Combined high-order algorithms in robust least-squares estimation with harmonic regressor and strictly diagonally dominant information matrix

(article starts on next page)

Combined high-order algorithms in robust least-squares estimation with harmonic regressor and strictly diagonally dominant information matrix

Alexander Stotsky

Abstract

This article describes new high-order algorithms in the least-squares problem with harmonic regressor and strictly diagonally dominant information matrix. Estimation accuracy and the number of steps to achieve this accuracy are controllable in these algorithms. Simplified forms of the high-order matrix inversion algorithms and the high-order algorithms of direct calculation of the parameter vector are found. The algorithms are presented as recursive procedures driven by estimation errors multiplied by the gain matrices, which can be seen as preconditioners. A simple and recursive (with respect to order) algorithm for update of the gain matrix, which is associated with Neumann series, is found. It is shown that the limiting form of the algorithm (algorithm of infinite order) provides perfect estimation. A new form of the gain matrix is also a basis for unification method of high-order algorithms. New combined and fast convergent high-order algorithms of recursive matrix inversion and algorithms of direct calculation of the parameter vector are presented. The stability of algorithms is proved and explicit transient bound on estimation error is calculated. New algorithms are simple, fast and robust with respect to round-off error accumulation.

Introduction

Recursive least-squares (RLS) algorithms are widely used in many applications such as adaptive control, signal processing, system identification and others.[1] The parameter vector is updated recursively in each step of RLS algorithm, using information available in the previous step as follows

$$\theta_i = \theta_{i-1} + \frac{\Gamma_{i-1}\varphi_i}{\lambda_0 + \varphi_i^T \Gamma_{i-1}\varphi_i}(y_i - \theta_{i-1}^T \varphi_i) \tag{1}$$

where θ_i is the parameter vector, φ_i is the regressor vector, y_i is the measured signal, Γ_{i-1} is a gain matrix and λ_0 is a forgetting factor, $i = 1, 2, ...$ Equation (1) shows clearly that round-off errors that are accumulated in step i-1 propagate to step i. This propagation has a direct impact on the estimation performance and even on system stability (see the articles by Ljung and

Ljung,[2] Slock,[3] Liavas and Regalia[4] and references therein for quantification of the performance deterioration in the presence of round-off errors).

RLS algorithm (1) is a recursive realization of the solution of the following equation

$$A_i \theta_i = b_i \tag{2}$$

where a symmetric matrix A_i is called an information matrix and b_i is the vector that contains measured signal. To tackle a round-off error propagation problem,

Division of Electric Power Engineering, Department of Energy and Environment, Chalmers University of Technology, Gothenburg, Sweden

Corresponding author:
Alexander Stotsky, Division of Electric Power Engineering, Department of Energy and Environment, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden.
Email: alexander.stotsky@chalmers.se

equation (2) is solved with respect to θ_i in each step i. Two groups of methods are used to solve linear algebraic equation (2): exact methods and recursive methods.

Exact methods represent a finite number of calculations to find the parameter vector. Gaussian elimination method, Lower Upper (LU) decomposition, square root method and others are the best known exact methods for calculation of the parameter vector.[5–8] Exact methods do not give an exact solution of the algebraic equations due to round-off errors, which are always present in a finite precision implementation environment, and become computationally expensive for large-scale systems. Moreover, estimation of the accuracy of the calculated parameter vector is quite difficult for exact methods. Nevertheless, equation (2) can be solved using exact methods in each step with some errors, and those errors do not propagate to the next step.

Recursive methods, such as Jacobi and Gauss–Seidel iterative methods, provide a solution with controllable accuracy. However, (1) recursive methods may require a large number of iterations to achieve high accuracy, (2) the stability of the methods depends on the initial errors and (3) finally these methods are computationally expensive. In other words, the recursive methods are not directly applicable in practice due to the potential stability problems and computational complexity.

Performance of the recursive methods may be improved by taking into account the properties of information matrix A_i. This matrix is a strictly diagonally dominant (SDD) and positive definite matrix,[9,10] if the regressor φ_i is a harmonic regressor,[11] which consists of trigonometric functions (sines and cosines) at different frequencies.

Notice that numerical methods for many physical systems, machine learning, random processes, computer vision, image processing, network analysis and computational biology are also a source of linear systems of equations with SDD matrices (see Ye,[7] Kelner et al.[12] and references therein).

Notice also that Jacobi and Gauss–Seidel methods converge, if information matrix is an SDD matrix, but the convergence may be slow.

The following inequality, which plays a key role in the algorithm design, is valid for SDD matrices

$$\left\| I - D_i^{-1} A_i \right\| < 1 \qquad (3)$$

where A_i is an SDD matrix, D_i is a diagonal matrix that contains diagonal elements of A_i, I is the identity matrix and the norm is defined as the maximum row sum matrix norm. This property can be used for performance improvement of the recursive algorithms, which in turn can be divided into two groups: recursive inversion of matrix A_i and direct recursive estimation of the parameter vector θ_i.

Recursive inversion algorithms use the diagonal matrix D_i^{-1} as a staring point for recursive inversion,

where the convergence is guaranteed by inequality (3). Recursive inversion algorithms can be classified with respect to order. The second-order algorithm is known as Hotelling–Bodewig[13] or Newton–Schulz algorithm,[14] and it is described in many books and articles (see, for example, Schulz,[15] Demidovich and Maron[16] and Ben-Israel[17]). The third-order algorithm is described in Krishnamurthy and Sen,[18] Li et al.[19] and Codevico et al.,[20] and finally, sixth- and seventh-order algorithms are proposed in Krishnamurthy and Sen[18] and Soleymani,[13] respectively. Moreover, some initial form of high-order algorithms was outlined in Li and Li[21] using a high-order error model.

Recursive algorithms of direct estimation of parameter vector are also based on property (3), where the matrix $I - D_i^{-1} A_i$ is directly associated with the error model. Such algorithms can also be classified with respect to order. The first-order algorithm was described in Stotsky,[9] and initial form of high-order algorithms was proposed in Stotsky.[22] The first-order algorithm is also known as Jacobi method[14] presented in the Richardson form.[23]

Since the convergence of recursive inversion algorithms and algorithms of direct calculation of the parameter vector is based on the same inequality (3), these algorithms can be combined. A combination of the second-order matrix inversion algorithm and high-order algorithm of direct calculation of the parameter vector is presented in Stotsky.[22]

Application of high-order algorithms to robust least-squares estimation has a high potential since both an accuracy and the number of steps to achieve this accuracy are controllable. This article is dedicated toward further development of recursive high-order algorithms. The contributions of this article can be summarized as follows:

1. Simplified forms for both the high-order matrix inversion algorithms and the high-order algorithms of direct calculation of the parameter vector are found. The algorithms are presented as recursive procedures driven by estimation errors multiplied by the gain matrices. A simple and recursive (with respect to order) algorithm for update of the gain matrices is found, where infinite order of the algorithm (the limiting form) provides perfect estimation. The gain matrix is associated with Neumann series $\sum_{d=0}^{n-1} F_0^d$, which converges to $(D_i^{-1} A_i)^{-1}$ as order $n \to \infty$, providing information about inverse of the information matrix, where $F_0 = I - D_i^{-1} A_i$. The convergence of this matrix series is guaranteed again by inequality (3). This form of the gain matrix provides also unification method for high-order algorithms and facilitates implementation and development of combined algorithms.

2. New combined and fast convergent high-order algorithms of recursive matrix inversion and algorithms of direct calculation of the parameter vector are the main contribution of this article. The

stability of algorithms is proved and explicit transient bound on estimation error is calculated.

This article is organized as follows. Problem statement of estimation of the frequency content of oscillating signal is given in section "Problem statement." Recursive algorithms of direct calculation of the parameter vector are presented in section "Recursive algorithms of direct calculation of the parameter vector." Inversion algorithms of an SDD matrix are discussed in section "Inversion algorithms of an SDD matrix," and combined algorithms are presented in section "Combined high-order algorithms." This article ends with brief conclusions in section "Conclusion."

Problem statement

Suppose that a measured oscillating signal y_k can be presented in the following form

$$y_i = \varphi_i^T \theta_* + \xi_i \tag{4}$$

where φ_i is the harmonic regressor and θ_* is the vector of constant unknown parameters defined as follows

$$\varphi_i^T = [1 \cos(q_1 i) \sin(q_1 i) \cos(q_2 i) \sin(q_2 i) \ldots \cos(q_r i) \\ \sin(q_r i)] \tag{5}$$

$$\theta_*^T = \begin{bmatrix} \theta_{0*} & \theta_{1*} & \theta_{2*} & \theta_{3*} & \theta_{4*} & \ldots & \theta_{(2r-1)*} & \theta_{(2r)*} \end{bmatrix} \tag{6}$$

where $i = 1, 2, \ldots$ is the step number; q_p, $p = 1, 2, \ldots, r$, are the frequencies and ξ_i is a zero mean white Gaussian noise. The model of signal (4) is presented in the following form

$$\hat{y}_i = \varphi_i^T \theta_i \tag{7}$$

with adjustable parameters

$$\theta_i^T = \begin{bmatrix} \theta_{0i} & \theta_{1i} & \theta_{2i} & \theta_{3i} & \theta_{4i} & \ldots & \theta_{(2r-1)i} & \theta_{(2r)i} \end{bmatrix} \tag{8}$$

Least-squares solution for estimation of the parameter vector can be written as follows[1]

$$\theta_i = A_i^{-1} b_i \tag{9}$$

$$A_i = \lambda_0 \left[\sum_{j=1}^{i-1} \varphi_j \varphi_j^T \right] + \varphi_i \varphi_i^T \tag{10}$$

$$b_i = \lambda_0 \left[\sum_{j=1}^{i-1} \varphi_j y_j \right] + \varphi_i y_i \tag{11}$$

where matrix A_i is the information matrix, and $0 < \lambda_0 < 1$ is a forgetting factor, $i = 2, 3, \ldots$ The matrix A_i is an SDD matrix for a sufficiently large i and forgetting factor λ_0 which is close to 1.[24] This matrix property is used in the next sections for the algorithm design and analysis. Notice that positive definiteness of

information matrix for system with harmonic regressor can also be shown using partitioning method for insufficiently large window size.[25]

Recursive algorithms of direct calculation of the parameter vector

First-order algorithm

Least-squares problem is a problem of calculation of the parameter vector θ_i with high accuracy. First-order algorithm of direct calculation of the parameter vector can be described as follows[9]

$$\vartheta_k = \vartheta_{k-1} - G_0(A_i \vartheta_{k-1} - b_i) \tag{12}$$

$$\tilde{\vartheta}_k = F_0 \tilde{\vartheta}_{k-1} \tag{13}$$

where $\tilde{\vartheta}_k = \vartheta_k - \theta_i$ is the estimation error, and the error matrix is defined as follows

$$F_0 = I - G_0 A_i \tag{14}$$

and $G_0 = D_i^{-1}$, and D_i is a diagonal matrix that contains diagonal elements of A_i, which means that the inverse of the diagonal matrix D_i^{-1} is used as approximation of A_i^{-1}.

The following property of the error matrix F_0 is established for SDD matrix A_i

$$\|F_0\| \leqslant \kappa < 1 \tag{15}$$

where the norm is defined as the maximum row sum matrix norm. Inequality (15) guarantees the stability of error model (13).

Notice that matrix G_0 is usually called the preconditioner (preconditioning matrix) and matrix F_0 is called the iteration matrix. Preconditioner is directly associated with estimate of the inverse of A_i. Moreover, algorithm (12) can be seen as Jacobi method, written in the Richardson form.[14]

High-order algorithm: three equivalent forms of the gain matrix

High-order algorithms of direct calculation of the parameter vector are described in Stotsky[22]

$$\vartheta_k = \vartheta_{k-1} - \underbrace{\Gamma_n}_{gain} G_0 \underbrace{\{A_i \vartheta_{k-1} - b_i\}}_{error} \tag{16}$$

$$\Gamma_n = \sum_{d=0}^{n-1} \frac{n!}{(n-d)! d!} (-1)^{n-d-1} (G_0 A_i)^{n-d-1} \tag{17}$$

$$\tilde{\vartheta}_k = F_0^n \tilde{\vartheta}_{k-1}, \quad \tilde{\vartheta}_k = F_0^{nk} \tilde{\vartheta}_0 \tag{18}$$

where $n = 1, 2, 3, \ldots$ is referred as an order of the algorithm, $k = 1, 2, 3, \ldots$ and $\|F_0\| < 1$.

High-order algorithms are tabulated in Table 1. The order of the algorithm is chosen according to the accuracy requirements. The order of the algorithm can be easily increased recursively in algorithms (16) and (17) by adding additional terms to the gain matrix. The gain

Table 1. A family of high-order algorithms.

Algorithms for direct calculation of parameters	
$\vartheta_k = \vartheta_{k-1} - \Gamma_n G_0 \{A_i \vartheta_{k-1} - b_i\}$, $F_0 = I - G_0 A_i$, $\|F_0\| < 1$	
Order	Gain matrix Γ_n
1	I
2	$I + F_0$
3	$I + F_0 + F_0^2$
4	$I + F_0 + F_0^2 + F_0^3$
...	...
n	$\sum_{d=0}^{n-1} F_0^d \to (G_0 A_i)^{-1}$ as $n \to \infty$

matrix Γ_n of the algorithm of order n, where $n = 2, 3, 4, \ldots$, is calculated recursively via the gain matrix Γ_{n-1} as follows

$$\Gamma_n = \Gamma_{n-1} + F_0^{n-1} \qquad (19)$$

where $\Gamma_1 = I$, and F_0^{n-1} is the gain increment. Identity (19) is proved via explicit evaluation of the first difference $\Gamma_n - \Gamma_{n-1}$, where Γ_n is defined in equation (17). Gain matrix Γ_n can also be presented as Neumann series as follows

$$\Gamma_n = \underbrace{\underbrace{\underbrace{I + F_0}_{sec\,ond\,order,\,n=2} + F_0^2}_{third\,order,\,n=3} + F_0^3}_{fourth\,order,\,n=4} + \cdots + F_0^{n-1}}_{nth\,order} \qquad (20)$$

and

$$\Gamma_n = \sum_{d=0}^{n-1} F_0^d. $$

Notice that incremental relation (19) depends on the order only and does not depend on estimation error. This relation allows to increase the order recursively and plays a key role in algorithm (16). This relation has a direct impact on the accuracy of estimation and allows to associate the gain matrix with convergent matrix series. This convergence property results in a new notion of the infinite order algorithm, described in the next section.

The infinite order algorithm: limiting form with $\Gamma_\infty = \lim_{n \to \infty} \Gamma_n$

The gain matrix $\Gamma_n = \sum_{d=0}^{n-1} F_0^d$ converges to the matrix $(G_0 A_i)^{-1}$ as order n tends to infinity, that is

$$\Gamma_n = \sum_{d=0}^{n-1} F_0^d \to (G_0 A_i)^{-1} = A_i^{-1} G_0^{-1}   \text{ as } n \to \infty \qquad (21)$$

where $\|F_0\| < 1$ and $F_0 = I - G_0 A_i$. Relation (21) follows directly from the convergence properties of the

matrix series (see theorem 5.6.15 in Horn and Johnson[26]). Notice that the convergence rate of the recursive matrix inversion algorithm based on Neumann series is compared to the convergence rate of the second-order algorithms (25) and (26) in Appendix 3 of Stotsky.[9]

Substitution of $\Gamma_\infty = \lim_{n \to \infty} \Gamma_n = A_i^{-1} G_0^{-1}$ in equation (16) results in the following identity $\vartheta_k = \theta_i$. Order n quantifies the proximity between the gain matrix Γ_n and the matrix $A_i^{-1} G_0^{-1}$, which contains information about the inverse of the information matrix and provides a perfect estimation, where

$$\Gamma_n = (G_0 A_i)^{-1} - (G_0 A_i)^{-1}(I - G_0 A_i)^n \qquad (22)$$

$$I - \Gamma_n G_0 A_i = (I - G_0 A_i)^n = F_0^n \qquad (23)$$

Substitution of Γ_n defined in equations (22) and (23) in equation (16) results in error model (18). Finally, the gain matrix Γ_n is presented in three equivalent forms (17), (19) and (22), where equation (19) is used in implementation, equation (22) is useful for stability analysis and equation (17) can be seen as intermediate form.

Notice that the product $\Gamma_n G_0$ in equation (16) can also be seen as a composite preconditioner, in which Γ_n is a polynomial preconditioner (see Chen[14] for other types of preconditioning).

Notice that the dynamics of the tracking error $\varepsilon_k = A_i \vartheta_k - b_i$ can be presented as follows

$$\varepsilon_k = A_i \left[\vartheta_{k-1} - \Gamma_n G_0 \underbrace{\{A_i \vartheta_{k-1} - b_i\}}_{\varepsilon_{k-1}} \right] - b_i \qquad (24)$$

$$= (I - A_i \Gamma_n G_0) \varepsilon_{k-1}$$

and substitution of $\Gamma_\infty = A_i^{-1} G_0^{-1}$ in equation (24) implies also that $\varepsilon_k = 0$.

Notice also that algorithm (16) has two loops and the estimation error can be minimized by (1) iterating of ϑ_k for a fixed order and minimizing the error $A_i \vartheta_{k-1} - b$ and (2) increasing the order, where the gain matrix converges to $A_i^{-1} G_0^{-1}$.

High accuracy of estimation may be achieved in one step, increasing the order of the algorithm. The performance of the algorithm is illustrated in Figure 1, where the error norm is plotted after the first iteration.

Inversion algorithms of an SDD matrix

Second-order algorithm

The parameter vector can be also calculated via equation (9) and an estimate of the inverse of the information matrix. Strict diagonal dominance of the information matrix A_i allows fast and computationally efficient estimation of the inverse of information matrix as follows

$$G_k = G_{k-1} + F_{k-1} G_{k-1} \qquad (25)$$
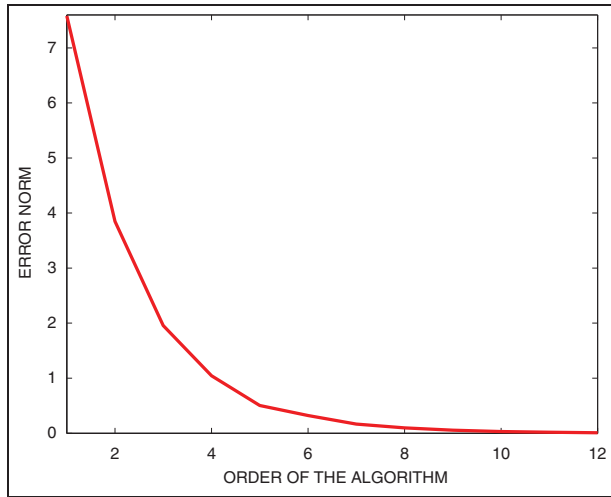
$$F_k = I - G_k A_i \qquad (26)$$

Figure 1. The error norm $\tilde{\vartheta}_1 = \vartheta_1 - \theta_i$ is plotted as function of the order of algorithm (16) after the first step.

where G_k is an estimate of the inverse of A_i, matrix F_k is associated with the inversion error, I is the identity matrix and $k = 1, 2, ...$ Algorithms (25) and (26) are initialized according to equation (14), where inequality (15) is valid.

Matrix F_k that represents the inversion error can be written as follows

$$F_k = I - G_k A_i = F_0^{2^k} \qquad (27)$$

and $\lim_{k \to \infty} G_k = A_i^{-1}$ due to property (15).

Algorithms (25) and (26) can be derived from the following error model

$$F_k = F_{k-1}^2 \qquad (28)$$
$$I - G_k A_i = (I - G_{k-1} A_i)^2 \qquad (29)$$

where equation (29) is solved with respect to G_k, and it is called the second-order algorithm or Newton–Schulz algorithm.[15–17]

High-order matrix inversion algorithm

A family of high-order matrix inversion algorithms is derived from the following error model

$$F_k = F_{k-1}^m \qquad (30)$$

where $m = 2, 3, ...$ is referred as the order of the algorithm, and $F_k = F_0^{m^k}$, where F_k and F_0 are defined in equations (26) and (14), respectively, $k = 1, 2, 3, ...$

Solving equation (30) with respect to G_k yields

$$G_k = G_{k-1} + \underbrace{F_{k-1}}_{error} \underbrace{L_m}_{gain} G_{k-1} \qquad (31)$$

where L_m is a gain matrix, which is defined as follows

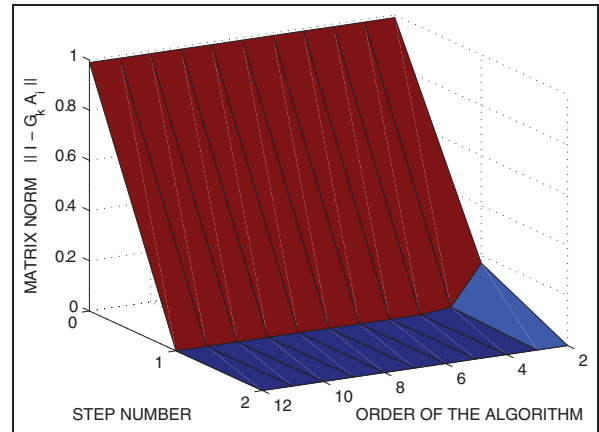$$L_m = \sum_{d=0}^{m-2} \frac{(m-1)!}{(m-d-1)! \, d!} (-1)^{m-d-2} (G_{k-1} A_i)^{m-d-2} \qquad (32)$$

Figure 2. The norm of the error matrix F_k is plotted for three steps $k = 0, 1, 2$ as a function of order m.

Table 2. A family of high-order matrix inversion algorithms.

Matrix inversion algorithms

$G_k = G_{k-1} + F_{k-1} L_m G_{k-1}$, $F_{k-1} = I - G_{k-1} A_i$, $||F_0|| < 1$

Order	Gain update algorithm L_m
2	I
3	$I + F_{k-1}$
4	$I + F_{k-1} + F_{k-1}^2$
5	$I + F_{k-1} + F_{k-1}^2 + F_{k-1}^3$
...	...
m	$\sum_{d=0}^{m-2} F_{k-1}^d \to (G_{k-1} A_i)^{-1}$ as $m \to \infty$

The gain matrix of order m is calculated recursively via the gain matrix of order $m-1$ (similar to equation (19)), where $m = 3, 4, 5, ...$ as follows

$$L_m = L_{m-1} + F_{k-1}^{m-2} \qquad (33)$$

where $L_2 = I$ and $L_m = \sum_{d=0}^{m-2} F_{k-1}^d$. Notice that the gain matrix L_m can be presented in the following form $L_m = (G_{k-1} A_i)^{-1} - (G_{k-1} A_i)^{-1} F_{k-1}^{m-1}$, $m = 2, 3, ...,$ and error model (30) is obtained after substitution of this gain matrix in equation (31).

High-order algorithms are tabulated in Table 2. The performance of high-order algorithms is illustrated in Figure 2, where the norm of the error matrix F_k is plotted for three steps $k = 0, 1, 2$ as a function of order m.

Notice that algorithms (31) and (33) are a unified form of high-order algorithms described in Soleymani,[13] Li and Li[21] and in many other articles.

The infinite order algorithm, $m = \infty$

The gain matrix $L_m = \sum_{d=0}^{m-2} F_{k-1}^d$ converges to the matrix $(G_{k-1} A_i)^{-1}$ as order m tends to infinity, that is

$$L_m = \sum_{d=0}^{m-2} F_{k-1}^d \to (G_{k-1} A_i)^{-1} = A_i^{-1} G_{k-1}^{-1} \quad \text{as } m \to \infty \qquad (34)$$

where $\|F_{k-1}\| < 1$. Substitution of $L_{\infty} = A_i^{-1}G_{k-1}^{-1}$ in equation (31) results in the following identity $G_k = A_i^{-1}$.

Combined high-order algorithms

Description of the algorithms

High-order algorithms described in section "High-order algorithm: three equivalent forms of the gain matrix" and section "High-order matrix inversion algorithm" can be combined as follows

$$F_{k-1} = \underbrace{I - G_{k-1}A_i}_{\text{inversion error}} \tag{35}$$

$$L_m = L_{m-1} + F_{k-1}^{m-2} = \sum_{d=0}^{m-2} F_{k-1}^d \tag{36}$$

$$G_k = G_{k-1} + F_{k-1}L_mG_{k-1} \tag{37}$$

$$F_k = I - G_kA_i \tag{38}$$

$$\Gamma_n = \Gamma_{n-1} + F_k^{n-1} = \sum_{d=0}^{n-1} F_k^d \tag{39}$$

$$\vartheta_k = \vartheta_{k-1} - \Gamma_n G_k \underbrace{\{A_i\vartheta_{k-1} - b_i\}}_{\text{estimation error}} \tag{40}$$

where G_k is an estimate of the inverse of information matrix A_i, F_k is an estimation error and ϑ_k is an estimate of the parameter vector θ_i. Algorithms (35)–(40) are initialized according to equation (14) with two parameters to be chosen. The first one is the order of the matrix inversion algorithm $m = 2, 3, 4, ...$, and the second one is the order of the algorithm of direct calculation of the parameters $n = 1, 2, 3, ...$ These orders have impact on the gain matrices L_m and Γ_n only. The computational burden of the algorithm may be reduced using the same calculations for Γ_n and L_m.

Notice that the substitution of $L_{\infty} = A_i^{-1}G_{k-1}^{-1}$ and $\Gamma_{\infty} = A_i^{-1}G_k^{-1}$ in equations (37) and (40) implies that $G_k = A_i^{-1}$ and $\vartheta_k = \theta_i$. Matrix G_k can be seen as preconditioner and matrices L_m and Γ_n are polynomial preconditioners.

The following error model is valid for algorithms (35)–(40)

$$\tilde{\vartheta}_k = F_k^n\tilde{\vartheta}_{k-1} \tag{41}$$

$$F_k = F_{k-1}^m \tag{42}$$

where $\tilde{\vartheta}_k = \vartheta_k - \theta_i$, $k = 1, 2, ...$ This model can be written in the following form

$$\tilde{\vartheta}_k = F_0^{\frac{n(m^{k+1}-m)}{(m-1)}}\tilde{\vartheta}_0 \tag{43}$$

where $F_0 = I - G_0A_i$, $\tilde{\vartheta}_0 = \vartheta_0 - \theta_i$ and $\|F_0\| < 1$, which guarantees the system stability.

Which order n or m?

Two parameters should be chosen in algorithms (35)–(40): order n and order m. Error model (43) can
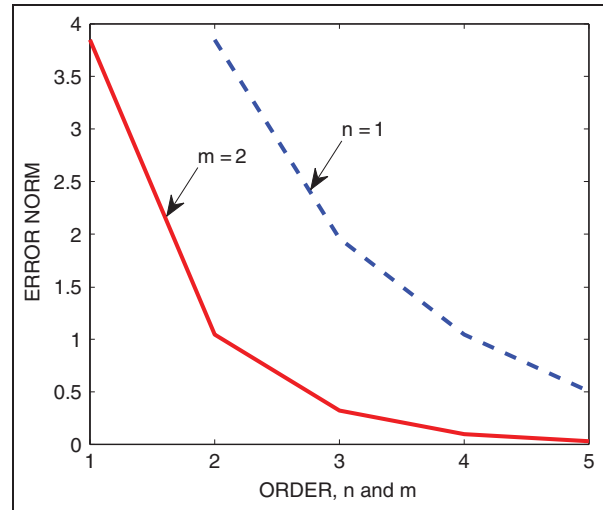

Figure 3. The error norm $\tilde{\vartheta}_1 = \vartheta_1 - \theta_i$ is plotted as a function of orders n and m of algorithms (35)–(40) after the first step.

be used to facilitate the choice of these orders. The norm of the error $\tilde{\vartheta}_1$ calculated in one step for two cases is plotted in Figure 3. The norm of the second-order matrix inversion algorithm, $m = 2$, is plotted with a red solid line for the order of the algorithm of direct calculation of the parameter vector which is equal to $n = 1, 2, 3, 4, 5$. The same norm for the first-order algorithm of direct calculation of the parameter vector, $n = 1$, is plotted with a blue dashed line for the order of the matrix inversion algorithm which is equal to $m = 2, 3, 4, 5$. The figure shows that higher order of the algorithm of direct calculation of the parameter vector should be chosen that essentially reduces estimation error. Notice that the error model for $m = 2$ and $n = 5$ and for $n = 1$ and $m = 5$ gets the form $\tilde{\vartheta}_1 = F_0^{10}\tilde{\vartheta}_0$ and $\tilde{\vartheta}_1 = F_0^5\tilde{\vartheta}_0$, respectively, with $k = 1$.

Conclusion

RLS algorithms, which are widely used in practice aiming to reduce computational complexity, suffer from round-off error accumulation problem. This is a main obstacle to real-time implementation of RLS algorithms and motivation to look back on classical solutions of algebraic equations. These solutions are usually computationally expensive, which is again an obstacle to real-time implementation. The performance of classical methods can be improved by taking into account the properties of the information matrix, which is an SDD matrix for systems with harmonic regressor. This idea attracted significant research efforts in recent years related to exact methods of solution of algebraic equations[7] and recursive methods.[13,18,21] Recursive methods provide solutions with controllable accuracy. Moreover, high-order algorithms allow also to control the number of steps to achieve this high accuracy. Computational complexity

of high-order algorithms necessitates simplifications. The algorithms are presented in this article as recursive procedures driven by estimation errors multiplied by the gain matrices. A simple incremental algorithm for update of the gain matrices is found, and the gain matrix is associated with Neumann series. Moreover, a new notion of infinite order algorithm (the limiting form) is introduced for quantification of the proximity between high-order estimator and perfect estimator. The results provide a new unification method for high-order algorithms with simplified structure, which makes algorithms implementable.

References

1. Ljung L. *System identification: theory for the user*. Upper Saddle River, NJ: Prentice Hall, 1999.
2. Ljung S and Ljung L. Error propagation properties of recursive least-squares adaptation algorithms. *Automatica* 1985; 21(2): 157–167.
3. Slock D. Backward consistency concept and round-off error propagation dynamics in recursive least-squares algorithms. *Opt Eng* 1992; 31(6): 1153–1169.
4. Liavas A and Regalia P. On the numerical stability and accuracy of the conventional recursive least squares algorithm. *IEEE Trans Signal Process* 1999; 47(1): 88–96.
5. Atkinson K. *An introduction to numerical analysis*. 2nd ed. New York: John Wiley & Sons, 1989.
6. Kelley C. *Iterative methods for linear and nonlinear equations*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1995.
7. Ye Q. Computing singular values of diagonally dominant matrices to high relative accuracy. *Math Comput* 2008; 77(264): 2195–2230.
8. Sauer T. *Numerical analysis*. 2nd ed. Upper Saddle River, NJ: Pearson, 2011.
9. Stotsky A. Recursive trigonometric interpolation algorithms. *Proc IMechE, Part I: J Systems and Control Engineering* 2010; 224(1): 65–77.
10. Stotsky A. Frequency determination in control applications: excitation-based approach. *Proc IMechE, Part I: J Systems and Control Engineering* 2012; 226(8): 1142–1148.
11. Bayard D. A general theory of linear time-invariant adaptive feedforward systems with harmonic regressors. *IEEE Trans Automat Contr* 2000; 45(11): 1983–1996.
12. Kelner J, Orecchia L, Sidford A, et al. *A simple, combinatorial algorithm for solving SDD systems in nearly-linear time*. Cornel University Library, Ithaca, NY, arXiv: 1301.6628v1 [cs.DS], 28 January 2013.
13. Soleymani F. A rapid numerical algorithm to compute matrix inversion. *Int J Math Math Sci 2012*; 2012: 134653 (11 pp.). DOI: 10.1155/2012/134653.
14. Chen K. *Matrix preconditioning techniques and applications*. Cambridge: Cambridge University Press, 2005.
15. Schulz G. Iterative Berechnung Der Reziproken Matrix. *Z Angew Math Mech* 1933; 13: 57–59.
16. Demidovich B and Maron I. *Numerical calculations*. Moscow: Fizmatgiz, 660 pp., 1963 (in Russian).
17. Ben-Israel A. An iterative method for computing generalized inverse of an arbitrary matrix. *Math Comput* 1965; 19(91): 452–455.
18. Krishnamurthy E and Sen S. *Numerical algorithms computations in science and engineering*. New Delhi, India: Affiliated East-West Press, 1986.
19. Li H, Huang T, Zhang Y, et al. Chebyshev-type methods and preconditioning techniques. *Appl Math Comput* 2011; 218(2): 260–270.
20. Codevico G, Pan V and Barel M. Newton-like iteration based on a cubic polynomial for structured matrices. *Numer Algorithm* 2004; 36(4): 365–380.
21. Li W and Li Z. A family of iterative methods for computing the approximate inverse of a square matrix and inner inverse of a non-square matrix. *Appl Math Comput* 2010; 215(9): 3433–3442.
22. Stotsky A. Harmonic regressor: robust solution to least-squares problem. *Proc IMechE, Part I: J Systems and Control Engineering* 2013; 227(8): 662–668.
23. Richardson L. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a Masonry Dam. *Philos T R Soc Lond* 1911; 210: 307–357.
24. Stotsky A. A new frequency domain system identification method. *Proc IMechE, Part I: J Systems and Control Engineering* 2012; 226: 111–124.
25. Stotsky A. Blade root moment sensor failure detection based on multibeam LIDAR for fault-tolerant individual pitch control of wind turbines. *Energ Sci Eng* 2014; 2(3): 107–115.
26. Horn R and Johnson C. *Matrix analysis*. Cambridge: Cambridge University Press, 1985.