Tracking Ill-Conditioning for the RLS-Lattice Algorithms

James R. Bunch*
Department of Mathematics
University of California, San Diego
La Jolla, CA 92093-0112

Richard C. Le Borne†
Department of Mathematics
University of Tennessee at Chattanooga
Chattanooga, TN 37403-2598

Ian K. Proudler
DERA, Malvern
Worcestershire
WR14 2NJ, UK

Abstract

The numerical performance of lattice-based adaptive signal processing algorithms is shown to involve the conditioning of a 2×2 matrix whose off-diagonal elements contain reflection coefficients. Degraded algorithmic performance for the a posteriori Recursive Least Squares Lattice (RLSL) is shown to be attributed to the ill-conditioning of this matrix. Theoretical results are given which may be used to separate the conditioning of the underlying problem from issues concerning algorithmic stability. Although our results are not restricted to the all-pole case, for simplicity we make use of this well-understood example since the condition number of the autocorrelation matrix will become arbitrarily close to singularity as the poles of an all-pole filter approach the unit circle. For a second-order prediction problem, four case studies of varying conditioning are provided which demonstrate the appropriateness of the theoretical bounds which analytically describe the sensitivity to perturbations in the residual update recursions. In doing so, this paper illustrates the use of numerical linear algebra analysis techniques to better understand the numerical performance of algorithms in signal processing as well as emphasizing that numerical performance is a function of the problem's conditioning as well as algorithmic stability.

1 Introduction

The class of fast Recursive Least Squares Lattice (RLSL) algorithms remains an active topic in research in the area of adaptive filtering as well as in the application of such algorithms. Lattice algorithms are attractive since their computational complexity increases linearly with respect to the updated parameters. This property is due primarily to the manner in which the algorithm updates in order, m, as well as in time, n; information from a previous order update can be carried over to the subsequent order update. Of the many favorable attributes enjoyed by the class of RLSL

*Partial support was provided by NSF grant MIP-9625482.
†The work of this author was supported by NSF-NATO grant GER-9552624.

algorithms, its fast rate of convergence and its insensitivity to the spectral radius of the underlying correlation matrix of the input data stand out. Related to this, but of a more numerical consequence, is the conditioning of the autocorrelation matrix. These algorithms do not compute this matrix explicitly, but the negative impact from working in finite precision with an ill-conditioned matrix is ever-present. In this paper we will show how the condition number of a 2×2 matrix whose elements depend on the algorithm's forward and backward reflection coefficients, $\Gamma_{f,m}(n)$ and $\Gamma_{b,m}(n)$, can be used to sense the underlying conditioning of the problem being solved.

These algorithms are expected to operate under stationary and nonstationary conditions in which the underlying stochastic nature of the problem is unknown. It would be useful, then, for the algorithm to efficiently estimate the conditioning of the underlying problem and, if deemed appropriate, warn the user. Furthermore, much research on the effects of finite precision arithmetic has been performed regarding the performance of lattice algorithms ([1], [5], [7], [9], [10]). It has remained unclear, however, how the conditioning affects algorithm performance: if an algorithm is shown to demonstrate poor numerical performance characteristics, is it the specific problem that is to blame, or is it because the algorithm is unstable? We show that for lattice-based algorithms, as in numerical linear algebra, conditioning is independent of the algorithm; the conditioning of the underlying system will demonstrate that regardless of the algorithm used, poor performance can result from processing implicit matrices which are ill-conditioned. From these results it can be concluded that a lattice-based algorithm, shown through simulations to exhibit poor numerical properties, may, in fact, be a good algorithm if the underlying conditioning was the cause of the numerical difficulties.

2 The Conditioning of the Residual Update

The advantages of lattice algorithms are well recognized [3, pp.228-229] and are distinguished from transversal filters through a transformation which pre-processes (decomposes) the input sequence, $u(n)$, into an equivalent but uncorrelated process of backward residuals, $b_m(n), 0 \leq m \leq M$, $0 \leq n \leq T$. Here M and T represent, respectively, the number of filter stages and the total number of time-update iterations that are performed during the execution of the filter algorithm. For this decomposition to be successful, the accurate computation of the backward residuals is required. This in turn requires accurate computations of the parameters which define their update recursion, namely the forward residual, $f_m(n)$, the backward reflection coefficient, $\Gamma_{b,m}(n)$, and all parameters on which they depend. Subsequent to this decomposition of the input data, the backward residuals are then linearly combined (the joint process) to produce the filter estimation. Below we give the update recursions for $f_m(n)$ and $b_m(n)$, respectively:

$$f_m(n) = f_{m-1}(n) + \Gamma^*_{f,m}(n) b_{m-1}(n-1) \tag{1}$$

$$b_m(n) = b_{m-1}(n-1) + \Gamma^*_{b,m}(n) f_{m-1}(n). \tag{2}$$

It is paramount that these updates be performed accurately at each update. In [1], Bunch and Le Borne performed a first order analysis to investigate the manner in which arithmetic effects contribute to inaccuracies in the order and time update recursions of the a posteriori RLSL algorithm. Additionally, in [6], [7], and [8], Ling, Manolakis, and Proakis introduce the direct updating scheme for the a priori and a posteriori RLSL which was shown through their experiments to be more robust to the numerical effects of finite precision when compared to its counterpart algorithm which they termed the indirect updating RLSL algorithm. Recently, in [5], Le Borne has given the theoretical rationale which justifies the robust behavior of the direct RLSL algorithm over the indirect RLSL algorithm. Despite these differences in algorithm performance, neither algorithm can be expected to perform well if any of the update recursions become sensitive to small perturbations; that is, if one update can dramatically change when any of the parameters from which it depends are slightly perturbed. When this is true, the impetus for algorithm divergence becomes a statistical probability. It is thus reasonable, [11] [12], to expect the likelihood for a nonminimal algorithm to suffer from finite arithmetic effects to increase if it is known that one or more of its update equations have become sensitive to small perturbations. The question of defining, and then detecting, this circumstance is the goal of this paper. By considering the forward and backward residual update recursions as a linear system of equations, and then considering the effect when a structured perturbation matrix is incorporated into the equation, we can readily determine upper bounds on the inaccuracies produced during the update. Letting

$$\mathbf{r}_m(n,p) = \begin{pmatrix} f_m(n) \\ b_m(p) \end{pmatrix} \tag{3}$$

$$\mathbf{H}_m(n) = \begin{pmatrix} 1 & \Gamma^*_{f,m}(n) \\ \Gamma^*_{b,m}(n) & 1 \end{pmatrix}, \tag{4}$$

we can rewrite (1) and (2) as:

$$\mathbf{r}_m(n,n) = \mathbf{H}_m(n)\ \mathbf{r}_{m-1}(n,n-1). \tag{5}$$

The perturbations in $\mathbf{H}_m(n)$ will have a special off-diagonal structure while the perturbations in $\mathbf{r}_{m-1}(n,n-1)$ will be more general:

$$\delta\mathbf{H}_m(n) = \begin{pmatrix} 0 & \delta\,\Gamma^*_{f,m}(n) \\ \delta\,\Gamma^*_{b,m}(n) & 0 \end{pmatrix}, \tag{6}$$

$$\delta\mathbf{r}_{m-1}(n,n-1) = \begin{pmatrix} \delta f_{m-1}(n) \\ \delta b_{m-1}(n-1) \end{pmatrix}. \tag{7}$$

Although we wish to compute $\mathbf{r}_m(n,n)$ via (5) at each given update (time and order), we actually compute the update using quantities which have been perturbed. Mathematically, this can be described by

$$\overline{\mathbf{r}_m(n,n)} = (\ \mathbf{H}_m(n) +\ \delta\mathbf{H}_m(n))\,(\ \mathbf{r}_{m-1}(n,n-1) +\ \delta\mathbf{r}_{m-1}(n,n-1))\,. \tag{8}$$

The above representation of (8) provides a natural mechanism to study the nonlinear and compounding effects of finite precision arithmetic. It is important to note that the perturbations $\delta\mathbf{H}_m(n)$ and $\delta\mathbf{r}_{m-1}(n,n-1)$ are defined at the algorithmic level, and hence, these terms are algorithm dependent. For example, the direct and indirect RLSL algorithms are distinguished by the manner in which the reflection coefficients are updated. It has been shown in [5] that $\delta\mathbf{H}_m(n)$ will differ between these algorithms. Independent of algorithmic implementation, however, is the sensitivity to perturbations that is governed by the update structure of (1) and (2). Because of the underlying matrix structure given by the residual update recursions, the perturbation analysis is straight forward once it is realized that we are not, at this juncture, interested in how or where the perturbations occur.

It is well known in numerical linear algebra that the condition number of a matrix measures its relative p-norm nearness to singularity [4]. For the matrix $\mathbf{H}_m(n)$, the following lemma gives the condition number with respect to the 2-norm.

Lemma 1 *Let $\mathbf{H}_m(n)$ be defined by equation (4). The condition number, $\mathcal{K}_2(\mathbf{H}_m(n))$, using the 2-norm for complex matrices is then given by*

$$\mathcal{K}_2(\mathbf{H}_m(n)) = \|\mathbf{H}_m(n)\|_2 \; \|\mathbf{H}_m(n)^{-1}\|_2 \tag{9}$$

$$= \frac{\max(Y_\pm)}{2\,|q|}, \tag{10}$$

where

$$Y_\pm = k \pm \sqrt{k^2 - 4\,|q|^2} \tag{11}$$

$$k = 2 + |\Gamma_{f,m}(n)|^2 + |\Gamma_{b,m}(n)|^2 \tag{12}$$

$$q = 1 - \Gamma_{f,m}(n)\,\Gamma_{b,m}(n). \tag{13}$$

Although the perturbations of the forward and backward reflection coefficients, $\delta\Gamma^*_{f,m}(n)$ and $\delta\Gamma^*_{b,m}(n)$, respectively, are algorithm dependent (e.g., [7], [5]). The sensitivity of equation (5) to perturbations given by (6) and (7) are algorithm independent, however, and can be bounded using well known perturbation theorems from linear algebra (c.f. [2, pp. 80]):

Theorem 1 *Suppose*

$$\mathbf{r}_m(n,n) = \mathbf{H}_m(n)\,\mathbf{r}_{m-1}(n,n-1) \quad \textit{and} \tag{14}$$

$$\overline{\mathbf{r}_m(n,n)} = (\mathbf{H}_m(n) + \delta\mathbf{H}_m(n))\,(\mathbf{r}_{m-1}(n,n-1) + \delta\mathbf{r}_{m-1}(n,n-1)) \tag{15}$$

as defined by (3), (4), (6), and (7), respectively. Then the relative error of the updated residual vector is given by

$$\frac{\|\mathbf{r}_m(n,n) - \overline{\mathbf{r}_m(n,n)}\|_2}{\|\mathbf{r}_m(n,n)\|_2} \le \mathcal{K}_2(\mathbf{H}_m(n)) \left\{ \frac{\|\delta\mathbf{H}_m(n)\|_2}{\|\mathbf{H}_m(n)\|_2} + \frac{\|\delta\mathbf{r}_{m-1}(n,n-1)\|_2}{\|\mathbf{r}_{m-1}(n,n-1)\|_2} \right\} \tag{16}$$

$$= \frac{\max(Y_\pm)}{2\,|1 - \Gamma_{f,m}(n)\,\Gamma_{b,m}(n)|} \left\{ \frac{\|\delta\mathbf{H}_m(n)\|_2}{\|\mathbf{H}_m(n)\|_2} + \frac{\|\delta\mathbf{r}_{m-1}(n,n-1)\|_2}{\|\mathbf{r}_{m-1}(n,n-1)\|_2} \right\}. \tag{17}$$

We see from Theorem 1 that the accuracy of the updated residuals can be adversely affected by the condition number of $\mathbf{H}_m(n)$. This means that the updated value of the forward and backward residual can be a factor $\mathcal{K}_2(\mathbf{H}_m(n))$ larger than the relative perturbations of the same residuals from the previous stage (and, in the case of the backward residual, previous time) update. More precisely, ill-conditioning occurs whenever $1+ \mid \Gamma_{f,m}(n)\ \Gamma_{b,m}(n) \mid^2 \approx 2\mathcal{R}e(\ \Gamma_{f,m}(n)\ \Gamma_{b,m}(n))$.

Before experimentally applying our results to a second order predictor, we show how they also apply to the more general case of an arbitrary order predictor. Consider the problem involving the solution to the following matrix equation:

$$\mathbf{R} \begin{pmatrix} a_0 \\ \vdots \\ a_M \end{pmatrix} = C\mathbf{e_1} \tag{18}$$

for some arbitrary $(M+1) \times (M+1)$ Hermitian Toeplitz matrix $\mathbf{R}$, a constant C, and the first column of the $(M+1)^{st}$-order identity matrix $\mathbf{e_1}$. It is then well-known (e.g., see Haykin [3]) that the polynomial $a(z) = a_0 + a_1 z^{-1} + \ldots + a_M z^{-M}$ will have its zeros strictly in the interior of the unit circle in the complex plane (strictly minimum phase) if and only if $\mathbf{R}$ is positive definite. Additionally, at least one zero of $a(z)$ must lie on the unit circle when $\mathbf{R}$ is positive seme-definite and singular. With respect to the reflection coefficients $\Gamma_{f,i}(n),\ i = 1, \ldots, M$, for $a(z)$ to be strictly minimum phase implies that all reflection coefficients must have modulus less than one ($\mid \Gamma_{f,i}(n) \mid < 1,\ i = 1, \ldots, M$). Lastly, whenever $\mid \Gamma_{f,i}(n) \mid = 1$ for some i, we know that $a(z)$ must have at least i zeros on the unit circle.

$\mathbf{R}$ becomes the theoretical autocorrelation matrix when the input data is formed from the all-pole filter (governed by $1/a(z)$). It is now seen that the conditioning of $\mathbf{R}$ will tend toward singularity whenever the poles of $1/a(z)$ approach the unit circle. Lemma 1 can be applied to estimate the conditioning of $\mathbf{R}$ since an ill-conditioned $\mathbf{R}$ requires at least one of the reflection coefficients to approach unity in modulus.

3 Experiments

In this section we demonstrate how arbitrarily large condition numbers can be detected using Lemma 1. Although our results are not restricted to the all-pole case, for simplicity we make use of this well-understood example since the condition number of the autocorrelation matrix will become arbitrarily close to singularity as the poles of an all-pole filter approach the unit circle. By driving the input sequence to the RLSL algorithm with an all-pole model of order two (with white noise), Theorem 1's ability to correctly detect a potential loss of accuracy in the forward and backward residuals can be easily evaluated. Furthermore, this example illustrates that for the general case in which the conditioning of the underlying autocorrelation matrix in not known, it may be estimated quickly and easily by borrowing well-known results from numerical linear algebra. As a consequence, estimates for the uncertainty in the forward and backward residual updates can then be computed and the main objective of this paper will then be achieved: to demonstrate how numerical linear algebra can be used to better understand the numerical performance of algorithms in signal processing, and how issues concerning numerical performance involve an algorithm's stability as well as the problem it is being asked to solve (i.e., ill-conditioning can undermine the proper execution of the lattice recursions).

We note that the conditioning of $\mathbf{H}_m(n)$ is independent of the algorithm used (e.g., whether the indirect or the direct a posteriori RLSL algorithm is used). Finally, we comment that, being algorithm independent, the condition number for the residual update recursions for other RLSL algorithms such as the QR-lattice algorithm [3] can easily be shown to be the same as that given in Lemma 1.

3.1 Conditioning and 2^{nd} order, all-pole AR processes

To maintain a tractable series of experiments we have chosen four case studies in which the direct RLSL algorithm is given the input process, $\mathbf{u}(i)$, $i = 1, 4096$, that is from a 2^{nd} order, all-pole AR process satisfying the difference equation, $v(n) = u(n) + a_1 u(n-1) + a_2 u(n-2)$. Here the process $v(n)$ is a zero mean, normally distributed white noise process with variance σ_v^2. When $a_1, a_2 \in \mathcal{R}$, we can ensure asymptotic stability (BIBO stable) as long as the AR coefficients satisfy certain inequalities [3, pp. 120-127] paramount to the condition that the roots, p_1 and p_2, to the characteristic equation $1 + a_1 z^{-1} + a_2 z^{-2} = 0$, remain inside the unit circle in the complex plane. If we consider the pairing (a_1, a_2), the inequalities describe the inner region of the triangle with vertices $(0, -1), (-2, 1)$, and $(2, 1)$ that is given in Figure 1 i). For comparative purposes, Figures 1 ii) and 1 iii) detail the regions for which $\mathcal{K}_2(\mathbf{H}_m(n)) \geq 100$ and 1000, respectively. As can be seen, the number of singular systems is vanishingly small but it is possible to approach arbitrarily close to them (cf. the singular matrices in numerical linear algebra). Using the well known fact that for stationary processes the forward and backward reflection coefficients satisfy $\Gamma_{f,m}(n) = \Gamma^*_{b,m}(n)$, it can be easily shown that the AR coefficients satisfy $a_0 = 1$, $a_1 = \Gamma_{f,1}(n) + \Gamma^*_{f,1}(n)\, \Gamma_{f,2}(n)$, and $a_2 = \Gamma_{f,2}(n)$. We understand that the reflection coefficients are constant, but we have nonetheless maintained the more general notation by including the time index, n. Summarizing, we can use (5) in conjunction with Theorem 1 to derive a one-to-one correspondence between the reflection coefficients and the conditioning of $\mathbf{H}_m(n)$. For stationary input processes there also exists a one-to-one correspondence between the conditioning of $\mathbf{H}_m(n)$ and the correlation matrix. Although this correspondence is more difficult to assess for the nonstationary case, Theorem 1 will still apply to (5).

3.2 Measured sensitivity to the condition number

In this section we present four case studies to illustrate the applicability of Theorem 1 for sensing and tracking the conditioning of (5). The first experiment uses a set of AR coefficients that gives rise to a small condition number for the matrix $\mathbf{H}_m(n)$. The second through fourth experiments use AR coefficients that cause the matrix $\mathbf{H}_m(n)$ to have varying degrees of ill-conditioning.

The direct RLSL algorithm, programmed in C, was run from Matlab[1]. The algorithm used is the same as given in [10]. To compute the normed quantities $\| \delta\mathbf{H}_m(n)\|_2$ and $\| \delta\mathbf{r}_{m-1}(n, n-1)\|_2$, the RLSL algorithm was run in both double and single precision. This effectively gives a normed measurement of the uncertainty in the quantities rather than the norm of the exact errors. The left hand quantity $\| \mathbf{r}_m(n,n) - \overline{\mathbf{r}_m(n,n)}\|_2$ was computed in a like manner. The details describing each AR process is given in Table 1. The variance of the input process, δ_v^2, was chosen in each case so that the output process, $u(n)$, would have unit variance. It was deemed appropriate to choose at least one experiment from the literature. The first case was selected from [3, pg. 123] and the third case studied was selected from [9]. The results from each study are presented in Table 2. Presented

[1]The authors would like to thank Richard North for supplying the direct RLSL algorithm.

in the middle three columns are the mean values for each of the terms on the right-hand side of (16). For comparative purposes, the two right-most columns give the computed mean values for each side of (16). It is seen that for each case studied, the relative perturbations of $\mathbf{H}_1(n)$ and $\mathbf{r}_0(n, n-1)$ are of order $O(10^{-7})$ and $O(10^{-8})$, respectively. This fact demonstrates the dramatic influence the conditioning has on (5). The last two columns of Table 2 for each of these case studies were also graphed at each iteration and are found in Figures 2 and 3. Each of the cases are graphed with the corresponding computed condition number appearing directly below. In each case the line representing the estimated perturbation bounds (the last column of Table 2) is found above the measured relative uncertainty (the fifth column of Table 2). The computed values of the condition numbers of $\mathbf{H}_1(n)$ also are seen to rapidly converge to their expected values.

4 Conclusions

In this paper we have demonstrated, by looking at the residual update recursions that comprise the lattice structure as a set of linear equations given by (5), it is possible to detect ill-conditioning which can grossly magnify any perturbations that naturally occur when arithmetic is performed using finite precision. This ill-conditioning is associated with the update recursions and therefore is independent of the manner from which parameters such as the reflection coefficients are updated. Although our results apply not only to the arbitrary-order predictor, but any condition that would affect the conditioning of $\mathbf{H}_m(n)$, we use low order AR filters in our case studies to reveal how the condition number can magnify small perturbations into much larger ones. Previously, the degree of influence that an algorithm's stability and a problem's conditioning has on algorithmic performance has been unclear. We hope that the results will more easily reveal unstable algorithms by demonstrating that by choosing a well-conditioned input process, any poor numerical effects must be a result of the algorithm's poor stability properties.

5 Acknowledgments

The authors wish to thank the reviewers for their valuable suggestions, in particular, the detailed recommendation to include the application of our results to the arbitrary order case for the all-pole AR filter.

References

[1] BUNCH, J.R. and Le BORNE, R.C.: 'Error accumulation effects for the a posteriori RLSL prediction filter,' *IEEE Transactions on Signal Processing*, 1995, **43**, pp. 150-159

[2] GOLUB, G.H. and VAN LOAN, C.F.: 'Matrix Computations', (The Johns Hopkins University Press, Baltimore and London, 1989) 2nd edn.

[3] HAYKIN, S.: 'Adaptive Filter Theory', (Prentice-Hall, Englewood Cliffs, NJ, 1991) 2nd edn.

[4] KAHAN, W.: 'Numerical linear algebra,' *Canadian Math. Bull.*, 1966, **9**, pp. 757-801

[5] Le BORNE, R.C.: 'Numerical roundoff effects and the direct and indirect RLSL algorithms,' *Proceedings to the 4th International Conference on Mathematics in Signal Processing*, 1996.

[6] LING, F., MANOLAKIS, D. and PROAKIS, J.G.: 'New forms of LS lattice algorithms and an analysis of their round-off error characteristics'. Proc. ICASSP, 1985, Tampa, FL, pp. 1739-1742

[7] LING, F., MANOLAKIS, D. and PROAKIS, J.G.: 'Numerically robust least-squares lattice-ladder algorithms with direct updating of the reflection coefficients,' *IEEE Trans. on Acoustics, Speech, and Signal Processing*, **ASSP-34**, (4), pp. 837-845

[8] LING, F. and PROAKIS, J.G.: 'Numerical accuracy and stability: two problems of adaptive estimation algorithms caused by round-off error'. Proc. ICASSP, 1984, San Diego, CA, pp. 30.3.1–30.3.4

[9] MATHEWS, J.V. and XIE, Z.: 'Fixed-point error analysis of stochastic gradient adaptive lattice filters,' *IEEE Trans. Acoustics, Speech, and Signal Processing*, 1990, **38**, (1)

[10] NORTH, R., ZEIDLER, J., KU, W. and ALBERT, T.: 'A floating-point arithmetic error analysis of direct and indirect coefficient updating techniques for adaptive lattice filters,' *IEEE Trans. on Signal Processing*, May 1993.

[11] REGALIA, P.: 'Numerical stability issues in fast least-squares adaptation algorithms,' *Opt. Engr.*, 1992, **31**, (6), pp. 1144-1152

[12] SLOCK, D.T.M.: 'Backward consistency concept and round-off error propagation dynamics in recursive least-squares algorithms,' *Opt. Engr.*, 1992, **31**, (6), pp. 1153–1169

$v(n) = u(n) + a_1 u(n-1) + a_2 u(n-2)$						
Case	AR coef's		Roots			Expected
#	a_1	a_2	$\lvert p_1 \rvert$	$\lvert p_2 \rvert$	σ_v^2	$\mathcal{K}_2(\mathbf{H}_1(n))$
1	0.100	-0.8	.9	.9	2.70×10^{-1}	3.000×10^{0}
2	1.970	.990	.99	.99	3.980×10^{-4}	1.980×10^{2}
3	-1.8	.81	1.0000	.8	3.790×10^{-3}	3.610×10^{2}
4	1.997	.999	.9995	.9995	3.996×10^{-6}	1.998×10^{3}

Table 1: Four 2^{nd} order AR processes and their associated conditioning.

Theorem 1: Computed Mean					
Case	$\lVert \cdot \rVert_2$			$\lVert \cdot \rVert_2$	$\lVert \cdot \rVert_2$
#	$\mathcal{K}_2(\mathbf{H}_1(n))$	$\frac{\delta \mathbf{H}_1(n)}{\mathbf{H}_1(n)}$	$\frac{\delta \mathbf{r}_0(n,n-1)}{\mathbf{r}_0(n,n-1)}$	Left-hand Side	Right-hand Side
1	2.758×10^{0}	1.3993×10^{-7}	2.3507×10^{-8}	1.9529×10^{-7}	4.5074×10^{-7}
2	1.953×10^{2}	1.7354×10^{-7}	3.1238×10^{-8}	4.2037×10^{-6}	3.9999×10^{-5}
3	3.531×10^{2}	3.3826×10^{-7}	3.1867×10^{-8}	1.1534×10^{-5}	1.3068×10^{-4}
4	1.984×10^{3}	1.8231×10^{-7}	3.2519×10^{-8}	1.9897×10^{-5}	4.2622×10^{-4}

Table 2: Effects from the conditioning of $\mathbf{H}_1(n)$ on the updated residuals

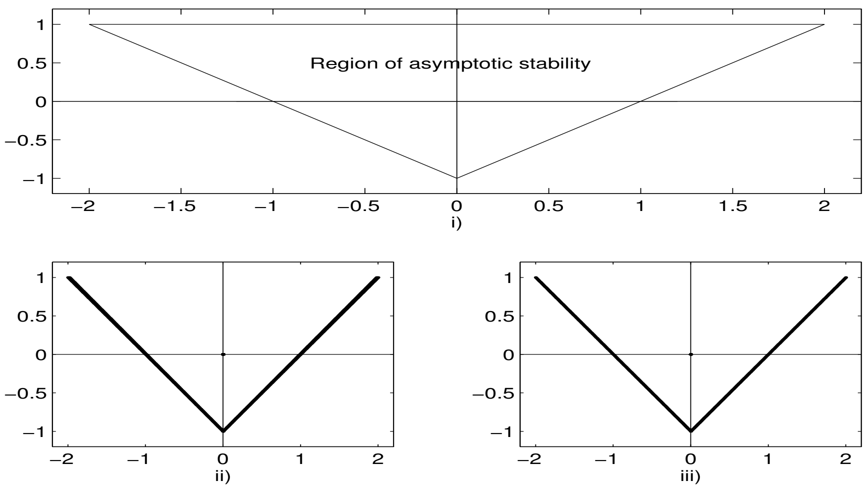

Figure 1: 2nd order AR coefficients (a_1, a_2): i) Triangular region bounding the permissible pairings to ensure asymptotic stability. ii) The thick region depicts all pairings that generate a condition number $\mathcal{K}_2(\mathbf{H}_m(n)) \geq 100$ and, iii) $\mathcal{K}_2(\mathbf{H}_m(n)) \geq 1000$.

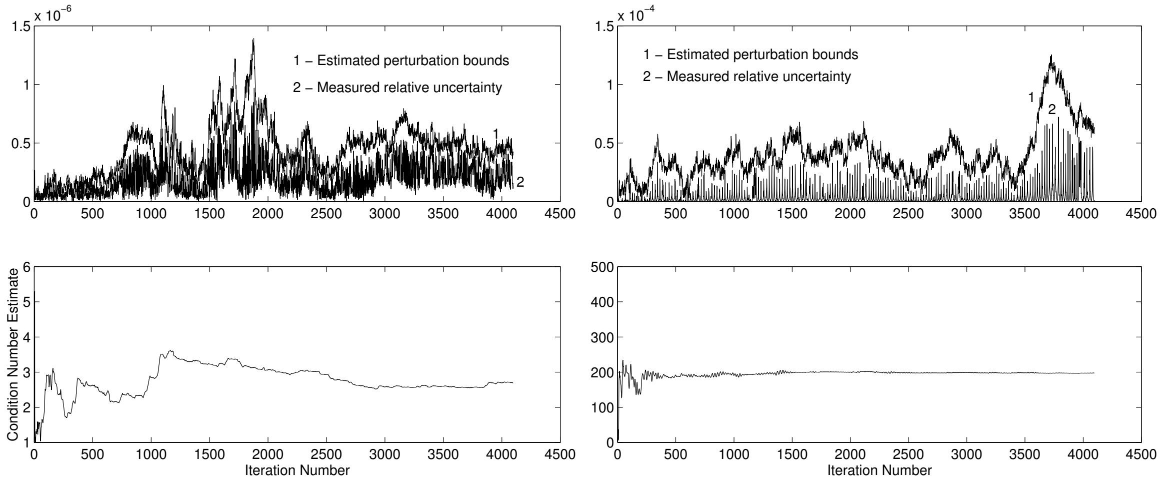

Figure 2: Two AR-models of order 2: Coefficients chosen to yield (left) a $O(1)$ condition number and (right) one that is $O(10^2)$.

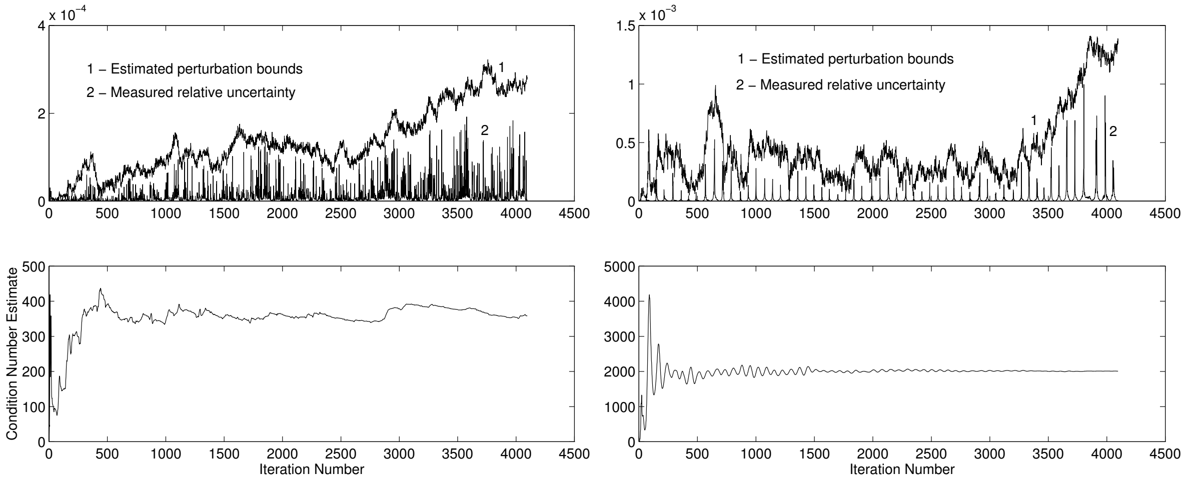

Figure 3: Two AR-models of order 2: Coefficients chosen to yield (left) a $O(10^2)$ condition number and (right) one that is $O(10^3)$.