

MILITARY TECHNICAL COLLEGE
CAIRO - EGYPT7th INTERNATIONAL CONF. ON
AEROSPACE SCIENCES &
AVIATION TECHNOLOGY

Recursive Identification Methods

Gamal A. El-Sheikh*

Abstract:

Since the dynamics of real systems are either time varying or nonlinear in nature, the self-tuning controller design is appropriate where a recursive identification algorithm is utilized to monitor or track the process behaviour. The systems identification algorithms are utilized to estimate the parameters representing the dynamics of the underlying process in linear, nonlinear or both forms, depending upon the required analysis. Therefore, the controller parameters can be tuned on line by reference to the periodically updated plant model. Ideally, the controller synthesis should optimize plant performance taking the uncertainty in the model into account as well as the performance criterion. However, for simplicity, the plant model is usually considered accurate during control synthesis yielding the *certainty equivalence self-tuner*. Self-tuning control combines the design of feedback controllers based on plant system models with the on-line estimation of the model parameters or the controller parameters using input and output data measurements. The robustness or alertness of self-tuning is totally dependent upon the robustness of the system identification algorithm and its ability to detect and track rapid changes in system performance. The identification techniques find its way into aerospace applications through identifying the flight parameters, adaptive control or adaptive signal and image processing. For example, it could be used to identify the attitude of an aircraft or a missile, engine parameters like engine load variations, fluctuations in fuel supply, and aerodynamic effects from the airstream that change considerably, ... etc. The subject of system identification has extremely diverse approaches depending upon the purpose of identifier and the underlying process to be identified. The choice between them depends on some factors among them are: model complexity, noise-to-signal ratio, convergence rate, and computational expenses. The most popular and easiest method for system identification is the least squares (LS) approach. Therefore, this paper presents some of these algorithms in a simplified way leading to the equations that must be solved through the specified algorithm, without going deeply into the derivation of each algorithm. Then, one of them is numerically implemented/evaluated for some examples. In addition, it highlights the identifiability which is a joint property of an identification experiment and a model and establishes the environment for adequate parameters estimates from the injected signal point of view yielding what is called the persistency of excitation. The results showed good convergence, fast response and good tracking utilizing some of the mechanisms or techniques that improve the numerical robustness of any identification algorithm.

Keywords: Systems Identification, Recursive Techniques, Adaptive Control.

1- Introduction

Self-tuning control combines the design of feedback controllers based on plant system models with the on-line estimation of the model parameters or the controller parameters using input and output data measurements. There is an intriguing interconnection between the control law selection and the identification rule operating simultaneously in closed loop. Many practical applications of control system design based on input-output measurements permit the repeated application of a system identification procedure operating in closed loop data together with successive refinement of the designed controller. The *a priori* assumption for self-tuning design is that the underlying system parameters are unknown and constant or slowly varying. However, the robustness or alertness of self-tuning is totally dependent upon the robustness of the system identification algorithm and its ability to detect and track rapid changes in system performance. The detection of change in process performance can be done using the prediction error in conjunction with some statistical rules [6,7]. Tracking the detected rapid variation can be carried out using various techniques or mechanisms such as random walk, covariance resetting, constant covariance trace, exponential forgetting and directional forgetting [1,3,4,10,13,14,15,16,19].

The dynamics of real systems are either time varying or nonlinear in nature, and therefore the self-tuning controller design is appropriate where an identification algorithm is utilized to monitor or track the process behaviour. Linear self-tuning techniques have the potential practical advantage that they can cope with some

* Lecturer (B.Sc., M.Sc., Ph.D., MIEEE) in the Guidance Department, Military Technical College, Cairo, Egypt

nonlinear phenomena by retuning some parameters to create a new locally linear approximation. Moreover, in case of severe nonlinearities, a nonlinear self-tuning controller can be used where the nonlinearity is identified and then incorporated into the design process as described in the [6,7]. Self-tuning has three stages: validation phase, modeling task, and implementation of the designed controller. The modeling task concerns the construction of a mathematical representation of a system in a form suitable for design utilizing a system identification technique. The linear parametric approach for system identification is the subject of this paper.

Self tuning control is based on recursive identification in which a process model is to be constructed based on a limited set of data. The subject of system identification has extremely diverse approaches depending upon the purpose of identifier and the underlying process to be identified. The choice between them depends on some factors among them are: model complexity, noise-to-signal ratio, convergence rate, and computational expense. There is no recommendation for one approach upon the other, however there are some requirements [1,5,9,10,11] that should be considered by the designer:

1. System configuration, either open-loop or closed-loop.
2. Input or control signals from the viewpoint of amplitudes and spectra.
3. Processing of data, either off-line or on-line.
4. Signal filtering and conditioning.
5. Sampling interval, especially in real-time experiments, and the frequency of data measurement or collection in case of multi-rate systems.
6. Hardware requirements, either the computational capacity of the computer or interfacing with the real process under investigation.

Identifiability is a joint property of an identification experiment and a model. It establishes that the model parameters can be estimated adequately from the experiment. The signal requirements for identifiability are expressed in terms of the persistency of excitation. The conditions for persistency of excitation are those conditions concerning the signals in an identification experiment to ensure adequate excitation of the system dynamics. They effectively specify how many independent components must be present in the input signal. This number depends on the order of the model to be identified. All of controller complexity, non-zero reference input and output noise/disturbances help or improve closed-loop identifiability. With the reference input zero, the feedback may destroy the linear independence of the regressors in the regressor vector which causes singularity of the covariance matrix and unfeasibility of the identification method. In a self-tuning control, the recursive identification allows a controller to tune/adjust its parameters on line by reference to a periodically updated plant model. Ideally, the control synthesis should optimize plant performance taking the uncertainty in the model into account as well as the performance criterion.

Experiment design: Identification is not a matter of applying standard techniques in a specified way and getting guaranteed results. It is a untidy combination of a model and experiment, computation, analysis and revision. Every stage is uncertain, and common experience is that each identification exercise raises some new problem. This is a reflection not primarily of immaturity in identification techniques, but rather of the immense variety of dynamical behaviour, experimental constraints and purposes for modeling. For these reasons, designing a software package for identification is extremely difficult. Adequacy of input and output is usually justified by persistency excitation conditions. The input sequence $\{u(t)\}$ is persistently exciting (p.e.) of order p if no selection from every p successive samples was exactly linearly dependent. The frequency domain counterpart is that the input contains power at p or more frequencies. The persistency of excitation conditions can be used in identification or self-tuning experiments to warn against over-simple choices of input signals. The modest p.e. requirement is to have non-zero power at some minimum number of frequencies that may ensure asymptotic convergence but do not guarantee satisfactory finite-sample performance.

The most popular and easier method for system identification is the recursive least squares (RLS) approach because of its conceptual simplicity, ease of implementation and relatively fast convergence rate. However, the basic RLS has also a number of shortcomings such as poor numerical performance due to the expected numerical singularity in case of non-informative signals. One of the most successful algorithms for improved numerical performance is Bierman's UD factorization algorithm [2,8,19]. But, this algorithm is not widely used due to its complexity for interpreting and implementing. Therefore, this paper presents some of these algorithms in a simplified way leading to the equations that must be solved through the specified algorithm, without going deeply into the derivation of each algorithm. Then, one of them is numerically tested for some examples.

2- System description (model structure)

Real processes evolve continuously in time and are nonlinear in nature. No mathematical model can ever display every nuance of the behaviour of the real system that it is constructed to represent. However, the demand is to obtain a model that is satisfactory under given experimental conditions in pursuing a chosen experimental goal. Therefore, any chosen model is only conditionally valid and the preservation of the relevant conditions is crucial. The most commonly used parametric models associated with the self-tuning area are linear both in the data and the parameters, which eases considerably the problems of design, estimation and analysis. The recursive nature of self-tuning algorithms suggest that discrete time modeling is appropriate. In addition, the output of real systems is usually corrupted by disturbances. Therefore, a dynamical system can be described by the linear difference equation:

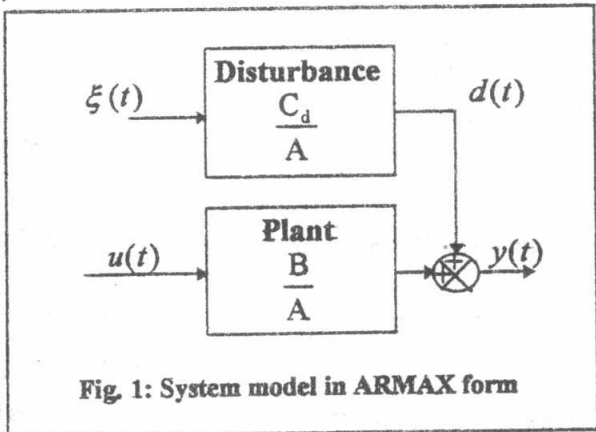


Fig. 1: System model in ARMAX form

$$y(t) = -\sum_{i=1}^{n_a} a_i y(t-i) + \sum_{j=0}^{n_b} b_j u(t-k-j) + d(t) \tag{1}$$

where, $u(t)$ represents the input signal or the control variable, $y(t)$ represents output signal from the system, $d(t)$ represents some disturbance acting on the system output, n_a represents the number of poles of the plant, and n_b is the number of zeros of the plant. Introducing the backward shift/delay operator z^{-1} defined by $z^{-1}y(t) = y(t-1)$, the system model can be expressed in the following discrete time transfer function form:

$$A(z^{-1})y(t) = B(z^{-1})u(t) + d(t) \tag{2}$$

where the B and A are polynomials in the unit shift operator z^{-1} given as:

$$A(z^{-1}) = 1 + a_1 z^{-1} + \dots + a_{n_a} z^{-n_a}, B(z^{-1}) = z^{-k} [b_0 + b_1 z^{-1} + \dots + b_{n_b} z^{-n_b}],$$

$n_a = \text{deg}(A)$, $n_b = \text{deg}(B) - k$; $b_0 \neq 0$, $k \geq 1$. Depending on the characteristics of $d(t)$, two cases of dynamical systems or models can be distinguished: Linear regression model and ARMAX model. In the linear regression model $d(t)$ has unspecified character and the model (2) can be written as

$$y(t) = \theta^T \phi(t) + d(t) \tag{3}$$

where the variable θ denotes the parameter vector formed as

$$\theta = [a_1 \ a_2 \ a_3 \ \dots \ a_{n_a} \ b_0 \ b_1 \ \dots \ b_{n_b}]^T \tag{4}$$

The variable ϕ represents the regression/signal vector defined as

$$\phi = [-y(t-1) \ -y(t-2) \ \dots \ -y(t-n_a) \ u(t-k) \ u(t-k-1) \ \dots \ u(t-k-n_b)]^T \tag{5}$$

In the ARMAX model $d(t)$ is modeled as a moving average (MA) of a serially uncorrelated white noise sequence $\zeta(t)$ i.e. $d(t) = C_d(z^{-1})\zeta(t)$, where $C_d = 1 + c_1 z^{-1} + c_2 z^{-2} + \dots + c_{n_c} z^{-n_c}$ and $n_c = \text{deg}(C_d)$. This ARMAX model is shown in Fig. 1 and is the most appropriate to represent practical systems for control design. In this figure, the input signal $u(t)$ is a control variable, $\zeta(t)$ is a white noise disturbance and $y(t)$ is the system output. Therefore, the system response $y(t)$ can be related to the control input $u(t)$ and the corrupting disturbance $\zeta(t)$ through the following linear difference equation

$$y(t) = -\sum_{i=1}^{n_a} a_i y(t-i) + \sum_{j=0}^{n_b} b_j u(t-k-j) + \sum_{j=1}^{n_c} c_j \zeta(t-j) + \zeta(t) \tag{6}$$

where n_a represents the number of poles of the plant, n_b is the number of zeros of the plant and n_c is the number of zeros of the disturbance model. Therefore, introducing the backward shift operator z^{-k} yields the system's model in the form of discrete time transfer function as follows:

$$y(t) = \frac{B}{A} u(t) + \frac{C_d}{A} \zeta(t) \tag{7}$$

where C_d , B and A are polynomials in the unit shift operator z^{-1} as given above. Then, the model (7) can be put in a vector form as follows

$$y(t) = \theta^T \phi(t) + \zeta(t) \tag{8}$$

where the variable θ denotes the parameter vector formed as

$$\theta = [a_1 \ a_2 \ a_3 \ \dots \ a_{n_a} \ b_0 \ b_1 \ \dots \ b_{n_b} \ c_1 \ c_2 \ \dots \ c_{n_c}]^T \tag{9}$$

The variable ϕ represents the regression/signal vector defined as

$$\phi = [-y(t-1) \ -y(t-2) \ \dots \ -y(t-n_a) \ u(t-k) \ u(t-k-1) \ \dots \ u(t-k-n_b) \ \zeta(t-1) \ \zeta(t-2) \ \dots \ \zeta(t-n_c)]^T \tag{10}$$

This model is not linear regression since $\phi(t)$ contains the unobserved data $\zeta(t)$. Then, applying any of the available identification methods to the model (3 or 8) yields estimates for system polynomials $A(z^{-1})$, $B(z^{-1})$ and $C_d(z^{-1})$ which constitute the parameter vector θ of the respective model.

3- The Recursive Least Squares Method (RLS)

Recalling the model (3) as: $y(t) = \theta^T \phi(t) + d(t)$ with the definitions of θ and ϕ as in (4) and (5) and the disturbance term $d(t)$ is not specified, the objective is to obtain estimates of the parameter vector θ from measurements of $y(t)$ and $\phi(t)$. Since $d(t)$ in eqⁿ (3) is not specified, a prediction of $y(t)$ given the previous values of $\{y(t-1), \dots, u(t-k), \dots\}$ will be: $\hat{y}(t/\theta) = \phi^T(t) \hat{\theta}(t-1)$ and the prediction error is, therefore, defined as:

$$\varepsilon(t) \equiv y(t) - \hat{y}(t/\theta) = y(t) - \phi^T(t) \hat{\theta}(t-1) \tag{11}$$

A natural way to obtain estimates of θ is to try and minimize the prediction error, Eqn (11), using the following criterion:

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^N \{y(t) - \phi^T(t) \hat{\theta}(t-1)\}^2 \tag{12}$$

This criterion is quadratic in θ and can therefore be minimized (w.r.t. θ) analytically to yield the estimate as follows:

$$\hat{\theta}(N) = \left[\sum_{t=1}^N \phi(t) \phi^T(t) \right]^{-1} \cdot \left[\sum_{t=1}^N \phi(t) y(t) \right] \tag{13}$$

This is the non-recursive least-squares estimate of the parameter θ . However, real-time applications necessitates the recursive form of the estimate that can be described by the following RLS algorithm:

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t) \varepsilon(t) \\ \varepsilon(t) &= y(t) - \phi^T(t) \hat{\theta}(t-1) \\ K(t) &= \frac{P(t-1) \phi(t)}{1 + \phi^T(t) P(t-1) \phi(t)} \\ P(t) &= P(t-1) - \frac{P(t-1) \phi(t) \phi^T(t) P(t-1)}{1 + \phi^T(t) P(t-1) \phi(t)} \end{aligned} \tag{14}$$

where $\varepsilon(t)$ is the prediction error, which is the difference between the measured output $y(t)$ and the one-step ahead prediction $[\hat{y}(t/t-1, \hat{\theta}(t-1)) \equiv \phi^T(t) \hat{\theta}(t-1)]$ of $y(t)$ made at time $(t-1)$ and based on the model corresponding to the estimate $\hat{\theta}(t-1)$. The variable $K(t)$ is a weighting or gain factor, called kalman-gain factor, showing how much the value of $\varepsilon(t)$ will modify the different elements of the parameter vector. The form of the kalman-gain factor $K(t)$ in eq^m (14) is more convenient to be implemented in computation and updating of $P(t)$. The variable θ denotes the parameter vector formed as in Eqn(4) while ϕ represents the regression vector and defined as Eqn(5). If $\varepsilon(t)$ is small, the estimate $\hat{\theta}(t-1)$ is good and should not be modified very much. The covariance matrix $P(t)$ is given by the following relationship:

$$P(t) = \left[\sum_{t=1}^N \phi(t)\phi^T(t) \right]^{-1} \tag{15}$$

This matrix is called so because its expectation is proportional to the covariance matrix of the parameter estimates. The initial values that must be supplied to the algorithm are $\hat{\theta}(0)$ and $P(0)$. This algorithm gives unbiased estimates {i.e. estimates tend to the true parameters at steady state} only if $d(t)$ and $\phi(t)$ are uncorrelated, which holds when $d(t)$ is a white noise sequence.

4- The Recursive Instrumental Variables Method (RIV)

This method is a modification of the RLS algorithm and is designed to obtain unbiased estimates in cases where RLS would lead to biased estimates. Recalling the model (3) $y(t) = \theta^T \phi(t) + d(t)$, with the disturbance term $d(t)$ and $\phi(t)$ are correlated (i.e. $d(t)$ is not white noise) which will lead to biased estimates of θ . In this method (RIV), the signal vector $\phi(t)$ is replaced by a vector $\Psi(t)$, called the instrumental variable vector, such that: the signals $\Psi(t)$ and $d(t)$ are uncorrelated while the signal $\Psi(t)$ is as strongly correlated with $\phi(t)$ as possible. That is, the estimate give by Eqn(13) now becomes:

$$\hat{\theta}(N) = \left[\sum_{t=1}^N \Psi(t)\phi^T(t) \right]^{-1} \cdot \left[\sum_{t=1}^N \Psi(t)y(t) \right] \tag{16}$$

or

$$\hat{\theta}(N) = \left[\sum_{t=1}^N \Psi(t)\Psi^T(t) \right]^{-1} \cdot \left[\sum_{t=1}^N \Psi(t)y(t) \right] \tag{17}$$

where the form (16) is called RIV while (17) is called symmetric RIV. Considering Eqn(16), the RIV algorithm will be as follows:

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)\varepsilon(t) \\ \varepsilon(t) &= y(t) - \phi^T(t)\hat{\theta}(t-1) \\ K(t) &= \frac{P(t-1)\Psi(t)}{1 + \phi^T(t)P(t-1)\Psi(t)} \\ P(t) &= P(t-1) - \frac{P(t-1)\Psi(t)\phi^T(t)P(t-1)}{1 + \phi^T(t)P(t-1)\Psi(t)} \end{aligned} \tag{18}$$

A number of variants of the RIV algorithm exists, reflecting the freedom in choosing the instrumental variable $\Psi(t)$. A common approach is to define an unbiased auxiliary system driven by the actual input $u(t)$;

$$\hat{A}(z^{-1})y(t) = \hat{B}(z^{-1})u(t) \tag{19}$$

where; \hat{A} and \hat{B} are the estimated system polynomials. Therefore, the instrumental variable vector is chosen as

$$\Psi(t) = [-y_m(t-1) \ -y_m(t-2) \ \dots \ -y_m(t-n_a) \ u(t-k) \ u(t-k-1) \ \dots \ u(t-k-n_b)]^T \tag{20}$$

and the value of $y_m(t)$ is obtained from eqⁿ (20) or from the relation: $y_m(t) = \hat{\theta}^T(t)\Psi(t)$. Now, equations (19, 20) together with eq^m (18) define one version of the RIV algorithms. This method has the advantage that it gives unbiased estimates of the parameters of the model (3) even when $d(t)$ is not white. However, it has a larger computational expense than the RLS method.

5- Recursive Maximum Likelihood (RML)

Recalling the ARMAX model (8) with the definitions of the parameter vector θ given by Eqn(9) and the regressor vector by Eqn(10), the derivation of the RML algorithm involves forming a prediction $\hat{y}(t/\theta)$ of the output $y(t)$, then minimizing the error between the predicted and measured values of the output. This approach leads to the family of methods known as Prediction Error Methods (PEM). Let the prediction error be defined as $\varepsilon(t, \theta) = y(t) - \hat{y}(t/\theta)$ and the criterion function defined by

$$V_N(\theta) = \frac{1}{2} \sum_{t=1}^N \varepsilon^2(t/\theta) \tag{21}$$

The optimal prediction of the output of the ARMAX model, Eqn (7), is given by [3]:

$$C_d(z^{-1})\hat{y}(t/\theta) = [C_d(z^{-1}) - A(z^{-1})]y(t) + B(z^{-1})u(t) \tag{22}$$

Inspection of Eqⁿ(22) reveals that $\hat{y}(t/\theta)$ is a non-linear function of θ . This means that the criterion $V_N(\theta)$ can not be minimized analytically, and non-recursive numerical methods have to be used. Thus, for obtaining a Recursive Prediction Error algorithm some approximations have to be made as follows: the regression vector ϕ , Eqⁿ (10), contains the unobserved components $\{\zeta(t-1)\dots\zeta(t-n_c)\}$ which can be approximated by the residual error defined as $\varepsilon_r(t) = y(t) - \phi^T(t)\theta(t)$. That is, the regression vector has the form:

$$\phi = \begin{bmatrix} -y(t-1) & -y(t-2) & \dots & -y(t-n_a) \\ u(t-k) & u(t-k-1) & \dots & u(t-k-n_b) \\ \varepsilon_r(t-1) & \varepsilon_r(t-2) & \dots & \varepsilon_r(t-n_c) \end{bmatrix}^T \tag{23}$$

The minimization of $V_N(\theta)$, eqⁿ (21), gives the following RML algorithm:

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)\varepsilon(t) \\ \varepsilon(t) &= y(t) - \phi^T(t)\hat{\theta}(t-1) \\ K(t) &= \frac{P(t-1)\psi(t)}{1 + \psi^T(t)P(t-1)\psi(t)} \\ P(t) &= P(t-1) - \frac{P(t-1)\psi(t)\psi^T(t)P(t-1)}{1 + \psi^T(t)P(t-1)\psi(t)} \\ \psi(t) &= \frac{\phi(t)}{\hat{C}_d(z^{-1})} \end{aligned} \tag{24}$$

where $\psi(t)$ is called the gradient vector. Looking at this algorithm it seems to be close to the RLS except that the regressor vector is extended by the components of $\varepsilon_r(t-1)$ and the gradient vector $\psi(t)$ is used.

6- Recursive Extended Least Squares (RELS): {or Pseudo-Linear Regression (PLR)}

In the RELS method it is attempted to cast the ARMAX model (8) in the form of a linear regression (3) and then applying the ordinary RLS algorithm to the obtained model. From the ARMAX model representations, it is clear that this is not a linear regression since the components $\{\zeta(t-1)\dots\zeta(t-n_c)\}$ in the regression vector $\phi(t)$ are unknown which must be replaced by its estimates given the current parameter estimate. That is, $\zeta(t) \equiv \varepsilon_r(t) = y(t) - \phi^T(t)\hat{\theta}(t)$ and the regressor vector is consequently modified to contain the residual errors. Therefore, applying the RLS algorithm to model (8), the RELS algorithm will be as follows:

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)\varepsilon(t) \\ \varepsilon(t) &= y(t) - \phi^T(t)\hat{\theta}(t-1) \\ K(t) &= \frac{P(t-1)\phi(t)}{1 + \phi^T(t)P(t-1)\phi(t)} \\ P(t) &= P(t-1) - \frac{P(t-1)\phi(t)\phi^T(t)P(t-1)}{1 + \phi^T(t)P(t-1)\phi(t)} \end{aligned} \tag{25}$$

$$\varepsilon_r(t) = y(t) - \phi^T(t)\hat{\theta}(t)$$

Comparing algorithms (14) and (25) shows that the RLS and the RELS methods are computationally identical, except that the RELS is complemented by the residual errors given by: $\varepsilon_r(t) = y(t) - \phi^T(t)\hat{\theta}(t)$. That is, the parameter and regression vectors are extended by the inclusion of the disturbance term. Comparing algorithms (24) and (25) reveals that the only difference is that the vector $\phi(t)$ is replaced by the filtered vector $\psi(t)$ in the equations of $K(t)$ and $P(t)$. This filtered vector is obtained by filtering $\phi(t)$ through the estimated polynomial \hat{C}_d . This is the reason for calling the RELS, also, as the Approximate Maximum Likelihood (AML) method. This filtering operation represents the increased computation required for the RML algorithm when compared to the RELS algorithm.

7- Implementation of System Identification

It is assumed to implement system identification algorithms with time-invariant systems whose parameters are constant but unknown. However, in practice the systems will vary with time so that those algorithms must be equipped with some means of tracking these variations, which can be considered as time-varying parameters, or time-varying noise variance. Therefore, appropriate techniques [2,3,4,8,12,13,17,18,20] should be used to improve the robustness, convergence and tracking capabilities of the identification algorithm working in such corrupted environments. Among these techniques is the forgetting factor approach in which an appropriate weight is put on each measurement according to its contribution to the estimate. If the system to be identified has parameters which vary with time, however, then old data may not be relevant to the actual model and their influence should be reduced by using a weighting factor which increases with time. This idea is achieved using an exponential weighting (λ_f) in the criterion of the identification algorithm which is slightly less than unity {e.g. between 0.95 and 0.99}. That is, with increasing the time t , the measurements obtained previously are discounted. A measurement received n samples ago will have a weight proportional to $\lambda_f^n = e^{-n \ln(\lambda_f)}$. The smaller the value of λ_f , the quicker the information in previous data will be forgotten. The choice of the value of λ_f is a trade off between tracking ability and noise sensitivity: a smaller value of λ_f gives faster discounting or tracking but leads to a greater noise sensitivity, and the inverse is true for λ_f close to one. That is, if $\lambda_f = 1$ the algorithm will have good convergence and if $\lambda_f < 1$ it will be more sensitive and the parameter estimates change quickly. An alternative technique is the Kalman gain approach in which the parameter vector is modeled as a random walk or a drift and consequently the covariance matrix P is modified by the addition of the matrix R_1 which used to describe how fast the different components of ζ are expected to vary. Therefore, the covariance matrix $P(t)$ will no longer tend to zero, and $K(t)$ will not decrease to zero and consequently the parameter estimates will change continually.

Thus, λ_f and/or R_1 are design variables to be chosen by the user via a trade-off between alertness and ability to track time variations of the parameters {which requires λ_f small or R_1 large} on the one hand, and good convergence properties and small variances of the estimates for time-invariant systems {which requires λ_f close to 1 or R_1 close to 0} on the other hand. The kalman gain approach offers more flexibility than the forgetting factor approach does, since the whole matrix R_1 can be set by the user giving the possibility to choose different diagonal elements for describing different time variations for the different parameters. If the system exhibits some fast parameter changes, some modified methods are necessary, through which a fault detector will be used to test for the occurrence of significant parameter changes. Once a change is detected the algorithm could be restarted, at least partly, by decreasing the forgetting factor temporarily or to increase R_1 or parts of the matrix $P(t)$.

If the variance of the disturbance term $d(t)$ is varying, a weight inversely proportional to the variance should be put on each measurement. Let $d(t)$ be a white noise with variance $\sigma(t)$, so to account for this variation the criterion is modified by dividing the error components $\sigma(t)$. That is, the more uncertain measurements the less significance will be. This approach could be combined with the forgetting factor approach to yield alert or robust identification algorithm.

Practically, it is reasonable to take the initial value for $\hat{\theta}(0)$ as *a priori* estimate of θ and to let $P(0)$ reflect the confidence in this initial estimate $\hat{\theta}(0)$. If $P(0)$ is small then $K(t)$ will be small for all t and the parameter estimates will therefore not change too much from $\hat{\theta}(0)$. If $P(0)$ is large, the parameter estimates will quickly jump away from $\hat{\theta}(0)$ and therefore $\hat{\theta}(t)$ will be close to the off-line estimates. Without any *a priori* information, it is common practice to take $\hat{\theta}(0) = 0$ and $P(0) = \sigma_1 I$; where σ_1 is a large number.

Steady-State Offsets: Usually the identification algorithms are concerned only by the ARX models which did not include the effects due to external disturbances. These ignored disturbances may easily degrade the performance of the identification algorithm resulting in inaccurate model. However, those including the external disturbances considered it as unknown quantities although inputs to the system. This situation has the

benefits of avoiding the nonlinear form resulting from considering these types of system structures. Unfortunately, in practice this assumption rarely satisfied resulting in some sort of steady-state offsets. This problem can be overcome by inserting a unity in the regressor vector then identifying the system offset through the parameter vector. However, this approach have the drawback of increasing the order of the regressor and parameter vectors and consequently the required number of numerical processes. An alternative approach to tackle this problem is to purify the data measurements or the set of observations by removing the steady-state offsets.

8- Numerical implementation

Now the RELS algorithm is implemented to identify the model parameters of a given system whose parameters are either constant or time-varying (the zero of the process is shifted five times its initial location).

Example 1: fixed parameters plant

Consider a plant whose parameters are fixed with polynomials (process and disturbance) are defined as follows:

$$W_p \equiv \frac{B}{A} = \frac{z^{-1}(0.2 + 0.02z^{-1})}{1 - 1.5z^{-1} + 0.56z^{-2}} \quad \& \quad W_d \equiv \frac{C_d}{A} = \frac{(1 - 0.5z^{-1})}{1 - 1.5z^{-1} + 0.56z^{-2}}$$

The input signal $u(t)$ is a pseudo-random binary sequence (PRBS) with amplitude ± 10 and the disturbance $\zeta(t)$ is a white noise with zero mean and 0.1 variance. The evolution of parameter estimates is shown in Fig. 2, using constant forgetting factor with $\lambda_f = 0.995$ and $\lambda_{f_c} = 1.0$, and initial covariance matrix $P(0) = 10^6 I$. From the figure it is clear that the parameters converge very fast to the correct values with zero steady state errors while the disturbance polynomial C converge with relatively slow profile. The initial excursion of estimates is due to the initial poor information which overcome quickly during operation of the process.

Example 2: Plant with varying parameters

Consider the previous plant whose zero changed during the evolution of identification process five times its initial location after 100 sample. That is, the parameter $b_1 = 0.1$ instead of 0.05. The evolution of parameter estimates is shown in Fig. 3 using constant forgetting factor with $\lambda_{f_c} = 0.85$ and initial covariance matrix $P(0) = 10^6 I$. From this figure it is clear how the algorithm detects the change in parameters during operation and modify itself quickly to track the new parameters value.

Conclusions

The paper highlights the need for systems identification, especially the recursive approach, in self-tuning control with real applications. Then it presented the most widely used models in systems identification: the linear regression model and the ARMAX model in addition to the concepts of identifiability and experiment design. The recursive least squares (RLS), the recursive instrumental variables (RIV), the recursive maximum likelihood (RML) and the recursive extended least squares (RELS) methods for systems identification were described in algorithmic approach ready to implement on computers directly, using any of the known or available high level language. The results obtained using programs due the author on MATLAB and C-Language. The paper addresses the point of how to improve the robustness of the identification algorithm and its convergence. The results show how the plant polynomials (A and B parameters) converge very fast to the correct values with zero steady state errors while the disturbance polynomial C converge with relatively slow profile. In addition, the algorithm detects the change in parameters during operation and modify itself quickly to track the new parameters value. The point to be remembered here is how to design the experiment and choose the appropriate value for the forgetting factor and the covariance matrix, as highlighted in this paper.

References

- [1] Astrom K.J. and B. Wittenmark, *Adaptive Control*, Addison Wesley, 1989.
- [2] Bierman, G.J., *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York, 1977.
- [3] Bittanti, S., P. Bolzern and M. Campi, *Convergence and Exponential Convergence of Identification Algorithms with Directional Forgetting Factor*, Automatica, Vol. 26, No. 5, pp 929-932, 1990.
- [4] Chen and J.P. Norton, *Estimation Technique for Tracking Rapid Parameter Changes*, Int. J. Control, Vol. 45, No. 4, 1987, pp 1387-1398.
- [5] Dstrom, K.J., and P. Eykhoff, *System Identification: A Survey*, Automatica, Vol. 7, pp 123-162, 1971.

[6] El-Sheikh, G.A., M.J. Grimble and M.A. Johnson, *On the Performance of GH_∞ Self-Tuning for AeroEngine Control*, Control'94, Warwick University, UK, March 21-24, 1994a.

[7] El-Sheikh, M.A. Johnson, and M.J. Grimble, *Explicit Self-Tuning Control Using Generalized Cost Functions*, Third International Conference on Computer Applications in Industry, Cairo, Egypt, Dec. 26-29, 1994b.

[8] El-Sheikh, G.A., *ARMAX Augmented UD Identification Algorithm (AUDIX)*, Submitted the 7th ASAT Conference, Military Technical College, Cairo, Egypt, May 13-15 1997.

[9] Favier, G. and D. Dubois, *A Review of K-Step-Ahead Predictors*, Automatica, Vol. 26, No. 1, pp 75-84, 1990.

[10] Goodwin, G.C. and K.S. Sin, *Adaptive Filtering, Prediction and Control*, Prentice Hall, 1984.

[11] Greene, M.E. and T.S. Denney, *Real-Time Estimator for Control of an Orbiting Single Tether System*, IEEE Transactions on Aerospace and Electronic Systems, Vol. 27, No. 6, November 1991.

[12] Gustavsson, L. Ljung and T. Soderstrom, *Survey Paper: Identification of Processes in Closed Loop Identifiability and Accuracy Aspects*, Automatica, Vol. 13, pp 59-75, 1977.

[13] Ljung, L., *System Identification: Theory for the User*, Prentice-Hall, 1987.

[14] Ljung, L. and T. Soderstrom, *Theory and Practice of Recursive Identification*, The MIT Press, 1983.

[15] Seborg, D.E., T.F. Edgar and S.L. Shah, *Adaptive Control Strategies for Process Control: A survey*, AIChE Journal, Vol.32, No.6, pp 881-913, June 1986.

[16] Soderstrom, T. and P. Stoica, *System Identification*, Prentice Hall, 1989.

[17] Weinfeld, R., *Towards More Precise Recursive Prediction Error Algorithm*, Automatica, Vol. 26, No. 5, pp 925-928, 1990.

[18] Wellstead, P.E. and P. Zanker, *Techniques of Self-Tuning, Optimal Control Applications and Methods*, Vol.3, No.4, pp305-322, Oct-Dec 1982.

[19] Wellstead, P.E. and M.B. Zarrop, *Self-Tuning Systems: Control and Signal Processing*, John Wiley & Sons, UK, 1991.

[20] Zhang, H.G., J.Chen, and H.Y.Zhang, *A Recursive Identification Algorithm for Estimating Abrupt Parameter Changes*, IEE International Conference Control 1991, number 332, Vol.1

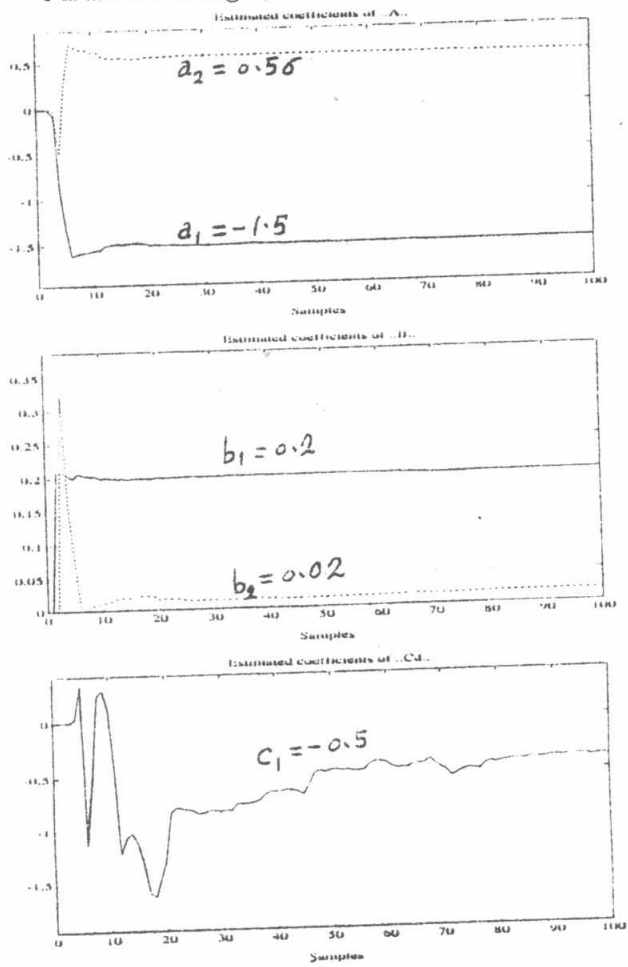


Fig. 2: The evolution of parameter estimates

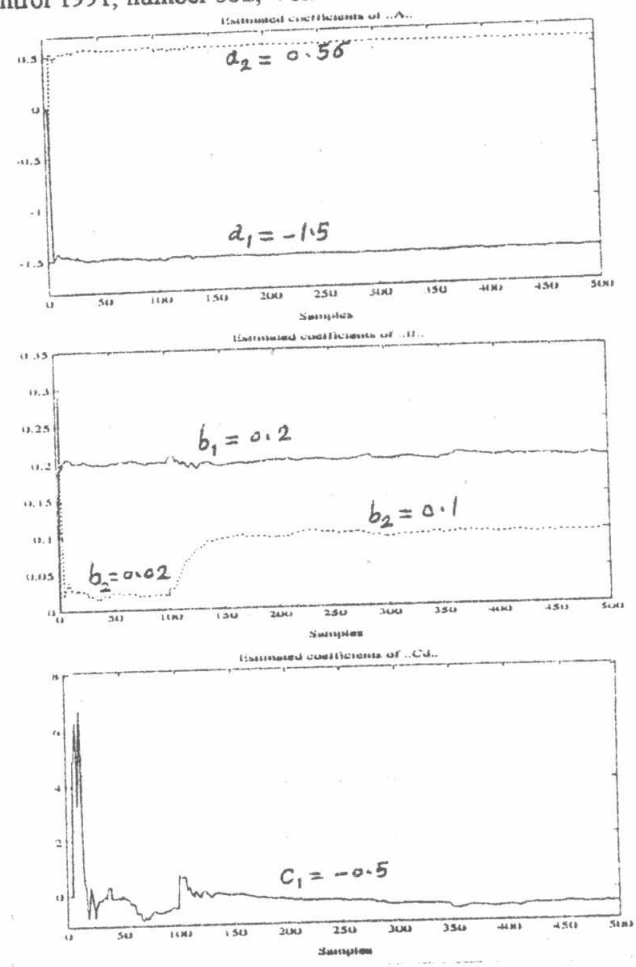


Fig. 3 The evolution of parameter estimates when one of them changed during operation