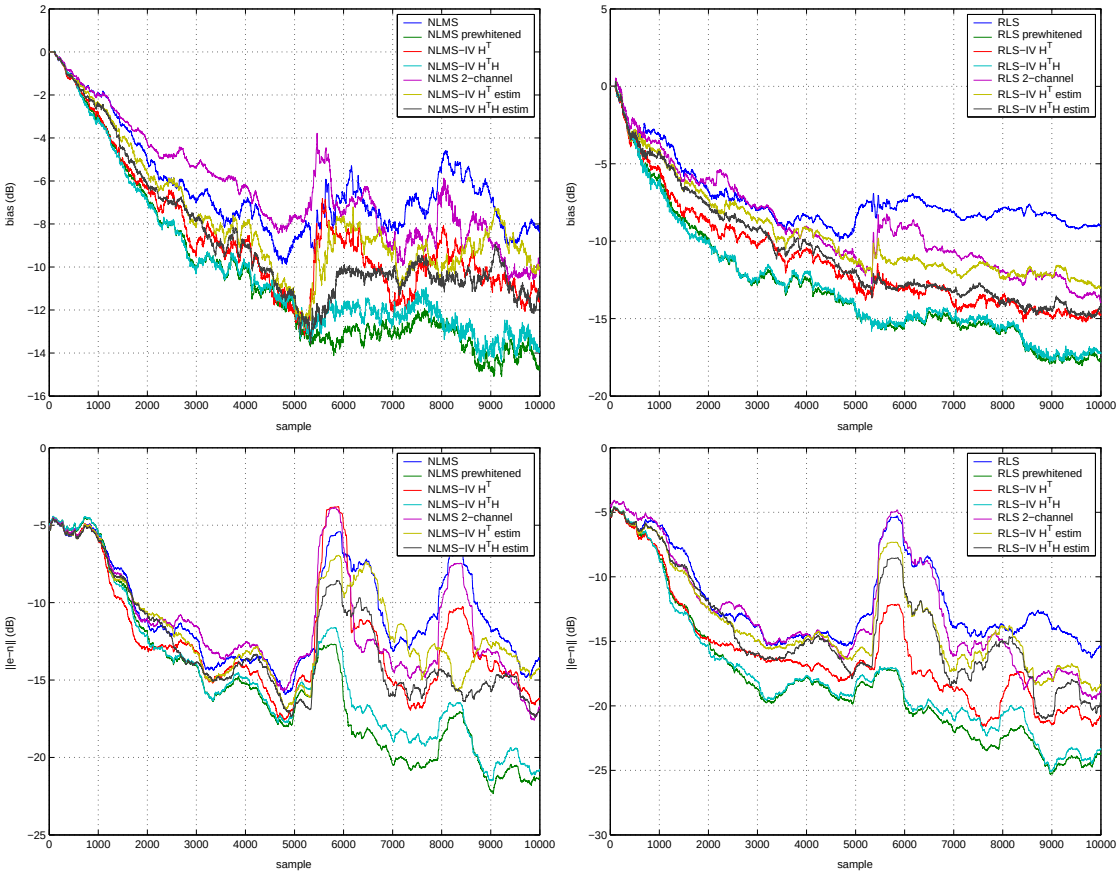

Figure 27: Typical on–line performance on a non–stationary AR process that is changed every 100 samples : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model on signal y.

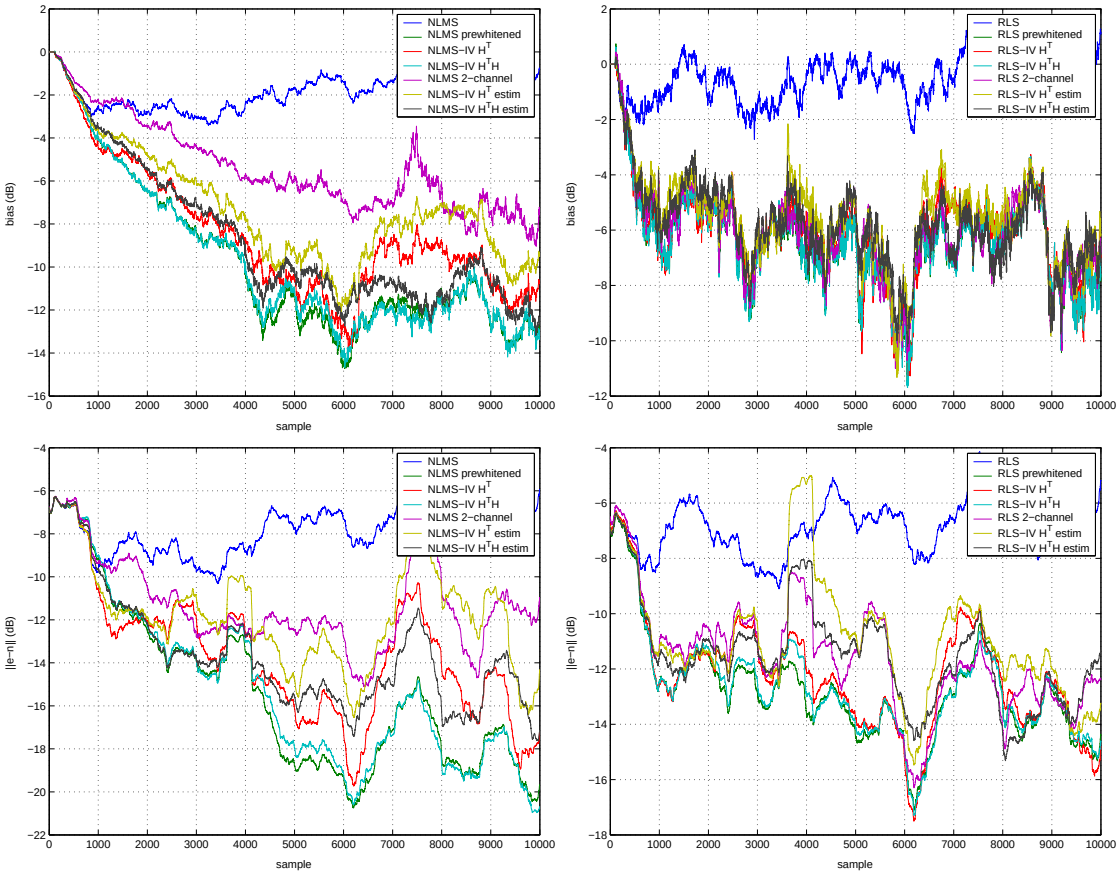

Figure 28: Typical on–line performance on a non–stationary AR process that is changed every 100 samples : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model on signal y.

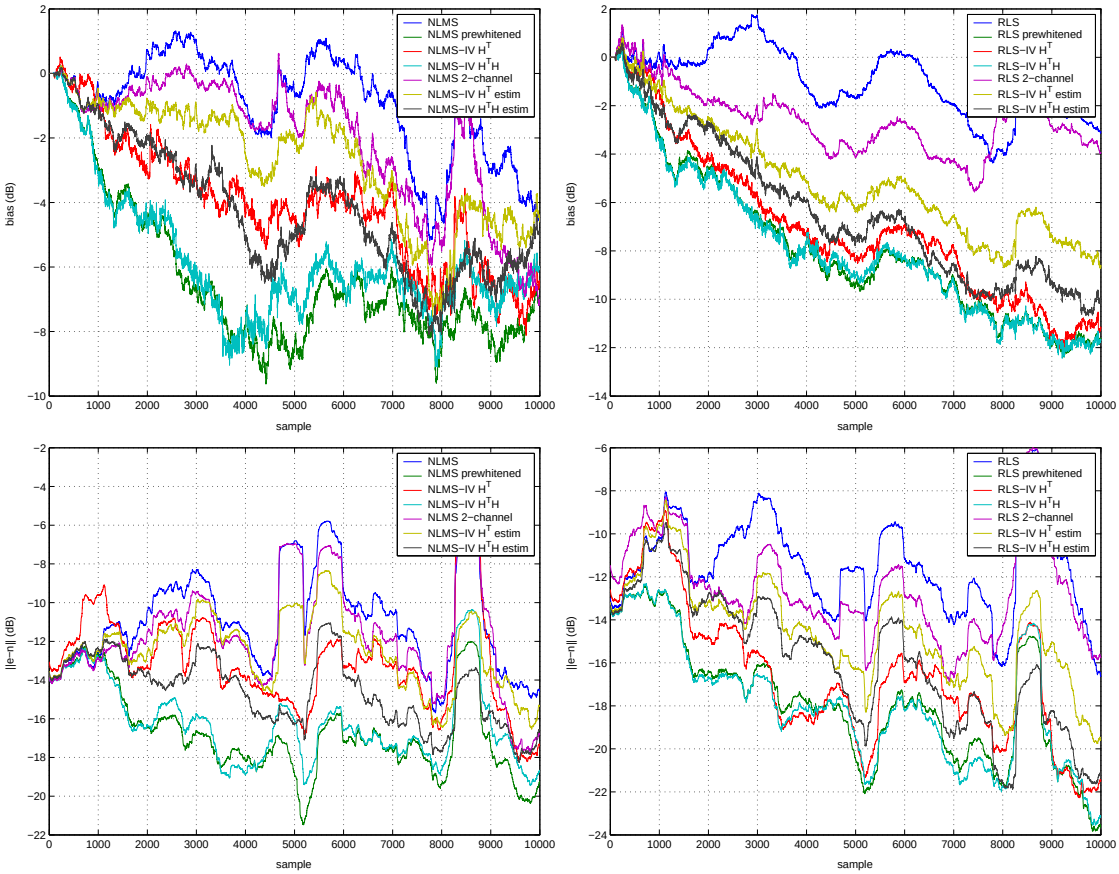

Figure 29: Other example of off–line performance on a non–stationary AR process that is changed every 100 samples : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model.

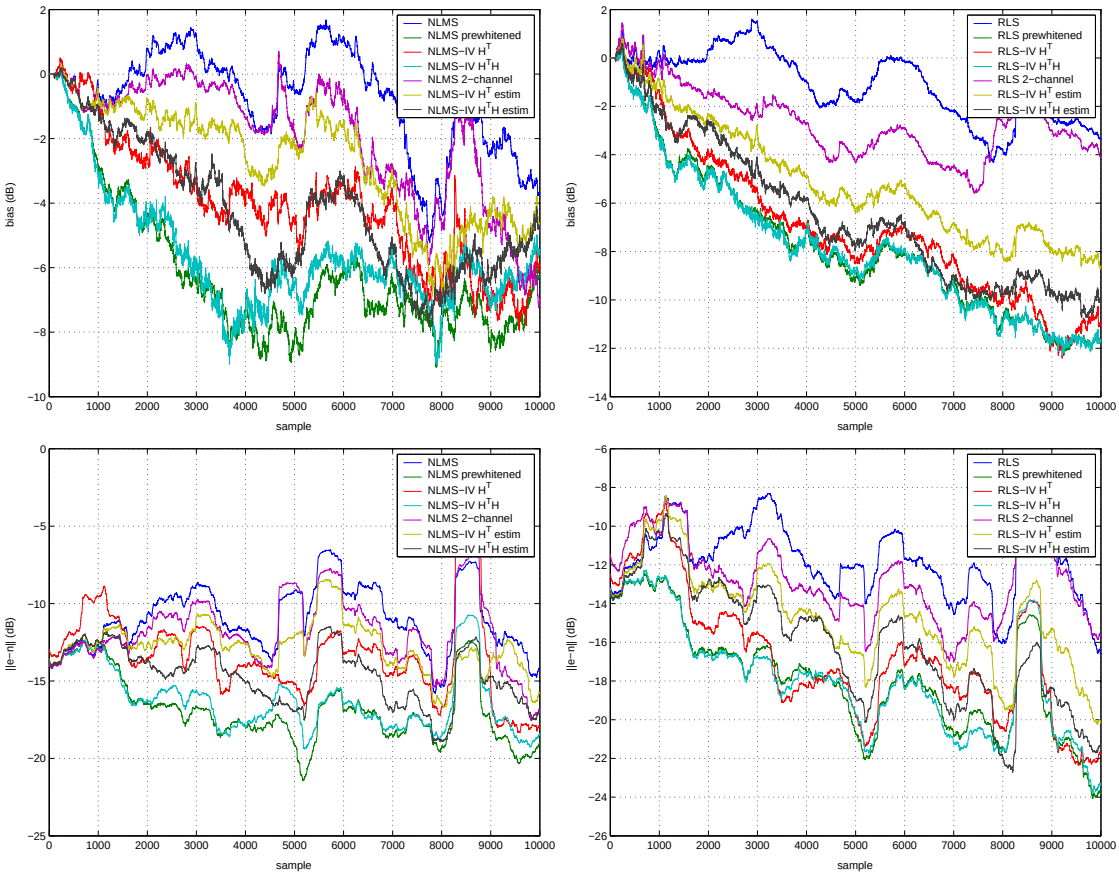

Figure 30: Other example of on–line performance on a non–stationary AR process that is changed every 100 samples : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model on signal y.

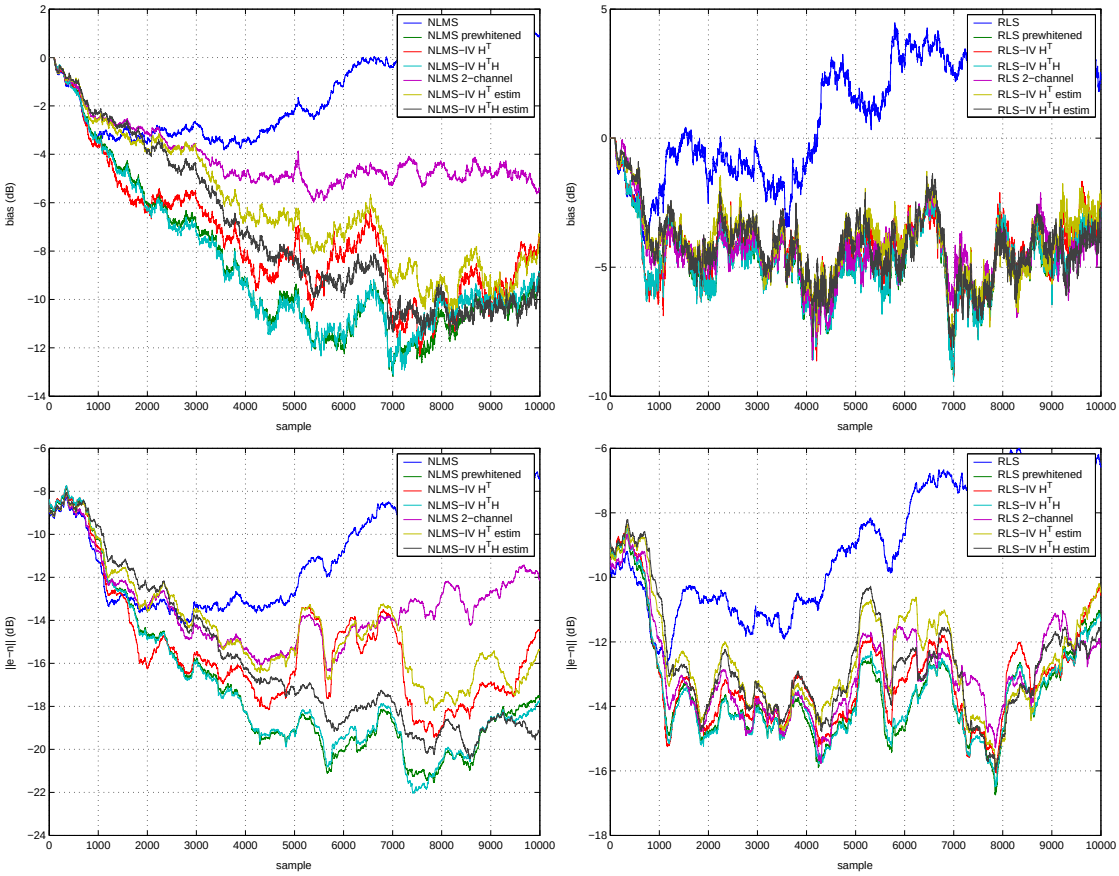

Figure 31: Other example of on–line performance on a non–stationary AR process that is changed every 100 samples : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model on signal e.

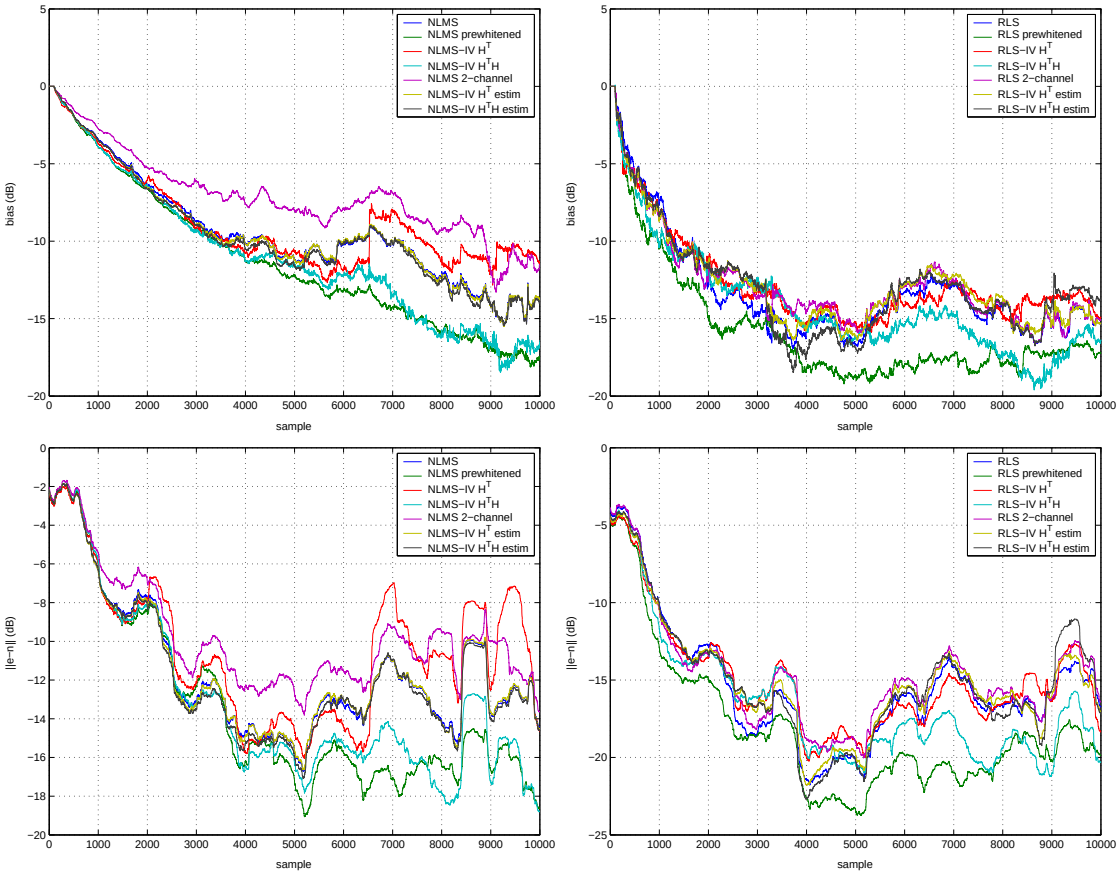

Figure 32: Typical off–line performance on a non–stationary AR process that is changed every 10 samples : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model.

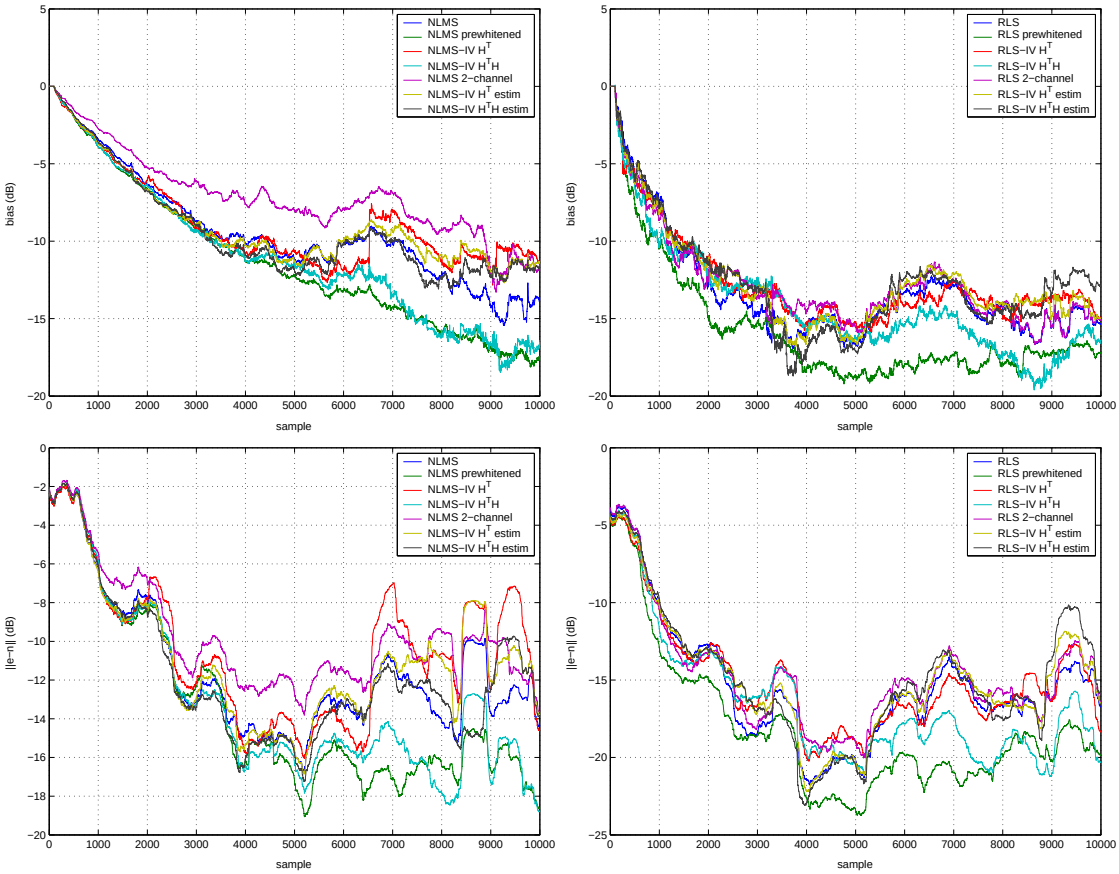

Figure 33: Typical off–line performance on a non–stationary AR process that is changed every 10 samples : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model.

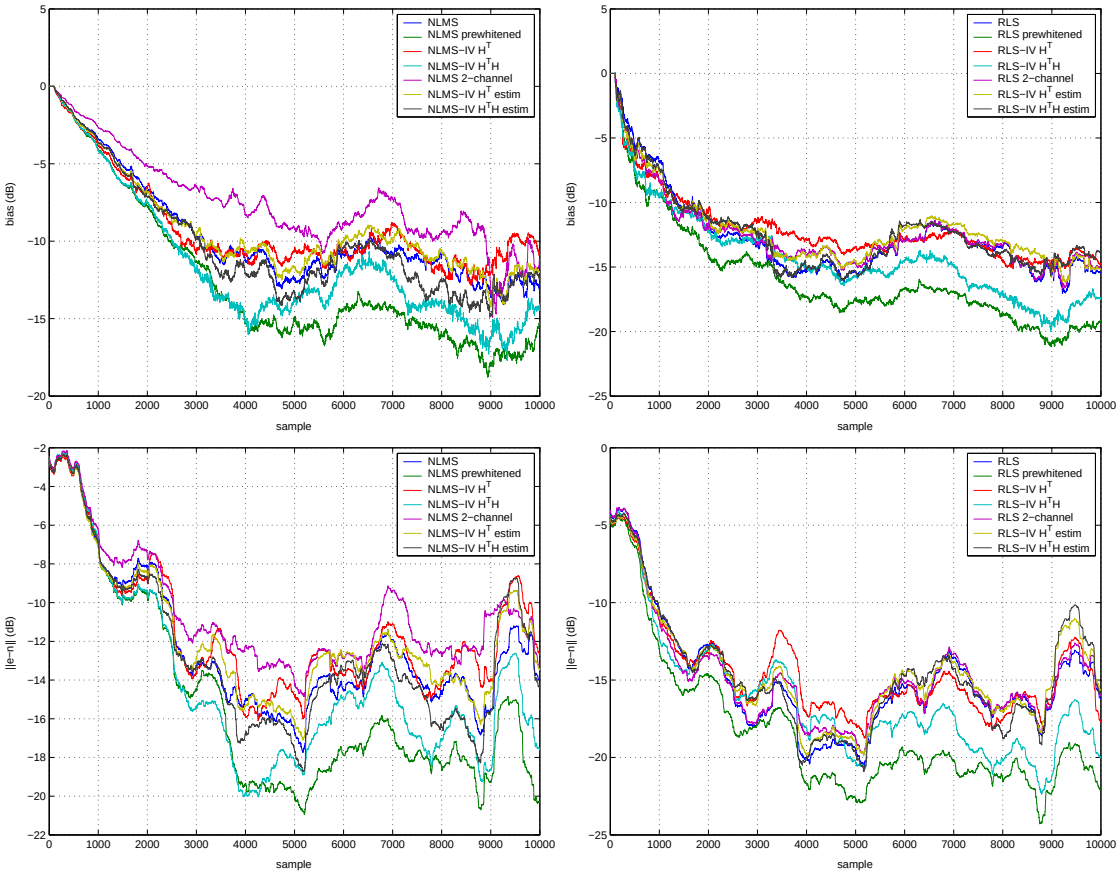

Figure 34: Typical on–line performance on a non–stationary AR process that is changed every 10 samples : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model on signal y.

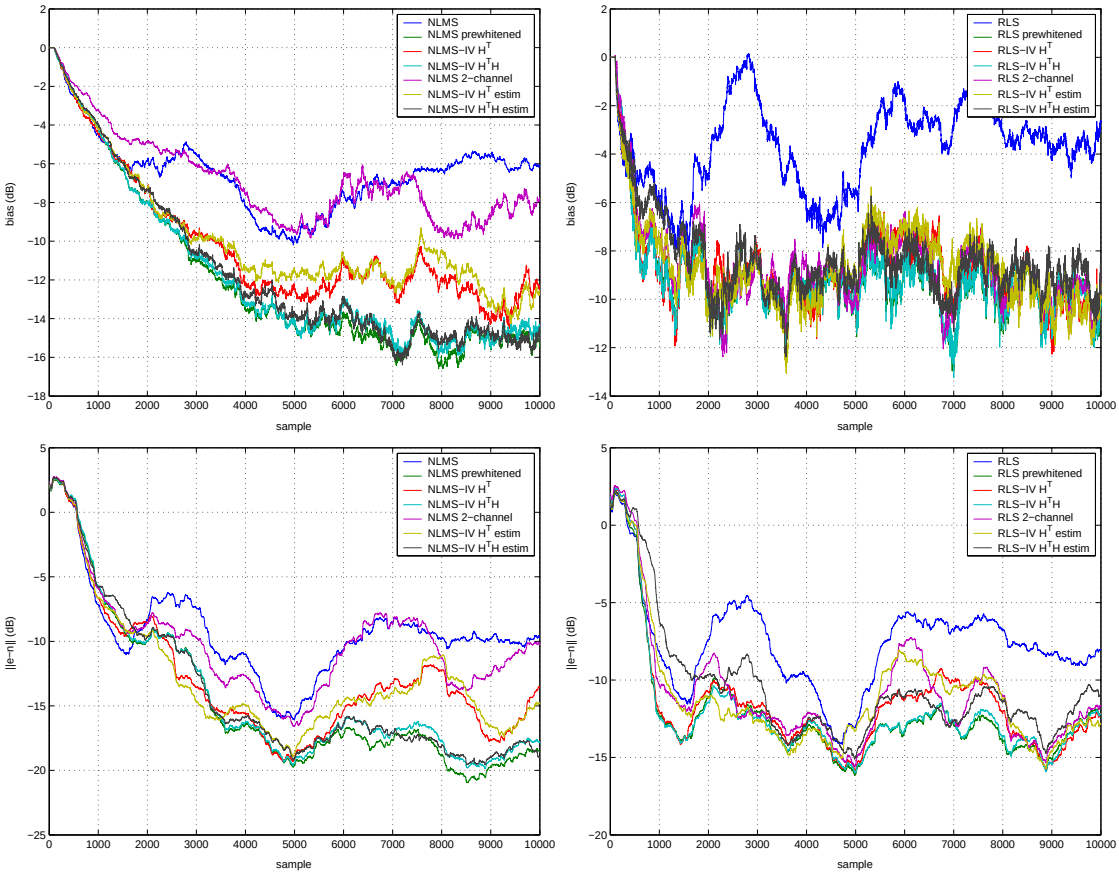

Figure 35: Typical on–line performance on a non–stationary AR process that is changed every 10 samples : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model on signal e.

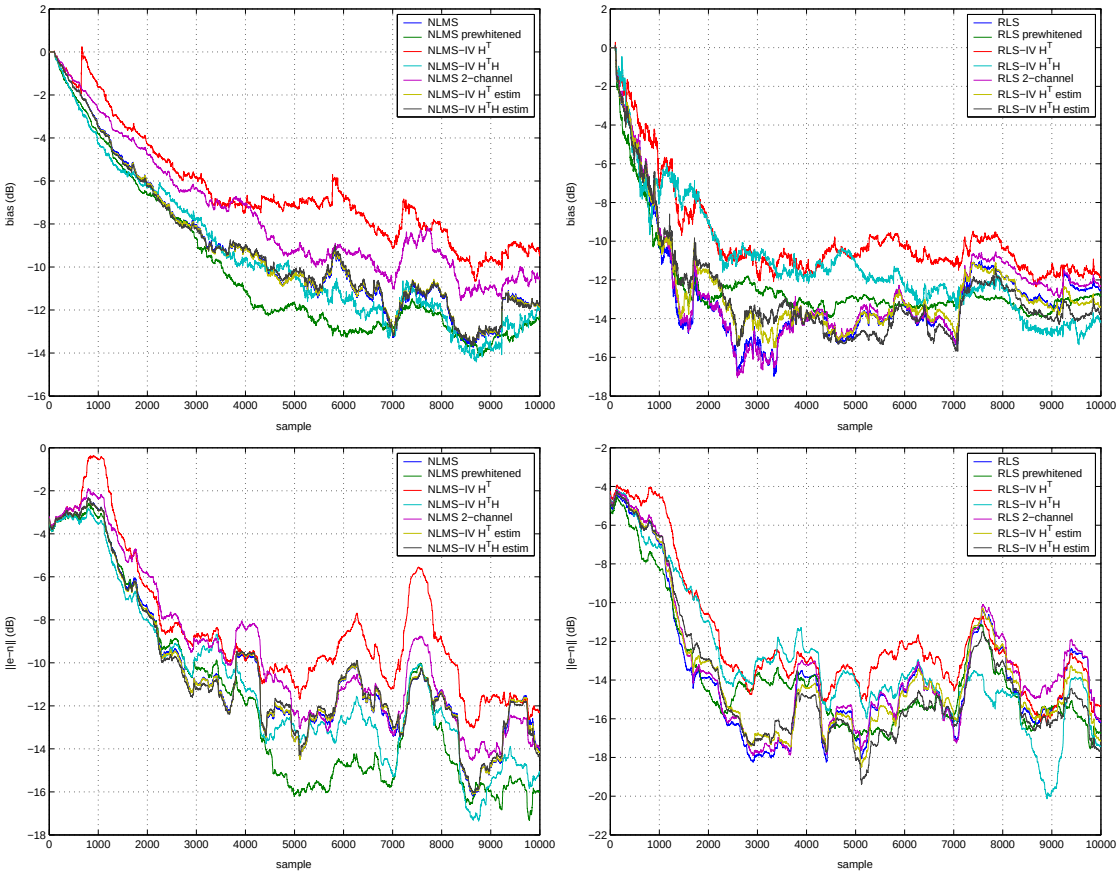

Figure 36: Typical off–line performance on a non–stationary AR process that is changed every sample : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^TH) is done based on a long (48–th order) model.

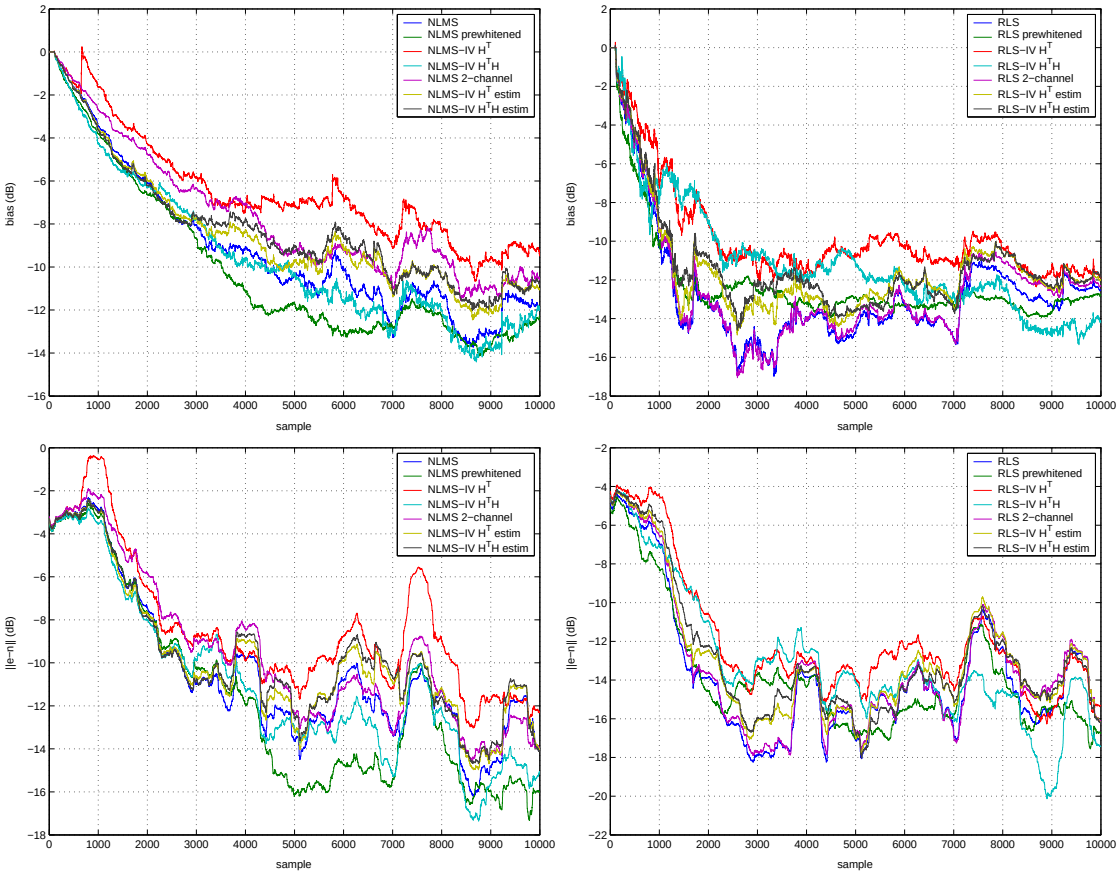

Figure 37: Typical off–line performance on a non–stationary AR process that is changed every sample : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model.

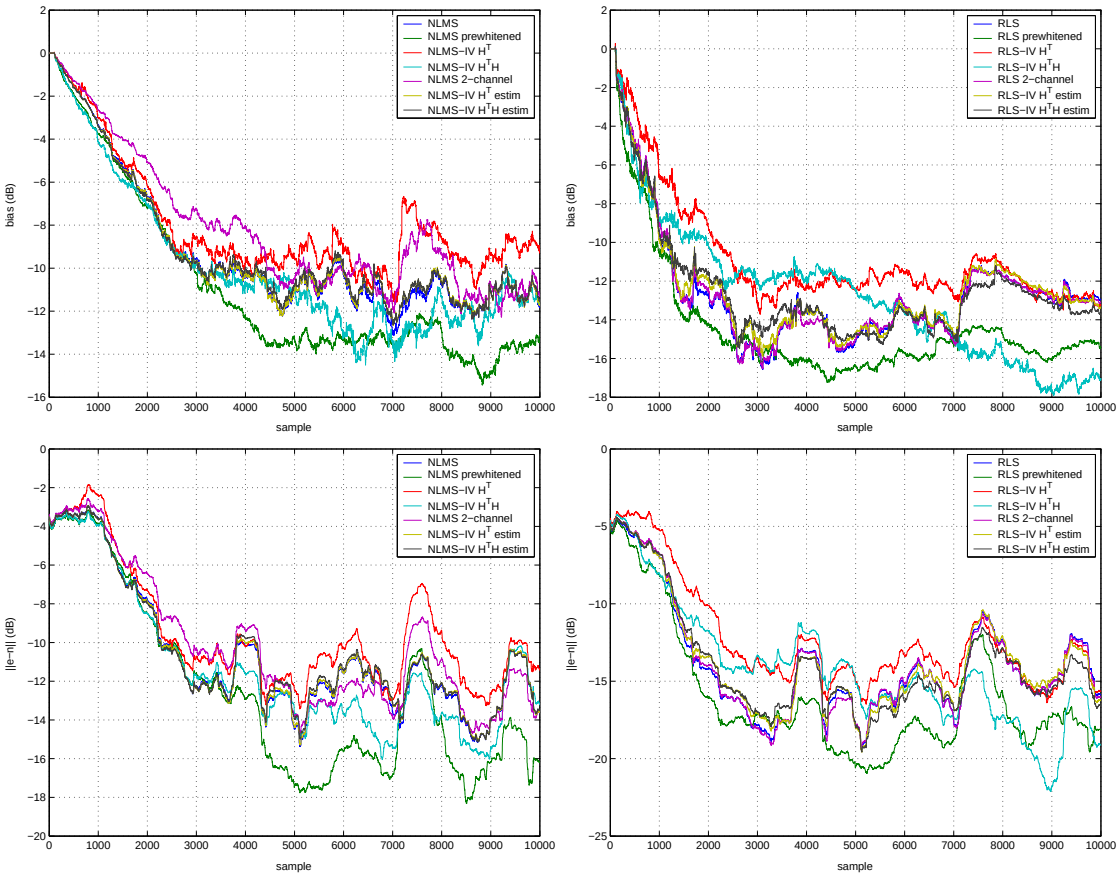

Figure 38: Typical on–line performance on a non–stationary AR process that is changed every sample : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model on signal y.

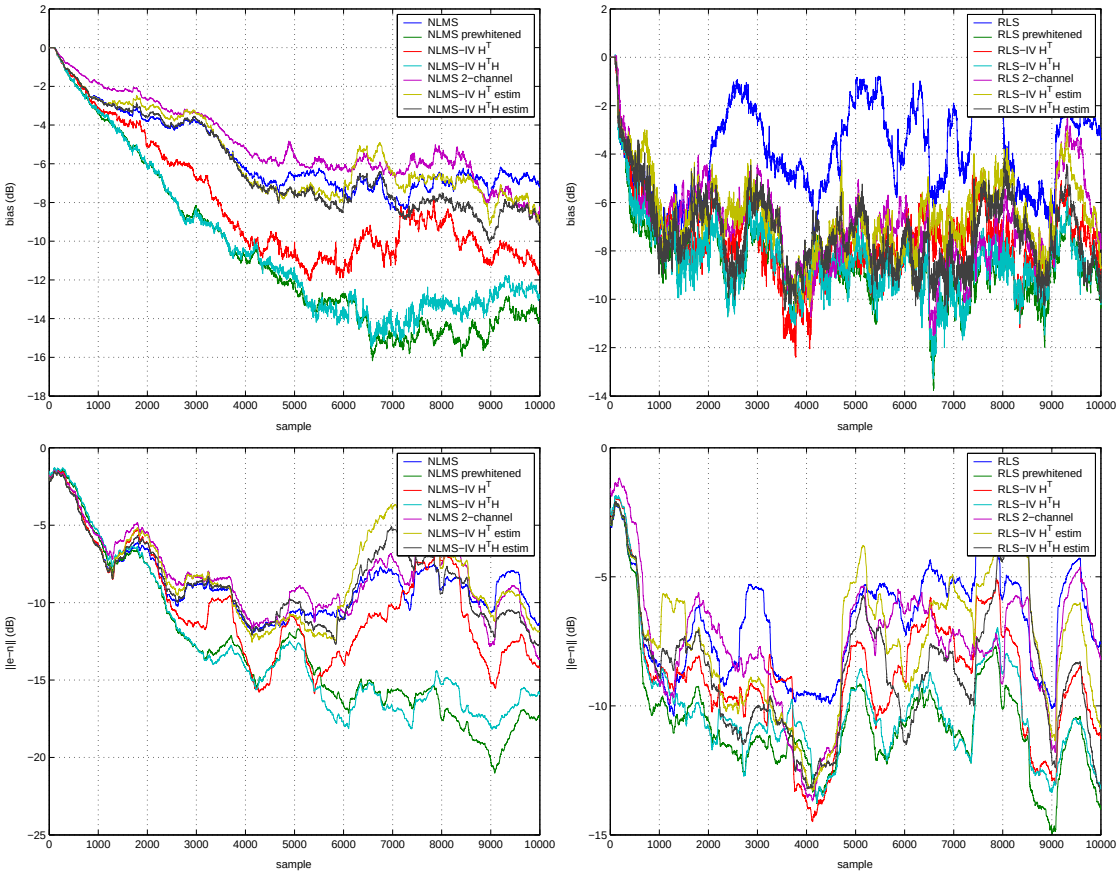

Figure 39: Typical on–line performance on a non–stationary AR process that is changed every sample : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a long (48–th order) model on signal e.

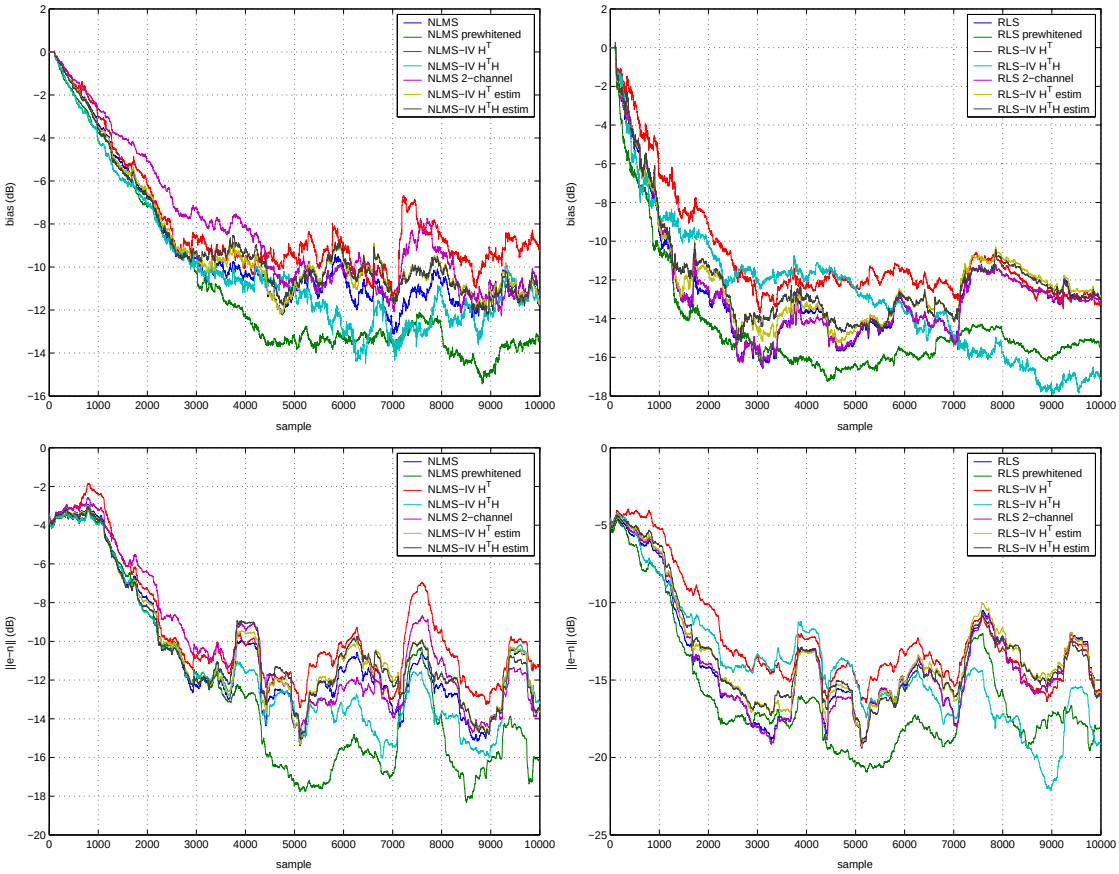

Figure 40: Typical on–line performance on a non–stationary AR process that is changed every sample : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–$\mathrm{H}^T\mathrm{H}$) is done based on a short (9–th order) model on signal y.

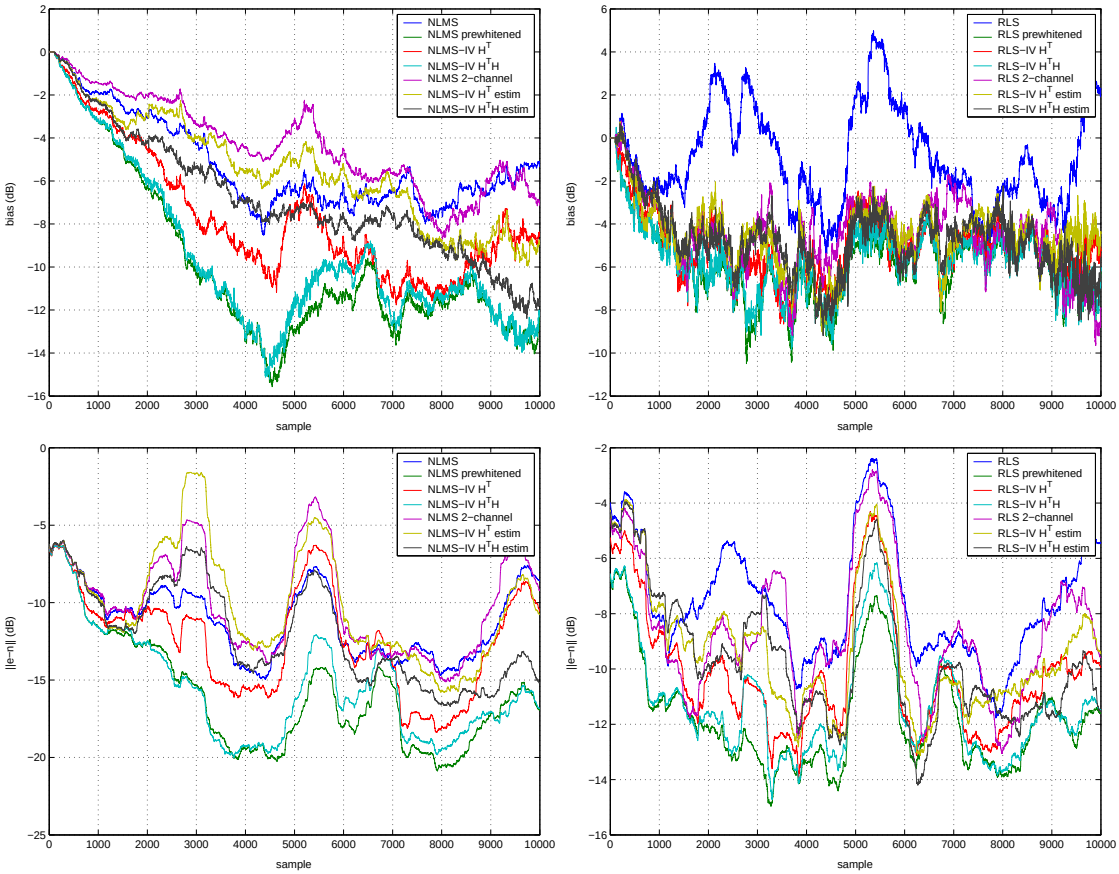

Figure 41: Typical on–line performance on a non–stationary AR process that is changed every sample : δ_1 is shown at the top, δ_2 is shown at the bottom. The updating of the AR model (in the case of IV–H^T and IV–H^{T}H) is done based on a short (9–th order) model on signal e.