

BassID: The Simplest [Yet Optimal (?)] Empirical System Identification Algorithm

by

Robert W. Bass

Innoventek, Inc. [www.innoventek.com]

Here I shall formulate a generic Empirical System Identification (**ESID**) algorithm which upon a particular Challenge Problem (that was proposed without consideration of any specific approach to ESID) gave an answer that was slightly more accurate (99.1% vs 99%) and also faster than its solution by the most excellent “subspace” approach known to me (namely Larimore’s Canonical Variate Analysis [CVA] embodied in his valuable **ADAPT**x program [1]).

I’m well aware that more ambitious approaches to system ID, such as based upon Maximum Likelihood Estimation (MLE), as developed by former students of A.V. Balakrishnan, namely L.W. Taylor at NASA Langley and K. Iliff & R. Maine at NASA Ames, eventually embodied in the MATLAB Toolbox **MMLE3** [2]), seek not only to identify the system’s parameters but also the biases and covariance matrices of process disturbances and sensor or measurement noises, and that this and the Prediction Error approach in L. Ljung’s MATLAB Toolbox [2], are capable of the ultimate in accuracy & precision; however, these computer-intensive approaches necessarily ([6],[9]) involve three limitations: (i) their use requires some expertise and is truly user-intensive, (ii) they inherently require solution of large-scale nonlinear systems of equations by iterative methods, which may get “hung up” on a local minimum and fail to converge; (iii) their extreme accuracy occurs only when the correct system dimension n is employed, which they sometimes have difficulty finding. Accordingly I’m here concerned solely with methods that employ strictly linear algebra and that (for generic systems as defined below) never fail.

One certain advantage of the present algorithm is that its derivation can be understood by an undergraduate engineering student who merely knows matrix algebra, whereas comprehending the numerous published “subspace” approaches seems to me to be impossible without prior graduate work in both advanced statistics and abstract functional analysis (e.g. Hilbert Space analysis).

An even greater advantage is that it leads to a **RECURSIVE** algorithm in which each newly arriving pair of Input-Output signal vectors (in the Multiple-Input Multiple-Output [MIMO] case) provides small corrections to the previously-attained ID, which on average improve until no further improvement is possible (conditioned upon how much stochastic corruption existed in the original IO data).

The present work will doubtless strike experts in the field as “naïve” and “amateurish” but I am willing to learn that I am mistaken, in assertion of its seeming **optimality**, from anyone who is willing to engage in a “trial by numerical contest” to be supervised by a truly neutral third party IF such advocate of a more conceptually sophisticated approach prevails. [Experts: consult Appendices 1-4 below for further discussion.]

The **ESID problem** which I address concerns the discrete-time generic problem of an ID of a Linear Time Invariant (LTI) system, specified by a triad \mathcal{S} of unknown matrices (A,B,C) , of sizes respectively $n \times n$, $n \times m$, $l \times n$ where $l \leq n$, $m \leq n$. This system is assumed to be **generic** in the sense that, taking I_n to denote the $n \times n$ identity matrix, all of the poles of $(zI_n - A)^{-1}$ are inside the unit circle of the complex-frequency plane, i.e. if $\det(zI_n - A) = 0$, then $|z| < 1$, and, furthermore, the pair (A,B) is assumed to satisfy Kalman’s criterion of Controllability and the pair (A,C) is assumed to satisfy his criterion of Observability, namely, if T denotes vector-matrix transposition, and if $C := [B, A \cdot B, A^2 \cdot B, \dots, A^{n-1} \cdot B]$, $O := [C^T, A^T \cdot C^T, (A^T)^2 \cdot C^T, \dots, (A^T)^{n-1} \cdot C^T]$, then $\det(C \cdot C^T) > 0$ & $\det(O \cdot O^T) > 0$.

It is well known [3] that for such Bounded-Input \rightarrow Bounded-Output (**BIBO-stable**) systems there must exist positive numbers $\gamma \geq 1$ and $\lambda = \max(|z_i|) < 1$ such that for all positive integers $k \in \mathbb{Z}$ it is true that

$\|A^k\| \leq \gamma \cdot \lambda^{k-1}$, ($k = 1, 2, 3, \dots$); consequently sufficiently large powers of A are numerically negligible and may be replaced by the $n \times n$ null matrix in computational algorithms (the **key fact** innovatively exploited herein).

Thus we consider a generic system S of the form

$$x^{k+1} = A \cdot x^k + B \cdot u^k, \quad (1)$$

$$y^k = C \cdot x^k, \quad (2)$$

where there are N time-epochs $k = 1, 2, 3, \dots, N$. Here the m -vectors u^k represent the **input**-signal sequence $\{u^k\}$, and the l -vectors y^k represent the **output**-signal sequence $\{y^k\}$. Typically the input signals represent *actuator* commands and the output signals are the results of *sensor* measurements. Moreover we assume that the unknown system or “black box” to be identified is started from **quiescence**, so that $x^1 = 0$ & $y^1 = 0$, i.e. the **given data** consists of the matrices U & Y of dimension respectively $m \times (N-1)$ & $l \times (N-1)$ defined as

$$U = [u^1, u^2, u^3, \dots, u^{N-1}], \quad Y = [y^2, y^3, y^4, \dots, y^N], \quad (3)$$

where (contrary to what might be assumed at first glance) NEW data would consist of the pair $[u^N, y^{N+1}]$.

An initial formulation of the **ESID problem** might be: given the Input-Output (IO) data $[U, Y]$, find the ID-triad $S = (A, B, C)$. However it turns out to be essential to reformulate the problem (as did Kalman’s student B.L. Ho in the all-important Ho-Kalman algorithm [3, p. 201],[4],[5],[6, pp. 276-280]) in terms of the **Markov Parameters** $\{M_k\}$, defined as

$$M_k = C \cdot A^{k-1} \cdot B, \quad (k = 1, 2, 3, \dots, 2n), \quad (4)$$

where use has been made of what seems to be the inadequately appreciated fact that only the first $2n$ of the $\{M_k\}$ need to be used in order to ID the triad S perfectly. In fact, the author will gladly send to anyone who requests a copy from him at <donquixote@innoventek.com> the .txt-file versions of two MATLAB functions $[H, zH] = \text{getHankelMatrix}(n, MRKV)$ & $[A, B, C] = \text{HoKalmanHankel}(n, H, zH)$, wherein $MRKV$ denotes the $\{M_k\}$ arrayed as an $l \times m \times 2n$ tensor, which IMHO (subject to correction if need be!) leave nothing to be desired in the way of algebraic parsimony, numerical robustness, and rapidity of execution. (When I use double-precision arithmetic on S in (4), the resultant S recovered from $MRKV$ via the Ho-Kalman algorithm agrees to double-precision with the initial S .) Accordingly the **genuine problem** [assuming n known, or to be ascertained pragmatically via trial-and-error] is: given the data $[U, Y]$, find $MRKV$.

It is well-known [3], and easy to prove by elementary induction, that for $k = 2, 3, 4, \dots, N$,

$$(y^k)^T = (u^1)^T \cdot (M_{k-1})^T + \dots + (u^{k-1})^T \cdot (M_1)^T, \quad (5)$$

where, in moving from left to right, the indices of the u ’s are increasing and those of the M ’s are decreasing. This is just a restatement of the familiar fact regarding LTI systems started from quiescence that the output sequence comprises a convolution of the input sequence with the system’s **impulse response**, which in this case is just the complete set of all $\{M_k\}$ for arbitrary k up to $N-1$. From (5) one may construct the following system of **determining equations**. Let

$$M := [M_1, M_2, \dots, M_{N-1}] \quad (6)$$

and then deduce from (5) that

$$Y^T = U \cdot M^T, \quad (7)$$

where the first column of the block-matrix U is U^T and the 2nd column is formed by replacing u^1 in U by $0 \cdot u^1$ and then replacing u^2 by u^1 and continuing on in this manner until replacing u^{N-1} by u^{N-2} , and then transposing the result, and continuing thusly to shift the u ’s to the right until the final column of U consists of the transposition of the matrix obtained by replacing every entry in the definition of U in (3) by $0 \cdot u^1$ except for u^{N-1}

which is replaced by u^1 . In other words, U is a lower-triangular block-matrix which has only repetitions of the row-vector $(u^1)^T$ down its main diagonal, and only null-matrices of size $1 \times m$ above the main diagonal.

The next and crucial step is to recognize that for $k > K \geq 2n$, where $K \& J := m \cdot K \leq (N-1)$ are suitably chosen, we may ignore M_k and in (7) replace M by $M_{\text{cut}} := [M_1, M_2, \dots, M_K]$, while simultaneously cutting out all but the first J actual columns of U to form the $(N-1) \times J$ matrix U_{cut} . Accordingly the effective determining equations are now

$$Y^T = U_{\text{cut}} \cdot (M_{\text{cut}})^T, \quad (8)$$

where, by inspection, the bottom row of U_{cut} consists of $[(u^{N-1})^T, (u^{N-2})^T, \dots, (u^k)^T]$. Finally define the symmetric non-negative definite $J \times J$ matrix

$$P := (U_{\text{cut}})^T \cdot U_{\text{cut}}, \quad (9)$$

and invoke the standard hypothesis (which is universally required, and without which NO known ESID algorithm succeeds) that *the input U is “sufficiently richly exciting”* in my new sense that $P = P^T > 0$. Although there are elegant characterizations of “sufficiently richly exciting” none of them can be implemented without in effect completing the ID first, so one needn’t apologize for not being able to assert before step (9) that the present ID procedure will achieve success, which consists of using the matrix left divide (*ml*) or “backslash operator \backslash ” in MATLAB to perform a highly-developed and numerically robust Gaussian elimination procedure to yield

$$M_{\text{cut}} = (P \backslash ((U_{\text{cut}})^T \cdot Y^T))^T. \quad (10)$$

Now extract from M_{cut} the first $2n$ of the $\{M_k\}$, which is possible because $2n \leq K$, and the ID is complete. In case P turns out not to be positive-definite, the failure of (10) is NOT to be construed as a failure of the present approach, but rather a failure of the “experiment design” in which the user-selected input commands U were not sufficiently well-chosen as to excite ALL of the system’s modes sufficiently robustly.

The reader may object that no explicit attention has been paid to the question of how additive stochastic disturbances on the RHS of (1) & (2) affect the precision of the new algorithm (8)-(10). My reply is two-fold:

Firstly, the solution (10) is effectively an Ordinary Least Squares (OLS) solution of an over-determined system of linear equations (8) and it is well known that OLS provides a certain resistance to errors in the data in the sense that as said errors become smaller the solution (10) tends asymptotically toward the theoretically ideal solution [see Appendix 3 for a proof]. Indeed if the problem (8) be expressed in a simpler notation as

$$Y^T = U \cdot M^T, \quad (11)$$

then the addition of zero-mean Gaussian stochastic processes dY & dU to the IO data $[U, Y]$ merely results in replacement of the preceding deterministic problem by the stochastic problem

$$(Y + dY)^T = (U + dU) \cdot M^T, \quad (12)$$

whence by means of the expectation operator \mathcal{E} , wherein $\mathcal{E}\{Y\} = Y$, $\mathcal{E}\{dY\} = 0$, $\mathcal{E}\{U\} = U$, $\mathcal{E}\{dU\} = 0$, the usual arguments [6]-[9] may be applied as in Appendices 1-2 below to prove that (10) does provide an **unbiased minimal-variance** estimate of the unknown matrix of Markov Parameters M in the commonly-assumed [7], [8] though arguably oversimplified case wherein the input commands U are perfectly known and so $dU \equiv 0$ is postulated. For the more general case, see Appendices 2-3 below.

Secondly, in numerical experiments I have added an 0.1-percent variance Gaussian white noise disturbance to the input-command data U used to generate the output data Y , and then further corrupted this output by an additional similarly-scaled white noise simulation of sensor-measurement-error, and yet the solution (10) continued to exceed 99% accuracy in a significant “Challenge Problem” with $l = 2$, $m = 3$, $n = 5$, $N = 300$.

Now suppose that the initial batch-process (8)-(10) has been performed and it is desired to switch to a **recursive** algorithm (which can be executed in real time online/onboard, enabling Adaptive Control based upon an ID which changes as evolving environmental conditions render the initial LTI linearization invalid).

Henceforth we shall “freeze” the integers J & K and therefore omit the subscripts on U & M and shall replace P by P .

If the reader writes down the preceding derivation (8)-(10) in more painstakingly explicit detail, and then asks how to continue it when new IO data-vectors $u_{\text{new}} = u^N$, & $y_{\text{new}} = y^{N+1}$ are received, and then subtracts off from this result the previously attained $Y^T = U \cdot M^T$ and then applies what Kalman has called the Bass Lemma (a special case of the Matrix Inversion Lemma [8, p. 428], [9, p. 225]) for inversion of the sum of a non-singular matrix plus a dyad, it will be perceived by inspection that the definitions

$$M_{\text{new}} = M + dM, \quad P_{\text{new}} = P + dP, \quad (13)$$

can be implemented by the following remarkably simple algorithm: firstly compute the $J \times 1$ vector

$$\mathbf{u}_{\text{new}} := [(u_{\text{new}})^T, U_{\text{bot}}]^T, \quad U_{\text{bot}} := [(u^{N-1})^T, (u^{N-2})^T, \dots, (u^{K-1})^T]. \quad (14)$$

where U_{bot} is the bottom row of U after excision of its right-most block, and the **prediction error**

$$dy := y_{\text{new}} - M \cdot \mathbf{u}_{\text{new}}, \quad (15)$$

which by (5) is self-evidently a null-vector if the ID of M has been perfect. Next, define the J -vector

$$\mathbf{v} := P \setminus \mathbf{u}_{\text{new}} \quad (16a)$$

and the positive scalar

$$\alpha := 1/(1 + \mathbf{v}^T \cdot \mathbf{u}_{\text{new}}), \quad (0 < \alpha < 1), \quad (16b)$$

and complete the update-procedure by means of

$$dM = \alpha \cdot \mathbf{v} \cdot dy, \quad dP = -\mathbf{u}_{\text{new}} \cdot (\mathbf{u}_{\text{new}})^T. \quad (17)$$

The author has implemented the preceding **novel recursive ID algorithm** (13)-(17) in MATLAB and tried it on the aforementioned Challenge Problem, with the gratifying success exhibited in the following 5 Figures. (For comparison with known RLS algorithms of this type, see Appendix 4 below.)

The original version of said problem already had 0.1 percent disturbances & noises added to the IO data before application of the algorithm (8)-(10), with the result that after using the first 299 samples of the $N = 300$ IO data-vectors the relative errors in (A, B, C) , i.e. (dA, dB, dC) were

$$\|dA\|/\|A\| = 0.0097, \quad \|dB\|/\|B\| = 0.0098, \quad \|dC\|/\|C\| = 0.0072, \quad (18)$$

or better than 99% accuracy. But from the first of the following Figures, it is clear that even using the very last sample of IO data together with the preceding algorithm results in a further improvement of the ID!

Obviously if there is no mistake in the algebra leading to (13)-(17), then use of the batch procedure up to the point (10) while using only the first $2 \cdot m^2 \cdot n = 90$ epochs of IO data and then recursive application of (17) for the remaining $N - 90 = 210$ epochs should lead to an identical result as (18), although it actually did a trifle better, as can be seen in Figure 3 which resulted in:

$$\|dA\|/\|A\| = 0.0094, \quad \|dB\|/\|B\| = 0.0098, \quad \|dC\|/\|C\| = 0.0071. \quad (19)$$

As another test, the author added altogether unrealistic additional disturbances & noises providing an **extra 10%** corruption. Of course the initial ID using the first 90 samples then gave a completely unacceptable error of more than 100% in the ID of (A, B, C) , but now consider what happened to the errors (dA, dB, dC) when the preceding simple algorithm was applied recursively to the remaining 210 samples, as depicted in the second

of the Figures. Even in the presence of this enormously unrealistic corruption, the preceding novel recursive algorithm was sufficiently powerful to reduce the ID error by 60%, i.e. from more than 100% to less than 40%!!

Finally, in the most stringent test imaginable, the author wondered if the new definition (15) of **System ID Error for negative feedback** were powerful enough to “bootstrap” into a successful ID even if the initial M in (15) were taken to be an appropriately-dimensioned null matrix and the initial P in (16) were taken to be the $J \times J$ identity matrix I_J multiplied by 10^{-6} at the initiation of the recursion, which resulted in the amazing result displayed in Figures 4-5 and, after literally starting at scratch, the final errors

$$\|dA\|/\|A\| = 0.0128, \quad \|dB\|/\|B\| = 0.0089, \quad \|dC\|/\|C\| = 0.0057. \quad (20)$$

Note that, because each recursively additive dyadic update dP to P is non-negative definite, the norm of P cannot decrease, so that as N increases without limit the corrections to the ID may eventually start to decrease with the result that no further improvement is possible, but under the circumstances such a limiting steady-state condition is to be expected.

References

- [1] See <http://www.adaptics.com> &/or contact Dr. Wallace E. Larimore, President, Adaptics Inc., 1717 Briar Ridge Road, McLean, VA 22101 USA, TEL: 703 532-0062 FAX: 703 536-3319 EMAIL: larimore@adaptics.com .
- [2] See www.mathworks.com/products/matlab/ re **MMLE3 & System Identification Toolbox**.
- [3] W.J. Rugh, *Linear System Theory*, Prentice-Hall, 1996 (2nd ed.).
- [4] R.E. Kalman, P.L. Falb & M.A. Arbib, *Topics in Mathematical System Theory*, McGraw-Hill, 1969
- [5] H. Paul Zeiger & Amber McEwen, *Approximate Linear Realizations of Given Dimension Via Ho's Algorithm*, University of Colorado Dept of Computer Science, 1973, available online as CU-CS-013-73, from www.cs.colorado.edu/departments/publications/reports/r000.html 51k.
- [6] Jer-Nan Juang & Minh Q. Phan, *Identification and Control of Mechanical Systems*, Cambridge Univ. Press, 2001
- [7] M.B. Tischler & R.K. Remple, *Aircraft and Rotorcraft System Identification: Engineering Methods with Flight Test Examples*, AIAA 2006 {<http://www.aiaa.org> TEL: 800 682-2422}.
- [8] V. Klein & E.A. Morelli, *Aircraft System Identification: Theory and Practice*, AIAA 2006 (*ibid*)
- [9] R.V. Jategaonkar, *Flight Vehicle System Identification: A Time Domain Methodology*, AIAA 2006 (*ibid*)
- [10] R.W. Bass, *Discrete-Time Rhobustness*, 2006 (preprint available from author at donquixote@innoventek.com)
- [11] S. Van Huffel & J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM, 1991 (<http://www.siam.org/books/>).
- [12] R.W. Bass, invited *Review* of M.K. Grewal & A.P. Andrews, *Kalman Filtering: Theory & Practice*, Prentice-Hall, 1993, in *IEEE Trans.Aut.Control*, vol. 40, Nov. 1995 & also in *IEEE Proceedings*, vol. 84, No. 2, Feb. 1996, pp. 321-324.

Acknowledgement

It is a pleasure to thank DARPA Program Officer, Dr. Benjamin Mann, and the Project Monitor, Dr. Douglas Drake of Strategic Analysis Inc., for their generous helpfulness and willingness to consider my proposed Stochastic Generalization of the Ho-Kalman Algorithm, even though the “stochastic” aspect comprises little more than application of OLS to a novel but basically deterministic formulation, whose principal innovation (8) was pursuit of maximal algebraic parsimony (namely if A be assumed to have Companion Matrix form, then the number of unknown parameters in S equals $(1 + l + m) \cdot n = 30$ in the cited example, while the above $K \geq 2n$ & $J = m \cdot K \geq 2m \cdot n = 30$ turns out fortuitously to have the smallest size of J that could have possibly succeeded [though in other examples one may require $J \gg 2m \cdot n$ in order to have more equations than unknowns &/or to ensure that, for $k > K$, $\|A^k\|$ is sufficiently small to be ignored]). The algorithms implemented in the MATLAB functions mentioned above, which I shall place in the public domain, were developed under DARPA Purchase Order HR-0011-07-P-0006, hereby gratefully acknowledged.

Appendix 1

The statistically-oriented reader may object to the author's apparently cavalier attitude regarding the precise assumptions upon the stochastic processes dY and dU , but this is not a result of ignorance but rather of literally a half-century's experience with the actual interaction between Control & Estimation Theory and real-world Engineering Practice upon which I base the following several *caveats*.

In actual engineering applications of Kalman Filtering to estimate state-vectors online in real-time, in order to implement Optimal/Robust Control laws based upon state-vector feedback ([3}, [6]), it turns out that exact knowledge of the process disturbance covariance Q and measurement noise covariance R is usually irrelevant, because their use leads (as was repeatedly demonstrated emphatically by my late Hughes Aircraft Co. colleague Isaac Horowitz) to a numerically *fragile* rather than a *robust* state-estimator. In fact, Kalman [4, p. 55], credits me in 1963 with introducing the concept of an *asymptotic state-estimator*, in which the noises & disturbances are taken to be *deterministic but unknown*, and the objective is to synthesize an error-feedback gain-matrix K that will robustly minimize the future *rms* of said state-estimation error. By Kalman's celebrated Duality Principle, every such problem can be reformulated as a deterministic Regulator Problem, where the (Q,R,S) , with the usually ignored cross-variance S playing a vitally-essential role if stability robustness &/or fidelity robustness [10] is desired, have no relationship whatsoever to ANY statistical concepts although the identical algorithmic equations for the Kalman Filter and its gain-matrix $K = K(Q,R,S)$ finally result! Indeed, in engineering practice at Hughes Aircraft Co., McDonnell Douglas Astronautics, Litton G&CS, Rockwell Science Center, etc. what I observed was that the practicing engineers had to resort to "tuning the KF" by selection of artificial values of (Q,R,S) unrelated to any statistical concepts in order to design practically useful systems! This resulted in my IEEE papers on *robust tuning* of KFs (cited in [10]) which were used in actual practice with my colleague Dean Zes at Hughes Helicopters, with my colleague Joseph Ratkovic at Litton Data Systems and with my colleague Dan Hill at Rockwell.

Furthermore I am an admirer of C.D. Johnson's concept of Disturbance Accommodating Control (DAC) in which the disturbances & noises are assumed to be deterministic but unknown "waveforms" and means such as the EKF used to identify at least their lower-frequency components in real-time and modify the control feedback laws in such a manner as to nullify their presence (almost as if they were known precisely and could be subtracted from the system's dynamical equations by appropriately-biased control laws!).

In addition, many of the most-often used statistical paradigms pertaining to System Identification can be re-interpreted as merely Weighted Least Squares (WLS) or Total Least Squares (TLS) applied to systems wherein the corruptions (dY,dU) are not assumed to be stochastic processes but merely uncertain or unknown deterministic signals. For further discussion of this see the following Appendix.

Appendix 2

In an informative discussion of the various possible attitudes toward the problem (12) which are present in the technical literature, Klein & Morelli [7, p.79] discuss the **Bayesian, Fisher, and Least Squares** interpretations of (12) in the simplified case that $dU \equiv 0$.

In the present notation, we may assume that $\mathcal{E}\{dy_p^i \cdot dy_q^j\} = \sigma_p^2 \cdot \delta_{ij} \cdot \delta_{pq}$, where δ_{ij} denotes the Kronecker delta-function, whence, setting $\sigma_y^2 := \sigma_1^2 + \sigma_2^2 + \dots + \sigma_\ell^2$ we have $\mathbf{R} := \mathcal{E}\{dY^T \cdot dY\} = \sigma_y^2 \cdot \mathbf{I}_{N-1}$.

In the *Bayesian* approach, one assumes not only *a priori* knowledge of the prior mean M_p of the stochastic process M , but also assumes that its covariance $\mathbf{\Gamma}$ is known. In this case, the conditional expectation of M given $(M_p, \mathbf{R}, \mathbf{\Gamma})$ can be shown, by further elaboration upon the cited introductory definition in [7], to be

$$M^T = (\mathbf{U}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{U} + \mathbf{\Gamma}^{-1})^{-1} \cdot (\mathbf{U}^T \cdot \mathbf{R}^{-1} \cdot Y^T + \mathbf{\Gamma}^{-1} \cdot M_p^T).$$

which by inspection can be seen to be merely a weighted combination of the WLS solution (given below) together with the assumed prior knowledge M_p .

In the *Fisher* approach, one derives a Maximum Likelihood Estimator (MLE) which turns out to be

$$M^T = (\mathbf{U}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{U})^{-1} \cdot \mathbf{U}^T \cdot \mathbf{R}^{-1} \cdot Y^T,$$

that is, the same as assuming that the prior knowledge (M_p, Γ) is absent or ignored.

In the deterministic Weighted Least Squares (WLS) approach, one multiplies the error Z in (11), namely

$$Z := Y^T - U \cdot M^T,$$

by an arbitrary positive-definite weighting matrix R^{-1} , chosen by engineering “judgment” [8, p. 82] to define a weighted *error* $E := R^{-1} \cdot Z$ and then applies ordinary differential calculus to find the minimum of the quadratic form

$$\Phi := E^T \cdot \Theta \cdot E,$$

where $\Theta = \Theta^T > 0$ is another arbitrary positive-definite weighting matrix. It turns out that if one chooses $\Theta = R$ then Φ is minimized by use of precisely the same formula given above for the MLE, except that now the measurement error dY has been assumed to be merely a deterministic uncertainty rather than a stochastic process. By inspection, the Ordinary Least Squares (OLS) solution is obtained by the choices $\Theta = R = \sigma_y^2 \cdot I_{N-1}$.

The reader may yet object (and to this I readily assent) that in real-life problems it is almost never realistic to ignore process disturbances dU in favor of the mathematically simpler case $dU \equiv 0$ just surveyed. Still, even in this case there is a useful deterministic alternative, known [11] as **Total Least Squares** (TLS). Here one employs the Frobenius norm of a matrix M , defined as $\|M\|_F := (\text{trace}[M^T \cdot M])^{1/2}$ and exploits the convenient fact that for arbitrary matrices M_1, M_2 it is true that $\text{tr}[M_1 \cdot M_2] \equiv \text{tr}[M_2 \cdot M_1]$ when both products exist. Using the Singular Value Decomposition (SVD), it is possible to find **TLS estimates** (Y_e^T, U_e) of (Y^T, U) which minimize

$$\| [Y_e^T, U_e] - [Y^T, U] \|_F$$

and then to use (Y_e^T, U_e) as if they are “cleaned” versions of (Y^T, U) in the OLS solution. (Note also Appendix 3 below.)

In short, if (from a physical viewpoint) one believes that it is more “scientific” and less ideological/doctrinaire to deal with *deterministic uncertainties* rather than to imagine hypothetical *probability distributions*, then it is altogether reasonable to apply elementary undergraduate calculus and matrix algebra in avoidance of the statistical esoterica and yet to derive the very same algorithmic procedures with far less conceptual background/baggage pertaining to evanescent/imaginary “distributions”!

Appendix 3

Admittedly, if one wishes to take a statistical viewpoint, it is advisable to assume, as in Appendix 2, that not only is each component of the sequence $\{dy^k\}$ comprised of white noise, i.e. independently-distributed zero-mean Gaussian processes such that $\mathcal{E}\{dY\} = 0$ & $R := \mathcal{E}\{dY^T \cdot dY\} = \sigma_y^2 \cdot I_{N-1}$ but also that $\mathcal{E}\{du_p^i \cdot du_q^j\} = \sigma_p^2 \cdot \delta_{ij} \cdot \delta_{pq}$ whence, setting $\sigma_u^2 := \sigma_1^2 + \sigma_2^2 + \dots + \sigma_m^2$ we have, similarly, $Q := \mathcal{E}\{dU^T \cdot dU\} = \sigma_u^2 \cdot I_{N-1}$. It is furthermore reasonable to assume that corruptions to the IO vectors are uncorrelated, so that e.g. $\mathcal{E}\{dU^T \cdot dY\} = 0$ is a null matrix of dimension $J \times l$.

But this is insufficient to enable progress, because of the lower-block diagonal *structure* of U & dU . By painstakingly detailed examination of said structure, left as an exercise for the reader, one finds that

$$\Xi := \mathcal{E}\{dU^T \cdot dU\} = \sigma_u^2 \cdot \text{diag}(N-1, N-2, \dots, N-J) \cong (N-J) \cdot \sigma_u^2 \cdot I_J,$$

where in the final approximating equality we have assumed that $0 < J \ll N$. Now, as in the Fisher approach, assume that M is an unknown constant, and employ the matrix differential calculus as explained above in connection with the Frobenius norm, to derive the variance-minimizing result that the Best Linear Unbiased Estimate (BLUE) of M is given by

$$M^T = (U^T \cdot R^{-1} \cdot U + \Xi^{-1})^{-1} \cdot U^T \cdot R^{-1} \cdot Y^T, \quad R = \sigma_y^2 \cdot I_{N-1}, \quad \Xi = (N-J) \cdot \sigma_u^2 \cdot I_J,$$

where, once again, the fine details of the derivation are left as an exercise for the interested reader, but the final result proves our earlier assertion that as $(\sigma_y^2 + \sigma_u^2) \rightarrow 0$ the algorithms (10) & (13)-(17) have asymptotic consistency.

Appendix 4

The novel recursive algorithm (13)-(17) was first derived in a preliminary draft of this paper submitted to the DARPA Program Officer & Project Monitor named in the above Acknowledgement prior to my having learned of the existence of references [6]-[9], as a result of which I find in Jategaonkar [9, p. 222] and Klein & Morelli [8, p. 264] algebraically identical Recursive Least Squares (**RLS**) algorithms which, after minor manipulations involving mainly matrix inversions and elementary rearrangements, are identical to my allegedly “novel” algorithm (13)-(17). So must I plead guilty to having “rediscovered the wheel”? In terms strictly of pure algebra, disregarding robustness as considered in Numerical Analysis, the answer must be “YES, albeit a differently-spoked wheel!” However, it should be recalled that if the late Gerald Bierman had not discovered the “Square-Root Filtering” version of the Kalman Filter then [in Potter’s MIT Draper Lab independently-discovered version of “square-root filtering without square-roots”] without it the first manned soft lunar-landing would have crashed! This is because the subtraction of non-negative definite matrices from a matrix whose positive-definiteness is required for numerical success can lead, via round-off & truncation errors (as the late J. von Neumann noted in a paper on finite-arithmetic matrix-inversion which he personally handed me in 1950 while showing me the world’s first large stored-program digital computer at the Princeton IAS) to a theoretically positive eigenvalue becoming numerically null or negative, with resultant catastrophic numerical errors. But the previously published RLS algorithms all deal with repeated subtractions of dyads from P^{-1} rather than, as in my new algorithm (13)-(17), the successive additions of dyads to P . Consequently my differently-spoked wheel can justly be described as “admittedly a wheel, yet an innovative puncture-proof wheel!”

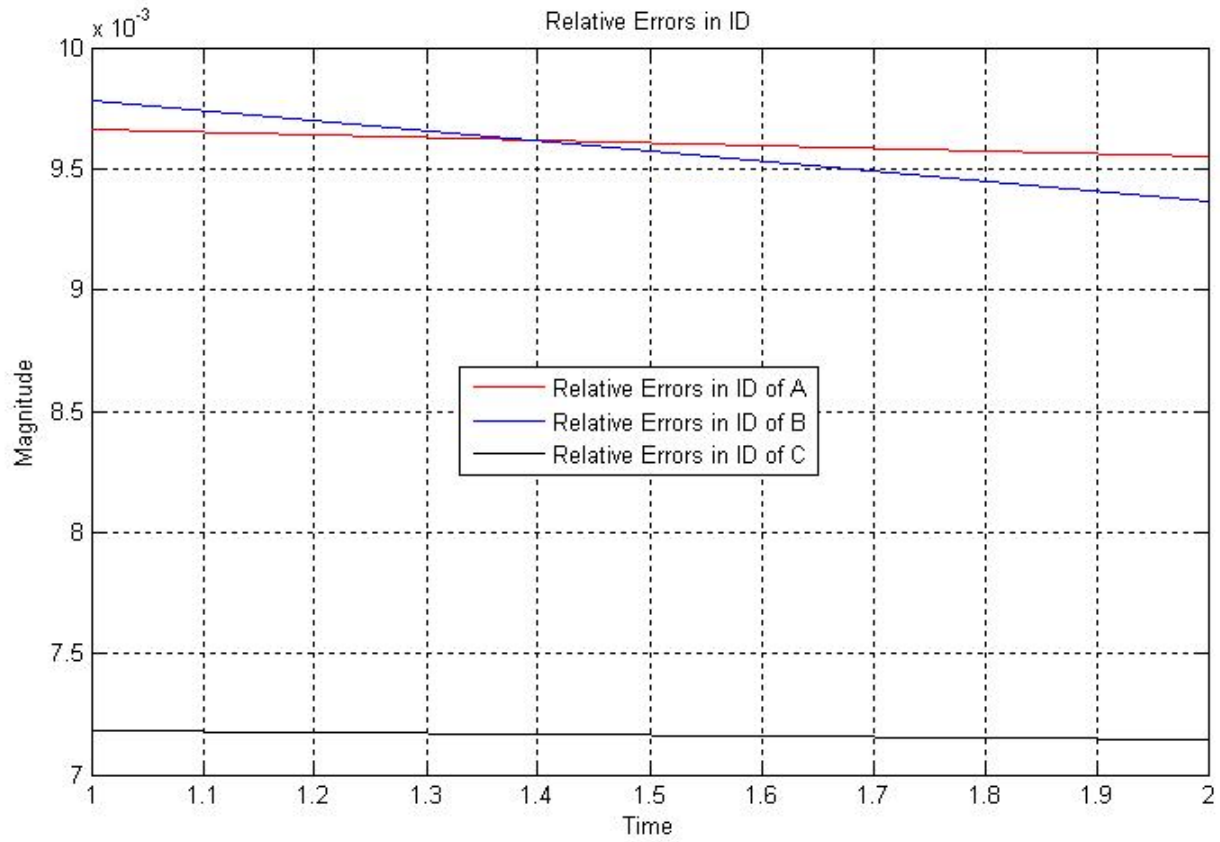


Figure 1

Illustration of the application of one step of the new recursive algorithm to the ID obtained by the batch-process algorithm using $N-1 = 299$ samples of the IO-data in a case wherein the batch-process had already attained an ID of better than 99% accuracy.

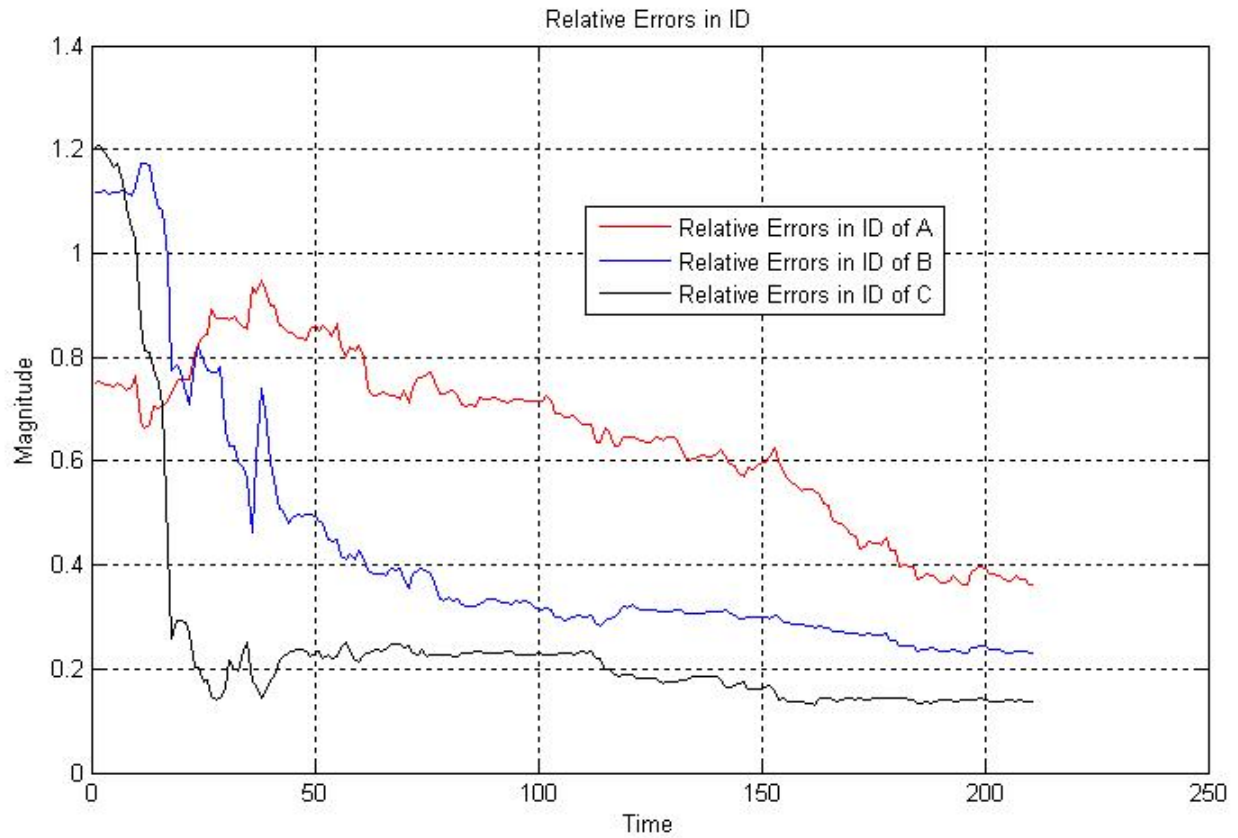


Figure 2

Illustration of the application of the new recursive algorithm 210 times to the initial ID obtained from the application of the batch-process ID to just the first 90 IO-data samples of the Challenge Problem with $N = 300$ samples, wherein the IO data had first been corrupted by the addition of an enormously unrealistic EXTRA **10%** disturbances/noises in addition to that already used in both the preceding and following illustrations.

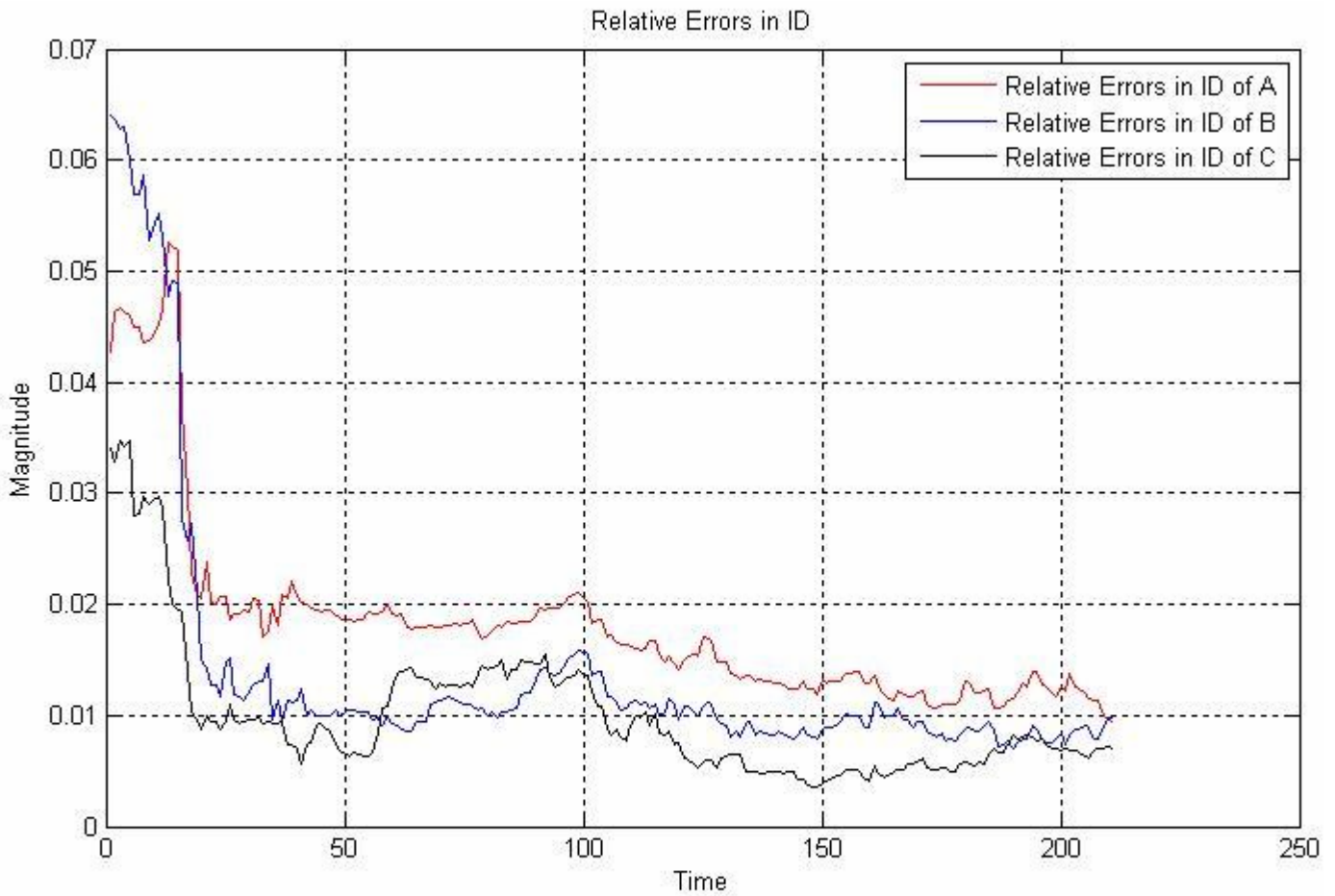


Figure 3

Illustration of the application of the new recursive algorithm applied 210 times to the initial 93% accurate ID obtained from the application of the batch-process ID to just the first 90 given IO-data samples of the Challenge Problem with $N = 300$ samples, illustrating that an identical final result of a trifle better than 99% accuracy is also obtained (as if the batch procedure had been applied simultaneously to all $N = 300$ samples of IO data!).

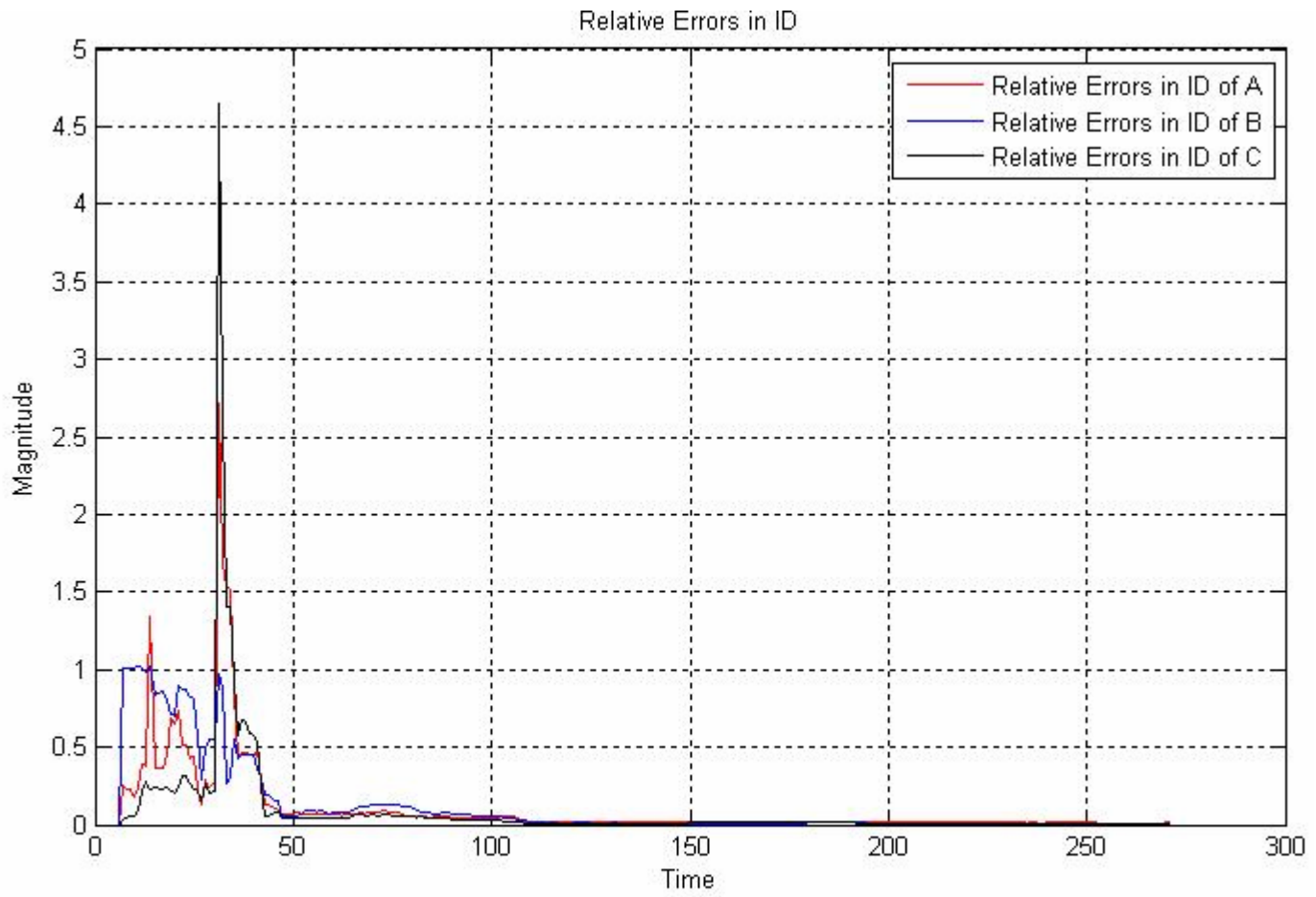


Figure 4

Illustration of the application of the new recursive algorithm 270 times with NO use whatsoever of the batch procedure; instead, the initial value of M in (15) was taken to be an appropriately-dimensioned null matrix and the initial P in (16) was taken to be the $J \times J$ identity matrix I_J multiplied by 10^{-6} at the initiation of the recursion. Note that after flailing around for about 50 recursions, starting from scratch, the P matrix had been built up to the point wherein strongly negative error feedback was introduced and although overshoots & undershoots continued to occur the over-all trend was decisively that to be expected from genuinely negative error-feedback! This Figure needs to be magnified from the 50th iteration onward in order to show that the final result is the same ~99% accuracy as had been obtained from a strictly batch procedure!

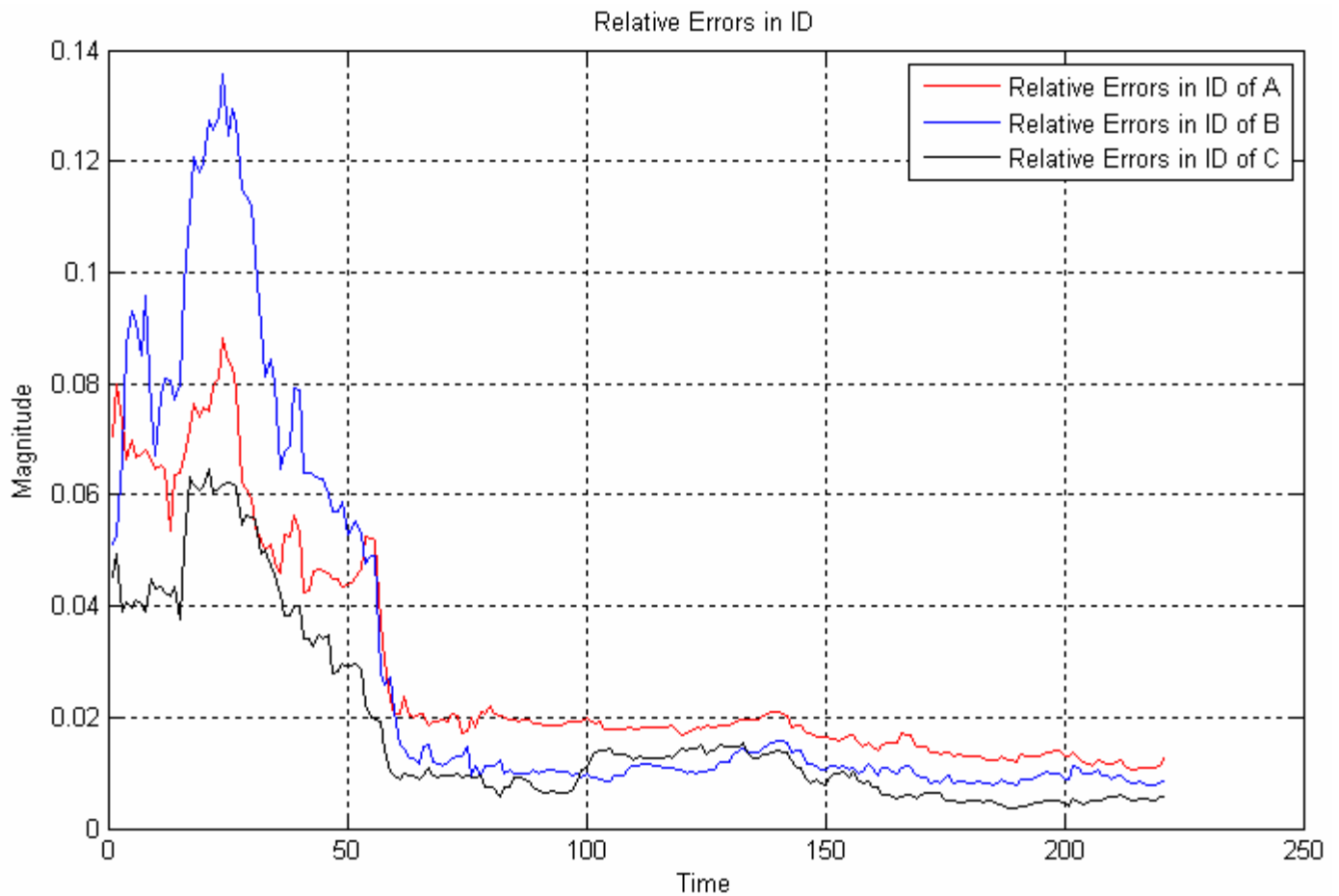


Figure 5

Magnification of final 220 iterations of the 270 iterations presented in Figure 4, showing that the new recursive algorithm (13)-(17) is so powerful that it can produce virtually the same 99% accurate ID as the batch procedure (10) even when started from scratch (after collecting the first $2.n.m = 30$ required IO-data samples) with no other initial assumptions nor starting-data whatsoever!!