Control Engineering, 1982,
Newcastle, 25-27 August

When will Adaptive Systems Really Adapt? The Robustness Issue

B.D.O. ANDERSON
Professor of Systems Engineering, Research School of Physical Sciences,
The Australian National University, A.C.T.
and
R.M. JOHNSTONE
Graduate Student, Department of Systems Engineering, Research School of Physical Sciences,
The Australian National University, A.C.T.

SUMMARY A robust adaptive system is one that in some way will respond to changes in its environment, coping satisfactorily with noise, disturbances, minor errors in modelling assumptions, and so on. Various types of adaptive identification and control algorithms are shown to be sometimes capable of nonrobust behaviour, and conditions for robust behaviour are described. These conditions are stated in terms of driving inputs or reference trajectories, i.e. signals which are known to the user of the adaptive system.

1 THE MEANING OF "ADAPTIVE" AND "ROBUST"

There are at least two distinct meanings for the adjective adaptive in the control system's literature and we need to make clear what these are and to which we are referring. First, one can conceive of a system which is time-invariant and unknown. The system has to be identified, and possibly controlled, so that all or part of the task is to learn a description of the system. Once the description has been learnt however, there is no need to keep the identification or learning algorithm running since the system is time-invariant. Secondly, one can conceive of a system the parameters of which are time-varying (commonly slowly time-varying, where slowness is defined in relation to the dominant time-constants of the system). Again, the system has to be identified and possibly controlled. Its initial set of parameter values may well be known. Now, part or all of the task is to learn and continuously relearn, the description of the system as it evolves in time. The identification or learning algorithm now must be kept running and never terminates.

To the extent that conventional usage of the word adaptive connotes an ability to adjust in response to changes in the environment it seems to us more appropriate to call the second type of algorithm adaptive. This is also probably the preferred choice for practical purposes.

To understand our usage of the word robust, consider a simple example from classical control. Suppose that an engineer is faced with the task of designing a controller securing closed-loop stability for a linear time-invariant plant with a pole at s=1. Then blind and simple application of theory suggests that a series controller with a zero at s=1 could be used to cancel the pole at s=1. However, the engineer should appreciate that the idealised design will not translate into reality satisfactorily, because the design is not robust. That is, plant parameter variations, or noise, or inexactitude in the compensation will destroy the effectiveness of the algorithm, with instability resulting in practice.

In the adaptive identification and control context, we similarly must be concerned about whether our idealised algorithms will translate satisfactorily into reality. In particular, we require robustness in the presence of

- noise, in both actual plant measurements, and in the implementation of the algorithm (Paradoxically, noise can sometimes make an otherwise nonrobust algorithm robust.)
- errors in modelling assumptions (We might assume a plant is linear, when it has some nonlinearity, or we might assume it has no high frequency modes when this is not the case.)
- time-variation in plant parameters
- changes in the nature of the input (i.e. if the algorithm works satisfactorily for a certain style of input, it should continue to work satisfactorily if the input style is altered).

2 TYPES OF PROBLEMS CONSIDERED

The range of adaptive problems for which reasonably definitive statements can be made is really quite limited, see e.g. the recent [1] and forthcoming [2] books. In most problems considered the assumption is made that the plant is linear, finite-dimensional and time-invariant. Further, it is usual to work with a transfer function description (or the equivalent, i.e. a differential or difference equation, or canonical state variable description), to assume known the denominator degree and quite commonly also the numerator degree, but to assume unknown the coefficients of the denominator or numerator polynomials.

Note that continuous-time modelling of pure time delays is not normally considered; nor are state-variable models where all but a few parameters in the matrices of the model are unknown. Thus two types of model which are most important for application purposes are immediately excluded.

A number of dichotomies arise in the treatment of adaptive system problems with models of the type noted. For example one can use

- continuous-time, or discrete-time models
- stochastic-approximation-type or recursive-least-squares-type parameter update algorithms
- parameter estimate update algorithms where the gain controlling updates to the parameter estimate is, or is not, turned off as time approaches infinity.

Each of these choices is worthy of some comment.

2.1 Continuous or Discrete Time

While the use of a computer in the adaptive mechanism argues for discrete-time, one must recognize two potential difficulties associated with working with a discrete-time sampled-data model of a continuous-time system. First, parameters with physical significance in the continuous-time model can get buried in the discrete-time model, implying a reduction in the intuitive content of the model. Second, even though the continuous-time model may be minimum phase, the associated discrete-time model may not be so, which means that any control problem is likely to be more difficult. For purposes of clarity, we shall restrict ourselves in the paper mainly to discrete-time models.

2.2 Stochastic-Approximation-Type or Recursive-Least-Squares-Type Parameter Update Algorithms

From the computational point of view, the former is simpler to implement while the latter involves an extra matrix inversion at each step. Assuming this computational burden can be overcome with adequate numerical accuracy, the RLS-type algorithms normally lead to speedier convergence. The analogy is with static optimization problems, SA-type algorithms corresponding to gradient algorithms and RLS-type algorithms to Newton-Raphson type algorithms.

2.3 Turning Off or Not Turning Off the Gain

Most adaptive algorithms contain a parameter update equation of the form

$$\hat{\theta}_{k+1} = \hat{\theta}_k + K_k \times \text{error}$$

Here, $\hat{\theta}_k$ is the parameter estimate at time k; the error is a quantity which would be zero were $\hat{\theta}_k$ equal to the correct parameter, with no noise present; K_k is a gain term, depending on for example the choice between SA-type or RLS-type algorithms. Statisticians have created a substantial body of literature in which $K_k \to 0$ as $k \to \infty$. The thinking behind this literature is as follows. The models under consideration are time-invariant and there is additive measurement noise. Despite the noise, it is possible (essentially by averaging) to get a better and better knowledge of the unknown parameters as $k \to \infty$. However, the noise itself may be confused with a true signal and lead to misadjustment of the parameters. Hence, as $k \to \infty$ one cuts down on K_k, thereby coupling in the noise to a lesser and lesser degree. It turns out one can get an appropriate rate of decrease of K_k which both allows learning of the true parameter and ultimate suppression of the noise effects. The difficulty with this approach is that it does not permit a situation where the plant is (slowly) time-varying. In terms of Section 1, true adaptive behaviour is precluded. Securing true adaptive behaviour requires $K_k \neq 0$, and carries with it the penalty that in the presence of noise, $\hat{\theta}_k$ can never be expected to tend to the correct value (though it is possible in a given situation for it to be close, say in a mean square sense). Both SA-type and RLS-type algorithms can be run with $K_k \neq 0$; RLS-type algorithms are usually described then as containing a forgetting factor.

Within the framework of identification problems, a choice between equation error and output error methods can be identified (crudely, a tradeoff is offered between the size of the capture region of a priori parameter estimates and the difficulties associated with measurement noise). Within the framework of adaptive control, a number of different problems can be identified, e.g.

- model reference adaptive control. (A model is given, and the initially unknown plant must be controlled so that the input-output performances of the model and the controlled plant approach one another.)
- reference trajectory following (the output of the plant must tend towards a prescribed reference trajectory)
- pole positioning (prescribed closed-loop poles must be attained)
- linear quadratic control (a linear-quadratic performance index must be minimized. As it turns out, the tracking problem rather than the regulator problem is relevant).

3. EXAMPLES OF NONROBUST AND ROBUST ALGORITHMS

Without giving an illustration of nonrobust behaviour for every conceivable algorithm, we do wish to give a representative number of examples which will convince the reader that the possibility of nonrobust behaviour must continually be considered.

Example 1 Equation error identification, with S-A type algorithm* [1].

The plant is

$$y_k = ay_{k-1} + bu_{k-1} \tag{3.1}$$

with a,b unknown, save that we know $|a| < 1$ and $b \neq 0$. We conceive of an adjustable model

$$\hat{y}_k = \hat{a}_{k-1}y_{k-1} + \hat{b}_{k-1}u_{k-1} \tag{3.2}$$

The equation error algorithm adjusts $\hat{a}_{k-1}$ and $\hat{b}_{k-1}$ to try to bring $y_k - \hat{y}_k$ to zero. The update algorithm is

$$\begin{bmatrix}\hat{a}_k \\ \hat{b}_k\end{bmatrix} = \begin{bmatrix}\hat{a}_{k-1} \\ \hat{b}_{k-1}\end{bmatrix} + \frac{\rho}{y_{k-1}^2+u_{k-1}^2+\nu}\begin{bmatrix}y_{k-1} \\ u_{k-1}\end{bmatrix}[y_k-\hat{y}_k] \tag{3.3}$$

where $\nu > 0$ (usually small) and $\rho \in (0,2)$, but again, ρ is usually small. (It is of course possible to contemplate exact computation of a,b given measurements of y_k, u_k over several time instants without using (3.3). However, our concern is to obtain algorithms which will be tolerant of noise and other non-idealities, and the equation error algorithm potentially has this property.)

By setting $\tilde{a}_k = a-\hat{a}_k$, $\tilde{b}_k = b-\hat{b}_k$, which are parameter estimation errors, one can derive

$$\begin{bmatrix}\tilde{a}_k \\ \tilde{b}_k\end{bmatrix} = \left[I - \frac{\rho\ \psi_{k-1}\ \psi_{k-1}'}{\psi_{k-1}'\ \psi_{k-1} + \nu}\right]\begin{bmatrix}\tilde{a}_{k-1} \\ \tilde{b}_{k-1}\end{bmatrix} \tag{3.4}$$

where

$$\psi_{k-1}' = [y_{k-1}\ u_{k-1}] \tag{3.5}$$

---

* In all examples, the algorithm gains given will be prevented from tending to zero.

Now consider the ideal equation under several operating conditions.

Case 1 $u_k \equiv 0$. Neglecting initial conditions, $y_k \equiv 0$, and $\hat{a}_k = \hat{a}_{k-1}$, $\hat{b}_k = \hat{b}_{k-1}$ for all k. Also $\hat{y}_k = y_k$. It will look to an observer of the plant and model that the parameter estimates have converged, since there is satisfactory tracking of plant output (3.1) by model output (3.2). But of course no identification has taken place.

Case 2 $u_k \equiv 1$. Then, after a possible initial transient, $y_k \equiv b(1-a)^{-1}$. Change the variables in (3.4) to

$$\begin{bmatrix} \tilde{c}_k \\ \tilde{d}_k \end{bmatrix} = \begin{bmatrix} b(1-a)^{-1} & 1 \\ 1 & -b(1-a)^{-1} \end{bmatrix} \begin{bmatrix} \tilde{a}_k \\ \tilde{b}_k \end{bmatrix} \qquad (3.6)$$

This leads to

$$\tilde{c}_k = \left(1 - \frac{\rho\, \psi_{k-1}'\psi_{k-1}}{\psi_{k-1}'\psi_{k-1} + \nu}\right) \tilde{c}_{k-1} = \lambda \tilde{c}_{k-1} \qquad (3.7)$$

$$\tilde{d}_k = \tilde{d}_{k-1}$$

where $\lambda \in (0,1)$. Now we see that one linear functional of $[\tilde{a}_k\ \tilde{b}_k]$ converges exponentially fast to zero, while one does not converge, assuming it is initially nonzero.

From (3.6) and (3.7)

$$\tilde{a}_k = \frac{b(1-a)}{b^2+(1-a)^2}\,\tilde{c}_k + \frac{(1-a)^2}{b^2+(1-a)^2}\,\tilde{d}_k$$

$$\tilde{b}_k = \frac{(1-a)^2}{b^2+(1-a)^2}\,\tilde{c}_k - \frac{b(1-a)}{b^2+(1-a)^2}\,\tilde{d}_k \qquad (3.8)$$

Thus, $\tilde{a}_k$ and $\tilde{b}_k$ converge exponentially fast to nonzero values, i.e. identification of the true plant parameters a and b does not occur. However, note that $y_k - \hat{y}_k \to 0$ exponentially fast.

Case 3 $u_{4k} = u_{4k+1} = 1$, $u_{4k+2} = u_{4k+3} = -1$. This leads to $y_{4k} = -y_{4k+2} = -b(1+a)(1+a^2)^{-1}$ and $y_{4k+1} = -y_{4k+3} = b(1-a)(1+a^2)^{-1}$.

Define

$$\psi_\pm = \begin{bmatrix} \dfrac{-ab \pm b}{1+a^2} & 1 \end{bmatrix}'$$

and

$$A_\pm = I - \frac{\rho\, \psi_\pm\, \psi_\pm'}{\psi_\pm'\, \psi_\pm + \nu} \qquad (3.9)$$

Then, from (3.4)

$$\begin{bmatrix} \tilde{a}_{2k+2} \\ \tilde{b}_{2k+2} \end{bmatrix} = A_+ A_- \begin{bmatrix} \tilde{a}_{2k} \\ \tilde{b}_{2k} \end{bmatrix} \qquad (3.10)$$

It is easy to see that

$\|A_+ A_- x\| < \|x\|$ for any $x \neq 0$, as follows. First observe that $\|A_\pm x\| \leq \|x\|$ (from (3.9)), with equality if and only if $\psi_\pm' x = 0$ (i.e. $A_\pm x = x$). Consequently, $\|A_+ A_- x\| \leq \|A_- x\| \leq \|x\|$ with equalities if and only if $A_+(A_- x) = A_- x$ and $A_- x = x$. But then $\psi_+' x = 0$ and $\psi_-' x = 0$ must hold (from (3.9)). This is impossible since ψ_+ and ψ_- are linearly independent.

What this shows is that (from (3.10)) over a time interval of length two, there is a guaranteed contraction in the size of $\left\| \begin{bmatrix} \tilde{a}_j \\ \tilde{b}_j \end{bmatrix} \right\|$. Thus, for this example it is certain that $[\tilde{a}_j\ \tilde{b}_j]'$ converges to zero exponentially fast. Also, $y_k - \hat{y}_k \to 0$ exponentially fast.

In Case 1, if u_k is known to be zero, common sense dictates that no identification is possible. But one could be misled in Case 2. What could happen in practice?

- round off noise in the implementation of (3.7) is likely to make $\tilde{d}_k$ behave as a Wiener process so that one linear functional of $\tilde{a}_k$, $\tilde{b}_k$ will actually diverge
- likewise, if (3.1) is only an approximation of a real plant, the behaviour of $\tilde{d}_k$ will be modified.

On the other hand, in Case 3 round off noise will do no more than cause minor fluctuations to the behaviour of $\hat{a}_k$, $\hat{b}_k$ because of the convergence mechanism inherent in (3.10). A similar consideration applies to inaccuracy in (3.1).

Even if neither of the above noted difficulties is considered a problem with the Case 2 example, a problem still arises in respect of a change of style of input. Thus, suppose $u_j = 1$ for $j = 1, \ldots, 1000$, and then $u_j = 10$. Since almost certainly $\tilde{d}_k$ will not be zero, one can check that the tracking of the true plant output by the model output after the change of input style may be very poor.

To get satisfactory behaviour of the algorithm, it is basically necessary that u_k be sufficiently complicated - a notion that will be made more precise subsequently. In retrospect, it is obvious that for the uncomplicated u_k of Case 1 and Case 2 one cannot expect satisfactory identification. [In Case 2, all one can expect to identify is the DC gain of (3.1), viz $b(1-a)^{-1}$.]

Example 2 Reference trajectory following (Goodwin-Ramadge-Caines approach [3]) using SA-type algorithm.

The plant is as in (3.1), with a,b unknown but possibly now $|a| > 1$. The reference trajectory is denoted by y_k^*. The general approach is to estimate a,b with estimates $\hat{a}_k$, $\hat{b}_k$ assumed available on measuring y_k. Then u_k is chosen to secure the right y_{k+1}^* acting as if $\hat{a}_k$, $\hat{b}_k$ were the correct a,b. Thus

$$y_{k+1}^* = \hat{a}_k\, y_k + \hat{b}_k\, u_k \qquad (3.11)$$

(If $\hat{b}_k$ is very small, the algorithm is adjusted.)

Now the identification algorithm will ensure that $y_k - y_k^* \to 0$ (for a proof, see [3]). Suppose $y_k^* \equiv 1$. Then (3.11) implies

$$u_k \to \frac{1-\hat{a}_k}{\hat{b}_k}$$

and (3.1) implies

$$1 - a - b\,\frac{1-\hat{a}_k}{\hat{b}_k} \to 0$$

Suppose a=2, b=1. Then

$$\hat{a}_k - \hat{b}_k \to 1 \qquad (3.12)$$

With this particular y_k^* sequence then, it appears as if $\hat{a}_k$, $\hat{b}_k$ cannot be separately identified. Indeed, this is easy to check. Suppose in fact that with $y_k^* \equiv 1$, $\hat{a}_j - \hat{b}_j = 1$ for some j, while a=2, b=1. Suppose also that $y_j = 1$. Then (3.11) implies

$$u_j = \frac{1 - \hat{a}_j}{\hat{b}_j} = -1$$

and (3.1) ensures that $y_{j+1} = 1$. Further, since $y_{j+1} - y_{j+1}^* = 0$, there will be no update of $\hat{a}_k$, $\hat{b}_k$, i.e. $\hat{a}_{k+1} = \hat{a}_k$, $\hat{b}_{k+1} = \hat{b}_k$. Everything looks fine. But what are we actually implementing? Combining (3.1) and (3.11) with a=2, b=1, we have

$$y_{k+1} = (2 - \frac{\hat{a}_k}{\hat{b}_k})y_k + \frac{1}{\hat{b}_k}\,y_{k+1}^* \qquad (3.13)$$

Suppose that $\hat{a}_k = 1.001$, $\hat{b}_k = 0.001$ and that $y_k^* = 1$ for k = 1,2,...j, while $y_{j+1}^* = 10$. Then far from convergence having occurred, we shall find $y_{j+1} = 9001$! This is an example of nonrobustness in the face of a change of "input" (here, reference trajectory) style.

In the presence of round-off noise in the parameter estimation algorithm one would have to expect that the linear functional $\hat{a}_k + \hat{b}_k$ behaved as a Wiener process, and that no control could be exerted over it. (Somewhat paradoxically, noise associated with y_j^* could, as it turns out, "stabilize" the algorithm.) So even if we began with roughly correct values of $\hat{a}_o$ and $\hat{b}_o$ the use of $y_j^* \equiv 1$ would probably mean that $\hat{a}_k + \hat{b}_k$ (but not $\hat{a}_k - \hat{b}_k$) would drift.

Example 3 Reference trajectory following (Widrow approach [4]) using SA-type algorithm.

The idea behind this reference trajectory following algorithm is to identify an inverse of the plant, and drive a copy of this inverse using the y_k^* sequence to generate the required plant input u_k. Figure 1 illustrates the general arrangement for the plant of (3.1). The plant must be stable for this scheme to work, and in its exact form, the plant must also have no zeros. However, an approximate form of the algorithm (not considered here) allows the plant to have zeros, even nonminimum phase zeros.

Suppose now that $y_k^* \equiv 1$, $\hat{c}_o + \hat{d}_o = 0$ and initial conditions of the plant are zero. Then $u_1 = 0$, $y_2 = 0$ and the identifier with zero inputs causes $\hat{c}_1 = \hat{c}_o$, $\hat{d}_1 = \hat{d}_o$. Then $\hat{c}_1 + \hat{d}_1 = 0$, $u_2 = 0$, $y_3 = 0$ and so on. No identification takes place, and in this instance, we do not even have $y_k - y_k^* \to 0$. Again, suppose that $y_k^* \equiv 1$ and $\hat{c}_j + \hat{d}_j = (1-a)b^{-1}$. Then $u_j = (1-a)b^{-1}$. Suppose also that $y_j = 1$. Then $y_{j+1} = 1$. It can be checked that the identifier will give $\hat{c}_{j+1} + \hat{d}_{j+1} = (1-a)b^{-1}$. So we see that $u_k = (1-a)b^{-1}$ for all k, and we would expect $\hat{c}_k - \hat{d}_k$ to drift. So long as y_j^* remains at 1, y_j would track it. But if y_j^* changed to 10 after a long time interval, i.e. there is a change of "input" style, the tracking of it by y_j may be very poor, depending on the value to which $\hat{c}_k - \hat{d}_k$ had drifted.

Example 4 Adaptive pole positioning with SA-type identification.

Suppose the plant is as in (3.1), without necessarily $|a| < 1$ and we have to find a feedback law $u_k = fy_k + u_k^*$ so that a + bf = d, where 0<d<1 with d prescribed. Because a, b are unknown, we estimate them via $\hat{a}_k$, $\hat{b}_k$, choose $\hat{f}_k$ so that $\hat{a}_k + \hat{b}_k\hat{f}_k = d$, and implement $u_k = \hat{f}_k y_k + u_k^*$.

One worry with this scheme is the problem of what one should do in case $\hat{b}_k$ is zero, or very small. Let us leave aside this question for the moment and consider a situation where nonrobust behaviour can arise.

The closed-loop plant will be

$$y_k = (a + b\,\frac{d-\hat{a}_{k-1}}{\hat{b}_{k-1}})\,y_{k-1} + bu_{k-1}^* \qquad (3.14)$$

and the prediction of y_k which the identifier will make will be

$$\hat{y}_k = dy_{k-1} + \hat{b}_{k-1}\,u_{k-1}^* \qquad (3.15)$$

In all treatments of this problem, not surprisingly the allegedly satisfactory algorithms require $y_k - \hat{y}_k \to 0$

If $u_k^* \equiv 1$, it is easily checked that this convergence implies

$$1 - \hat{a}_k - \hat{b}_k\,(\frac{1-a}{b}) \to 0 \qquad (3.16)$$

All this suggests the following converse argument. Suppose that initially

$$1 - \hat{a}_o - \hat{b}_o\,(\frac{1-a}{b}) = 0 \qquad (3.17)$$

and that

$$y_o = \frac{\hat{b}_o}{1-d} \qquad (3.18)$$

Then it is easily verified that $y_1 = \hat{y}_1 = \hat{b}_o(1-d)^{-1}$, and so $\hat{a}_1 = \hat{a}_o$, $\hat{b}_1 = \hat{b}_o$. More generally, y_k, $\hat{y}_k$, $\hat{a}_k$ and $\hat{b}_k$ will apparently remain constant for all time.

Suppose in particular that a=0, b=1, d=0.9. Then (3.17) implies $\hat{a}_o + \hat{b}_o = 1$. From (3.14) the actual closed-loop pole will then be at

$$\left[0 + 1\,\frac{0.9 - (1-\hat{b}_o)}{\hat{b}_o}\right] = 1 - \frac{0.1}{\hat{b}_o}$$

Obviously, if $\hat{b}_o$ is small, the actual closed-loop pole can be unstable, and far from 0.9.

It is hard to see how noise, modelling errors and the like will affect the ideal algorithm; however, the above example shows that, at the very least, caution should be exercised. One can also see that if the input u_j^* remains at 1 for a long period and then suddenly changes, the closed-loop system will, at least initially behave nothing like a constant system with pole at 0.9.

Example 5 Equation error scheme with RLS-type algorithm [1, 21].

The plant is still (3.1), now with $|a| < 1$ again, and the input is $u_k \equiv 1$. To secure an RLS algorithm with gains which do not go to zero, it is necessary to use the algorithm with exponential forgetting. One has

$$\begin{bmatrix} \hat{a}_k \\ \hat{b}_k \end{bmatrix} = \begin{bmatrix} \hat{a}_{k-1} \\ \hat{b}_{k-1} \end{bmatrix} + \frac{\rho\, P_{k-1}\psi_{k-1}}{\lambda+\rho\psi_{k-1}' P_{k-1}\psi_{k-1}} (y_k - \hat{y}_k) \tag{3.19}$$

with

$$P_k^{-1} = \lambda P_{k-1}^{-1} + \rho\, \psi_{k-1}\psi_{k-1}' \tag{3.20}$$

for some constant $\lambda < 1$, the forgetting factor being chosen constant for convenience. Here, $\psi_k = [y_k \; u_k]$

With $u_k \equiv 1$, $y_k \equiv b(1-a)^{-1}$. Let

$$T = \begin{bmatrix} b(1-a)^{-1} & 1 \\ 1 & -b(1-a)^{-1} \end{bmatrix}$$

and suppose $T P_o^{-1} T' = \gamma I$ for some positive γ.

Then, from (3.20)

$$T P_k^{-1} T' = \lambda T P_{k-1}^{-1} T' + \rho \begin{bmatrix} \{b^2(1-a)^{-2}+1\}^2 & 0 \\ 0 & 0 \end{bmatrix}$$

so that

$$T P_k^{-1} T = \begin{bmatrix} q_k & 0 \\ 0 & \lambda^k\gamma \end{bmatrix} \tag{3.21}$$

where

$$q_k = \lambda^k\gamma + \frac{1-\lambda^k}{1-\lambda}\rho\{b^2(1-a)^{-2}+1\}^2$$

Evidently, P_k becomes more and more ill-conditioned as time evolves. Its use in the update algorithm (3.19) becomes more and more susceptible to numerical error. At the same time, a simple calculation shows that for the ideal algorithm

$$T \begin{bmatrix} \hat{a}_k \\ \hat{b}_k \end{bmatrix}$$

has a first entry which converges exponentially fast and a second entry which is constant. Of course, noise and the like in practice will probably cause drifting of the second entry.

4. PERSISTENCY OF EXCITATION AND ROBUSTNESS

Suppose that one faced the task of estimating the coefficients of the transfer function of a system by using sine wave excitation at different frequencies, evaluating the transfer function at these different frequencies and inferring from these data the coefficients of the transfer function numerator and denominator. It is clear that a certain minimum number of sinusoids would have to be applied, depending on the number of unknown coefficients. One could also conceive of applying all the sinusoids at once, analysing the output into its different components and in this way identifying the transfer function. As well as requiring a certain minimum complexity of input for satisfactory identification, however, one would normally also need indefinite persistence of the input signal. It would not be satisfactory if the coefficients were slowly time-varying to apply the input signal for a finite interval only.

Roughly the same ideas, viz. that the input must be complex, to an extent dictated by the number of unknown coefficients and that it must persist, turn out also to be relevant in adaptive identification. The results go back at least to the mid-seventies [5,6,7], but for a more comprehensive treatment, see [8]. Now it is not of course necessary that an input be a linear combination of sinusoids, or periodic, for identification to take place. Accordingly, a time-domain condition is preferred for the specification of the input.

For a system with denominator degree n and with m numerator coefficients, we shall say that an input u_k satisfies a persistency of excitation condition if for all j, some S, some $\alpha_1 > 0$ and some $\alpha_2 > 0$,

$$\alpha_1 I \leq \sum_{k=j}^{j+S} \begin{bmatrix} u_k \\ u_{k-1} \\ \vdots \\ u_{k-m-n+1} \end{bmatrix} [u_k \; u_{k-1} \cdots u_{k-m-n+1}] \leq \alpha_2 I \tag{4.1}$$

(If u_k is a linear combination of at least $\frac{1}{2}(m+n)$ sinusoids this condition is guaranteed to hold. It will also hold for pseudorandom binary sequences of a sufficiently long period. It will not hold if u_k is a linear combination of fewer than $\frac{1}{2}(m+n)$ sinusoids where a DC level counts as half a sinusoid.)

The significance of this condition for identification is as follows: it is the key condition which ensures for equation error and output error identification, using SA-type or RLS-type algorithms with gains not going to zero, that the parameter estimation error approaches zero exponentially fast [9,10] (the time constant depends on e.g. α_1, α_2 and the system under consideration). As discussed in the next section, this implies various robustness properties. It certainly also gives robustness with respect to changes of input style and ensures that the error between the true plant output and model output goes to zero.

The condition is actually also necessary for exponentially fast convergence of the parameter error, which implies that there is no point in trying to relax it.

The condition is also relevant in adaptive pole positioning, where u_k should be replaced by the external input u_k^*. See [11] for an introduction to its relevance in this context.

What now of adaptive control problems such as reference trajectory following? It might be held that in order to solve an adaptive control problem one must have a model of the plant and therefore (4.1) is needed. Perhaps so, but the difficulty with (4.1) is that it involves the sequence $\{u_k\}$ which is not known a priori. It takes more work to obtain the result that in lieu of (4.1), one wants for all j, some S, some $\alpha_1>0$, some $\alpha_2>0$,

$$\alpha_1 I \leq \sum_{k=j}^{j+S} \begin{bmatrix} y_k^* \\ y_{k-1}^* \\ \vdots \\ y_{k-m-n+1}^* \end{bmatrix} [y_k^* \; y_{k-1}^* \cdots y_{k-m-n+1}^*] \leq \alpha_2 I \tag{4.2}$$

(There are additional conditions also needed - for example the Goodwin-Ramadge-Caines algorithm [3] requires the plant to have entirely stable zeros, while the Widrow algorithm [4] requires the plant to be stable and, in the first instance, to have no zeros, though this latter assumption can be relaxed.)

From an applications point of view (4.2) is disappointing. After all, one will often have reference trajectories y_k^* which comprise long intervals of constant signals and then (4.2) will not hold (unless perhaps S is taken to be extremely large). It appears to be standard folklore among users of adaptive controllers, that the adaption mechanism must be switched off in such long intervals, else unacceptable drifting of parameter estimates will occur which will manifest themselves as a possibly serious problem when a level change in y_k^* is effected. This is consistent with (4.2) being necessary as well as sufficient.

Equation (4.2), in retrospect, is entirely reasonable. Assuming that the control scheme converges, we must have $y_k \to y_k^*$ and so (4.2) would hold with y_k^* replaced by y_k. Then (4.2) amounts to the appropriate statement of (4.1) for the inverse of the plant rather than the plant itself. The inverse of course has the same number of unknown coefficients in its transfer function as the plant. It will therefore hardly be surprising to see (4.2) arising in the conditions for the adaptive linear-quadratic problem.

So far, we have stated conditions appropriate to deterministic inputs and reference trajectories. But what if in an identification problem the driving input is known simply to be gaussian random noise with a certain power spectrum? Then (4.1) certainly will not be applicable, since almost surely there will be an interval of length $S + m + n-1$ in which $|u_k| < \sqrt{\alpha_1}$, so that the lower bound will fail. (The upper bound also cannot hold for all j.) Yet common sense dictates that convergence should occur because of the obvious spectral richness of such a process. Indeed, exponential convergence results can be established; instead of (4.1), they require

$$0<\alpha_1 I \leq \sum \begin{bmatrix} u_k \\ u_{k-1} \\ \vdots \\ u_{k-m-n+1} \end{bmatrix} [u_k \; u_{k-1} \cdots u_{k-m-n+1}] \leq \alpha_2 I < \infty \tag{4.3}$$

for a stationary u_k process. (A more complicated condition appears possible for certain non-stationary processes, see e.g. [12-14] for an introduction to this type of analysis.) The analysis of the stochastic adaptive control problem [where an expectation is introduced into (4.2)] has not yet been completed, but of course there is little doubt as to the answer.

It is interesting to note that most, but not all, adaptive algorithms in their ideal forms are susceptible to the proof of a convergence result (typically to the effect that some tracking error goes to zero), irrespective of assumptions such as (4.1) - (4.3). However, there are two adaptive problems where no such result has been obtained, i.e. a persistency of excitation assumption is not just the way an algorithm is made exponentially convergent, but it is essential even to the ideal algorithm. These are the Widrow type reference trajectory following adaptive control scheme [15] and global adaptive pole placement (without persistency of excitation, only local results have been obtained, see e.g. [16]).

5. CONSEQUENCES OF THE PERSISTENCY OF EXCITATION CONDITIONS

We have just indicated that the persistency of excitation condition ensures, for the identification problem, exponentially fast convergence of the parameter estimates to their true values. In effect, the same is true for adaptive control problems. This conclusion of course applies to the ideal algorithm. What if the actual set-up is non-ideal in one of the ways described in the first section? The short answer is that if the departure from the ideal assumption is not too great, the departure from the ideal performance will not be too great also. However, when the ideal assumptions are departed from significantly, performance may collapse, e.g. by the appearance of some instability. In this section, we aim to indicate the general thinking behind this conclusion.

If one writes down a set of state variable equations for the system comprising the unknown plant and the adaptive identifiers and controllers, assuming no departures from ideality, one gets a set of equations, virtually always time-varying, usually nonlinear, of the form

$$\xi_{k+1} = f(\xi_k, z_k^*, p, k) \tag{5.1}$$

where ξ_k is the state vector, z_k^* denotes a persistently exciting signal (it may be an input or a reference trajectory) and p denotes a parameter vector (typically the coefficients of the plant transfer function).

Equation (5.1) is known to have an exponential convergence property. Now if one postulates departures from ideality, one can build these into the equations with the result that (5.1) may be replaced by

$$\xi_{k+1} = \tilde{f}(\xi_k, \tilde{z}_k^*, p_k, k) + \tilde{g}(\xi_k, \tilde{z}_k, p_k, k) \tag{5.2}$$

where now $\tilde{z}_k^*$ is like z_k^*, $\tilde{f}$ is like f and is zero when $\xi_k=0$, and $\tilde{g}(\xi_k, \tilde{z}_k^*, p_k, k)$ is small. (Of course, one can be more precise about what one means by 'like' and 'small'.)

Now these variations on (5.1) are like a number considered in texts on stability [17,18]. In particular, letting p become time-varying does not disturb exponential stability if the variation is slow enough, $\tilde{f}$ close enough to f will also not disturb the property, while small $\tilde{g}$, if small enough, will cause ξ_k to decay to a small ball containing the origin, the radius of which depends on g.

In outline, we are saying that if there are mistakes in the modelling assumptions (nonlinearity or high order modes being overlooked), or measurement noise introduced, or plant parameter variation taking place, one can appeal to known results on stability theory or extensions of them to conclude that, when the ideal algorithm is exponentially convergent, the nonideal algorithm will behave satisfactorily if the departure from the ideal is not great. By "behave satisfactorily" we mean that asymptotic parameter estimates will not be too much in error, the error in tracking a reference trajectory will not be too great, etc. The details of the calculation are often long and involved. For examples, see [19,20].

It might well be conjectured that mere asymptotic stability rather than exponential stability is all that will be needed to secure robust behaviour. This is certainly not so (and can be proven). It is however, instructive to look at a simple example. The equation

$$x_{k+1} = (1- \frac{1}{k+1})x_k \qquad k=1,2, \ldots \qquad (5.3)$$

ensures that $x_k = x_1 k^{-1}$, and so clearly $x_{k+1} \to 0$, i.e. the equation is asymptotically stable, but x_k does not decay exponentially fast. Suppose now a DC offset is introduced

$$x_{k+1} = (1- \frac{1}{k+1})x_k + n \qquad (5.4)$$

If this is to behave in a stable fashion, x_k would have to approach a steady state value, $\bar{x}$ say. But it is easy to check that no steady state value satisfies this equation - in fact x_k is unbounded. On the other hand, $x_{k+1} = 0.9\, x_k$ is exponentially stable, and a DC offset will produce a bounded change in the x_k trajectory and its steady state values.

6. GAPS IN OUR KNOWLEDGE

It cannot be stressed enough that the full spectrum of adaptive problems of potential application interest is still largely unaddressed. Let us note several examples:

- systems with time delay, especially variable time delay. (Setting up a discrete-time model may provide a solution, but only sometimes.)
- systems with a state-variable description of known structure, with a few unknown parameters
- most nonlinear systems
- most distributed parameter systems

Also, in respect of those problems for which advances have been made, there are aspects concerning which more quantitative data would be very helpful, for example:

- speed of convergence of the adaptive algorithm, including its relation to the external driving input or reference trajectory
- effect of noise

7. REFERENCES

1. LANDAU, Y.D. (1979). Adaptive Control. New York, Dekker.
2. GOODWIN, G.C. and SIN, K.S. (1982). Adaptive filtering, prediction and control. Prentice-Hall, to appear.
3. GOODWIN, G.C., RAMADGE, P.J. and CAINES, P.E. (1980). Discrete-time multivariable adaptive control. IEEE Trans. Autom.Contr., Vol. AC-25, pp. 449-454.
4. WIDROW, B., McCOOL, J.M. and MEDOFF, B.P. (1978). Adaptive control by inverse modelling. Proc. 1978 Asilomar Conf. on Circuits, Systems and Computers.
5. ANDERSON, B.D.O. (1974). Adaptive identification of multiple-input, multiple-output plants. Proc. 13th IEEE Conf. on Dec. and Control, pp. 273-281.
6. MORGAN, A.P. and NARENDRA, K.S. (1977). On the stability of nonautonomous differential equations $\dot{x} = [A + B(t)]x$, with skew symmetric matrix B(t). SIAM Jnl. on Control and Optimization. Vol. 15, No.1, pp. 163-176.
7. YUAN, J.S.C. and WONHAM, W.M. (1977). Probins Signals for Model Reference Identification IEEE Trans. Autom.Contr.,Vol. AC-22, pp. 530-538.
8. ANDERSON, B.D.O. and JOHNSON, C.R.(Jr.) (1982). Exponential convergence of adaptive identification and control algorithms. Automatica, to appear.
9. JOHNSON, C.R. (Jr.) and BITMEAD, R.R. (1982). Proof of the exponential convergence of RLS with exponential data forgetting. Proc. Conf. on Info. Sciences and Systems. Johns Hopkins.
10. JOHNSTONE, R.M., JOHNSON, C.R.(Jr.), BITMEAD, R.R. and ANDERSON, B.D.O. (1982). Exponential convergence of recursive least squares with exponential forgetting factor. Submitted for publication.
11. JOHNSTONE, R.M. and ANDERSON, B.D.O. (1982). Global adaptive pole placement : Detailed analysis of a first order system. Submitted for presentation at the 21st IEEE Conf. on Dec. and Control.
12. BITMEAD, R.R. and ANDERSON, B.D.O. (1979). Exponentially convergent behaviour of simple stochastic estimation algorithms. Proc. 17th IEEE Conf. on Dec. and Control, pp. 580-585.
13. BITMEAD, R.R. and ANDERSON, B.D.O. (1980). Lyapunov techniques for the exponential

stability of linear difference equations with random co-efficients. IEEE Trans. Autom. Contr., Vol.AC-25, pp. 782-788.

14. BITMEAD, R.R. and ANDERSON, B.D.O. (1980). Performance of adaptive estimation algorithms in dependent random environments. IEEE Trans. on Autom. Contr., Vol. AC-25, pp. 788-794.

15. ANDERSON, B.D.O. and JOHNSTONE, R.M. (1982). Convergence results for Widrow's adaptive controller. Proc. 6th IFAC Symp. on Identification and Sys. Parameter Estimation, to appear.

16. GOODWIN, G.C. and SIN, K.S. (1981). Adaptive control of nonminimum phase systems. IEEE Trans. Autom. Contr., Vol. AC-26, pp. 478-483.

17. HAHN, W. (1967). Stability of Motion. Berlin, Springer Verlag.

18. DESOER, C.A. and VIDYASAGAR, M. (1975). Feedback Systems : Input-Output Properties. New York, Academic Press.

19. JOHNSON, C.R. (Jr.) and ANDERSON, B.D.O. (1982). Reduced order adaptive output error identification and adaptive IIR filtering. IEEE Trans. Autom. Contr., to appear.

20. ANDERSON, B.D.O. and JOHNSTONE, R.M. (1981). Robust Lyapunov stability results and adaptive systems. Proc. 20th IEEE Conf. on Dec. and Control, San Diego.

21. JOHNSTONE, R.M. and ANDERSON, B.D.O. (1982). Exponential convergence of recursive least squares with exponential forgetting factor-adaptive control. Submitted for publication.

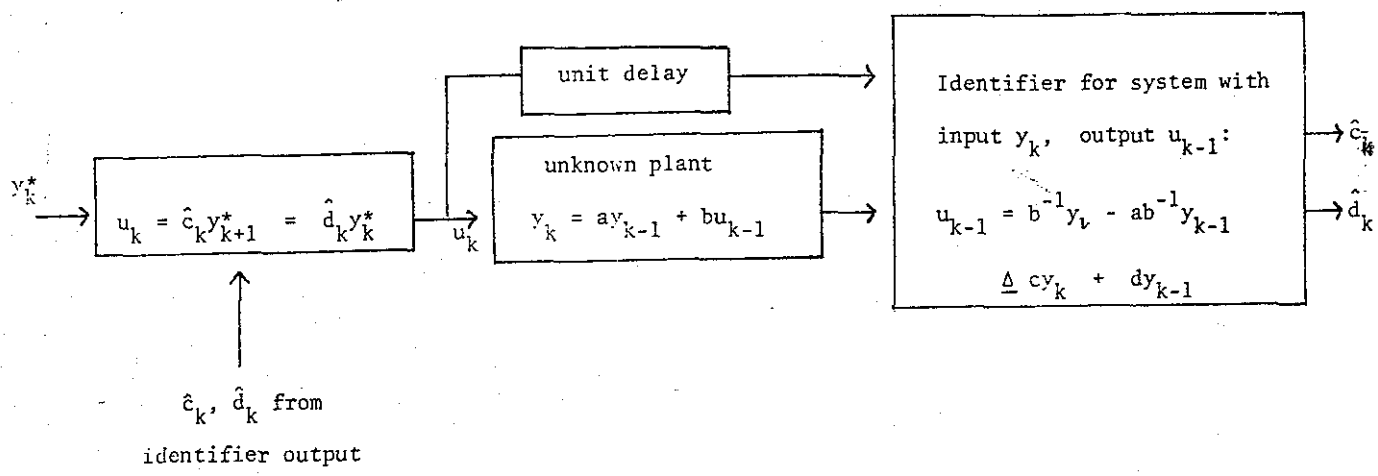

Figure 1 Illustration of reference trajectory following adaptive controller