APPLICATION OF NUMERICAL METHODS TO THE ACCELERATION OF THE CONVERGENCE OF THE ADAPTIVE CONTROL ALGORITHMS

THE ONE-DIMENSIONAL CASE

J. J. MIÑAMBRES and M. DE LA SEN

Departmento de Fisica, Facultad de Ciencias, Universidad del Pais Vasco, Leioa (Vizcaya), Apartado 644, Bilbao, España

(Received February 1986)

Communicated by E. Y. Rodin

Abstract—This paper applies numerical methods of acceleration of the convergence to the case of one parameter in adaptive control algorithms.

1. INTRODUCTION

The purpose of this paper is the application of numerical methods of acceleration of the convergence of sequences to the adaptive algorithms, when there is only one unknown parameter, as a first step towards presenting more general results. Obviously, it is claimed that this application achieves a faster convergence of the adaptive algorithms.

First, we will try to apply Aitken’s $\Delta^2$-method [1]. This is a quasi-Newton method available to accelerate one-dimensional sequences. It will be shown that the attempt raises unsolvable difficulties. Fortunately, Steffensen’s method [1] will be shown to solve them. However, application of the method to the adaptive control will be seen to raise new difficulties—due to the non-stationary characteristics of the iteration function. We will have to reformulate the method. So, we will translate all the developments and proofs [1, 2] from Newton’s method to the Steffensen one. This translation will be made taking into account the fact that the iteration function is not stationary.

Finally, some simulations will be made by considering the problem of controlling pH in acidic wastewater [3].

2. NUMERICAL METHODS

Let $\{X_t\}$ be a one-dimensional real sequence generated by

$$X_{t+1} = g(X_t) \quad t = 0, 1, 2 \ldots ,$$

with $X$ an arbitrary starting value. If this sequence converges to $X^*$ then the sequence $\{X'_t\}$, given by

$$X'_{t+1} = X_t - \frac{(X_{t+1} - X_t)^2}{X_{t+2} - 2X_{t+1} + X_t},$$

converges to $X^*$ faster than $\{X_t\}$.

Equation (2) is known as Aitken’s $\Delta^2$-method. When trying to apply this method to the normal adaptive algorithms, when there is only one unknown parameter, obviously, the one-dimensional sequence that we want to accelerate is the sequence of estimated parameters $\{\theta_t\}$, which converges to a value $\theta^*$. So, expression (1) will be the adaptive algorithm to be considered.
In the adaptive control case, equation (2) can be rewritten as follows:

\[
\hat{\theta}_{i+1} = \hat{\theta}_i - \frac{(\hat{\theta}_{i+1} - \hat{\theta}_i)^2}{\hat{\theta}_{i+1} - 2\hat{\theta}_i + \hat{\theta}_i},
\]

(3)

Consider the problems that equation (3) implies:

1. To obtain \( \hat{\theta}_{i+1} \), and so obtain the control law at time \( t + 1 \), we must know the values of the estimated parameter at time \( t + 2 \). To solve this problem we could consider retarded adaptive algorithms. But this would probably imply the loss of the advantages of the acceleration.

2. To obtain the accelerated sequence \( \{\hat{\theta}_i\} \) we must know \( \{\theta_i\} \). But \( \{\theta_i\} \) is the sequence issued when we use an adaptive algorithm without any modification. What happens if we use the accelerated parameter \( \hat{\theta}_i \) to generate the control law at time \( t \)? Obviously, we will have modified the adaptive algorithm; so, we will not be able to obtain posterior values of the sequence \( \{\theta_i\} \). Therefore, if the estimated parameter \( \theta_i \) depends on \( u_{i-d}, u_{i-d-1}, \ldots \), then Aitken’s \( \Delta^2 \)-method would be applicable for only \( d - 1 \) steps.

These are the difficulties Aitken’s \( \Delta^2 \)-method causes.

Now, let us consider Steffensen’s method. This method consists of the iterative application of the expression

\[
X_{i+1} = X_i - \frac{[g(X_i) - X_i]^2}{g[g(X_i)] - 2g(X_i) + X_i},
\]

(4)

where the iteration function \( g \) is such that

\[
g(X_i) = X_{i-1}.
\]

(5)

In the adaptive control case, considering the Goodwin \textit{et al.} [4] algorithm (GRC), formulae (4) and (5) can be rewritten as

\[
\hat{\theta}_{i+1}^\kappa = \hat{\theta}_i^\kappa - \frac{[g(\hat{\theta}_i^\kappa) - \hat{\theta}_i^\kappa]^2}{g(g(\hat{\theta}_i^\kappa) - 2g(\hat{\theta}_i^\kappa)) + \hat{\theta}_i^\kappa},
\]

(6)

and

\[
g(\hat{\theta}_{i-1}^\kappa) = \hat{\theta}_{i-1}^\kappa = \hat{\theta}_{i-1}^\kappa + \frac{x_i\phi_{i-d}(y_i - \hat{\theta}_{i-1}^\kappa \phi_{i-d})}{c_i + \phi_{i-d}^\kappa \phi_{i-d}},
\]

(7)

where

\[
\hat{\theta}_{i-1}^\kappa = (\theta^{*1}, \ldots, \theta^{*k-1}, \hat{\theta}_{i-1}^\kappa, \theta^{*k-1}, \ldots, \theta^{**}).
\]

What is the mean difference between this method and Aitken’s \( \Delta^2 \)-method? It is that we no longer need to know the sequence \( \{\theta_i\} \) issued to a normal adaptive control. So, if we use the accelerated sequence of estimated parameters to generate the control law we need not worry about the fact that the modification originates from posterior values of the estimated parameters. The method uses the modified values.

We have solved the second problem that Aitken’s \( \Delta^2 \)-method implied. However, a new difficulty arises. Looking at formula (7) we can observe that the iteration function \( g \) is not stationary. Actually, at time \( t \), \( g \) depends on the time-variant parameters \( a_i, \phi_{i-d}, y_i \), and \( c_i \). As Steffensen’s method supposes \( g \) is a stationary function, it must be reformulated.

Starting with Newton’s method:

\[
X_{i+1} = X_i - \frac{f_i(X_i)}{f_i'(X_i)},
\]

(8)
where \( f_i \) is a time-variant function. Approximating \( f'(.)(X_i) \) by the difference quotient

\[
\frac{f_i(X_{i-1}) - f_i(X_i)}{X_{i-1} - X_i}
\]

yields the iterative procedure

\[
X_{i+1}' = X_i - f_i(X_i) \frac{X_{i-1} - X_i}{f_i(X_{i+1}) - f_i(X_i)}
\]

But

\[
f_i(X_i) = X_i - g_i(X_i),
\]

so

\[
f_i(X_{i-1}) = X_{i+1} - g_i(X_{i+1}),
\]

where

\[X_{i-1} = g_i(X_i).\]

This yields

\[
X_{i+1}' = X_i - \left[ X_i - g_i(X_i) \right] \frac{g_i(X_i) - X_i}{g_i(X_i) - g_i[g_i(X_i)] - X_i + g_i(X_i)}
\]

\[= X_i - \frac{[g_i(X_i) - X_i]^2}{g_i[g_i(X_i)] - 2g_i(X_i) - X_i}.\]  

Equation (13) is Aitken's method for non-stationary functions.

Steffensen's method will be

\[
X_{i+1} = X_i - \frac{[g_i(X_i) - X_i]^2}{g_i[g_i(X_i)] - 2g_i(X_i) - X_i + g_i(X_i)} = X_i - \frac{X_i[g_i(X_i)] - [g_i(X_i)]^2}{X_i - 2g_i(X_i) + g_i[g_i(X_i)]}.\]

In the adaptive case, considering a component of the vector of estimated parameters:

\[
\frac{\theta^k_i}{\theta^k_{i-1}} = \frac{\theta^k_{i-1} g_i(\theta^k_{i-1}) - [g_i(\theta^k_{i-1})]^2}{\theta^k_{i-1} - 2g_i(\theta^k_{i-1}) - 2g_i(\theta^k_{i-1}) + g_i[g_i(\theta^k_{i-1})]}.\]

To obtain \( \theta^k_i \) we do not need a retarded algorithm.

Appendix A proves the convergence of the accelerated sequence generated by equation (15) to the same value as the unaccelerated sequence. In addition, it proves that the accelerated sequence converges faster than the original one.

Considering the GRC algorithm, we saw [equation (7)] that

\[
\theta_i^k = \theta_{i-1}^k + \frac{a_i \phi_{i-d}^k (y_i - \phi_{i-d}^k \theta_{i-1})}{c_i + \phi_{i-d}^k \phi_{i-d}},
\]

where

\[
\theta_{i-1} = (\theta^{*1}, \ldots, \theta^{*i-1}, \theta_i^k, \theta^{**}).
\]

Taking into account equation (17), equation (16) can be rewritten in vectorial form:

\[
\theta_i = \theta_{i-1} + \frac{a_i \phi_{i-d} (y_i - \phi_{i-d} \theta_{i-1})}{c_i + \phi_{i-d} \phi_{i-d}},
\]

where

\[
\phi_{i-d}^k = (0, \ldots, 0, \phi_{i-d}^k, 0, \ldots, 0).
\]

Assuming \( \phi_{i-d}^k \neq 0 \)—this condition ensures the denominator of equation (15) \( \neq 0 \)—expressions (15), (16) and (18) yield (Appendix B)

\[
\theta_i^k = \theta_{i-1} - \frac{y_i - \phi_{i-d}^k \theta_{i-1}}{\phi_{i-d}^k}.\]
Developing $\phi^{T}_{t-d} \hat{\theta}_{t-1}$,

$$\hat{\theta}^k = \hat{\theta}^k_{t-1} - \frac{1}{\phi^{k}_{t-d}} (y_t - \phi^{T}_{t-d} \theta^* - \cdots - \phi^{k}_{t-d} \hat{\theta}^k_{t-1} - \cdots - \phi^{*}_{t-d} \theta^*) \quad (21)$$

But

$$y_t = \phi^{T}_{t-d} \theta^* = \phi^{T}_{t-d} \theta^* + \cdots + \phi^{*}_{t-d} \theta^* \quad (22)$$

So, if $\phi^{*}_{t-d} \neq 0$,

$$\hat{\theta}^k_{t-1} - \frac{1}{\phi^{k}_{t-d}} [\phi^{*}_{t-d} (\theta^* - \hat{\theta}^k_{t-1})] = \theta^* \quad (23)$$

We have obtained a one-dimensional identifier. Assuming $\phi^{*}_{t-d} \neq 0$, a single application of the accelerated algorithm will yield the true value of the estimated parameter.

3. SIMULATION RESULTS

We consider the problem of controlling acidity in a continuous flow of industrial wastewater using a non-linear adaptive controller.

The model that we consider is that described by McAvoy et al. and used by Goodwin and Sang Sin [3]. A strong acid flows into a tank where it is thoroughly mixed with a strong base whose inward rate of flow is controlled in such a way as to produce a neutral outward flow from the tank. Because the acid and the base are strong, each is completely dissociated; also the dissociation of the water can be disregarded. This model can be described by the following equation:

$$\nu \frac{dy}{dt} = F_t (a - y_t) - u_t (b + y_t),$$

where

$y_t = [H^+] - [OH^-]$, the distance from neutrality,
$V$ = volume of the tank,
$F_t$ = rate of flow of the acid,
$a$ = concentration of the acid,
$u_t$ = rate of flow of the base
and
$\nu$ = concentration of the base.

The distance from neutrality, $y_t$, can be determined from the pH value, $P_t$, using the following non-linear transformation:

$$y_t = 10^{-p_t} - 10^{p_t} K_w,$$

where $K_w$ is the water equilibrium constant

$$K_w = 10^{-14} \text{g-ion/l}.$$

We suppose that $b$ is fixed and known, that $a$ is fixed, that $F_t$ can be measured on line and that $u_t$ can be given assigned values within certain limits.

An approximate discrete-time model, incorporating measurement and input actuator errors, can be developed as follows:

$$y_{t+1} = y_t + \frac{T}{V} [F_t (a - y_t) - u_t (b + y_t)] = \phi^T_t \theta_0,$$

where $T$ is the sampling interval and

$$\phi^T_t = [y_t, F_t, y_t - u_t (b + y_t), F_t]$$
Convergence of the adaptive algorithms

and

\[ \theta^*_n = [\theta_n^* \cdot \theta_n^* \cdot \theta_n^*] = \left[ 1 \quad \frac{T}{\bar{v}} \quad \frac{aT}{\bar{v}} \right] \]

In general, the control objective would be the estimation of the whole vector \( \theta_0 \). However, we suppose that two of the three parameters are known—this is due to the one-dimensional character of the developed method. We will apply this method of acceleration of convergence to the unknown parameter.

For a desired output sequence \([y_{n*}; \ t = t_0 + nT]\) we define the input \( u_n \),

\[ u_n = \frac{\theta^1_n y_n - \theta^2_n F_1 y_n + \theta^3_n F_1 - y_{n*}^*}{\theta^2_n (b + y_n)} \]

and choose

\[ u_n = \begin{cases} 
    u_n^0 & \text{if } 0 < u_n^0 \leq u_{\text{max}} \\
    u_{\text{max}} & \text{if } u_n^0 > u_{\text{max}} \\
    0 & \text{if } u_n^0 \leq 0.
\end{cases} \]

We assume that \( \theta^2 \) and \( \theta^3 \) are known, so we obtain

\[ u_n = \frac{\theta^1_n y_n - \theta^2_n F_1 y_n + \theta^3_n F_1 - y_{n*}^*}{\theta^2_n (b + y_n)} \]

Referring to \( y_{n*} \), taking into account \( y_n \), is the distance from neutrality, we have \( y_{n*} = 0 \).

Some simulation results are presented in Figs 1 and 2. The following values were adopted:

\[ V = 1 \text{ l}, \]
\[ T = 9 \text{ s}, \]
\[ a = 2 \text{ mol/l}, \]
\[ 0 \leq u_n \leq 4 \text{ l/min} \]

Fig. 1

![Diagram](image-url)
The acid flow rate was varied in a slow sine function given by

\[ F_r = 0.1125 + 0.0125 \sin \frac{\pi t}{25} \text{ l/min}. \]

These simulations were made by using the GRC algorithm. Similar results can be obtained using Lozano’s algorithm and the least-square one.

Figure 1 has been obtained for the initial value of \( \theta^0 \), \( \theta^0 = 10 \) and \( c, = c, = 10 \). Figure 2 has been obtained for \( \theta^0 = 5 \) and \( c, = c = 0.085 \).

It can be seen that the accelerated algorithm converges faster than the GRC one. Also, the convergence of the accelerated algorithm does not depend on the values \( \theta^0 \) and \( c \). This is in accordance with the fact that we have obtained a one-dimensional identifier.

Note that the output sequence \( y_i \) needs approx. 10 samples to track the desired output sequence \( y_i^* \). This is because the model considered is non-linear. So, although the parameter \( \theta^1 \) is identified in a sample, the convergence of \( y_i \) is not as fast.

4. CONCLUSIONS

An application of numerical methods of acceleration of the convergence of sequences to the adaptive algorithms has been developed in the one-dimensional case. Its purpose is to improve the rate of convergence of these algorithms. The difficulties involved in this attempt necessitated reformulating the numerical method, accommodating it to our needs. The application of this new method to the adaptive control has resulted in a one-dimensional identifier.

Simulation results are in accordance with this result. They show the faster convergence of the new algorithm and the independence of the convergence with respect to \( c \) and the initial value of the parameter to be estimated.
REFERENCES


APPENDIX A

Starting with Steffensen's method obtained for non-stationary functions:

\[ x'_{i+1} = x'_i - \frac{[g'_i(x'_i) - x'_i]^2}{g'_i[x'_i] - 2g'_i(x'_i) + x'_i} \]  

(A.1)

Assuming \( g_i \) is continuous in \([x_i, g_i(x_i)]\) and the derivative in \((x_i, g_i(x_i))\), Lagrange's half-value theorem [2] yields:

\[ g'_i(x'_i) - g'_i(x_i) = \frac{1}{2} g_i(x_i) [g_i(x'_i) - x'_i - (g_i(x_i) - x_i)]. \]  

(A.2)

where

\[ \xi \in [x_i, g_i(x_i)]. \]

So,

\[ g_i(x'_i) - 2g_i(x_i) + x_i = g'_i(\xi) [g_i(x'_i) - x'_i] - [g'_i(\xi) - 1] [g_i(x_i) - x_i]. \]  

(A.3)

Substituting equation (A.3) in equation (A.1), we obtain

\[ x'_{i+1} = x'_i - \frac{[g'_i(x'_i) - x'_i]^2}{[g'_i(x'_i) - 1] [g_i(x_i) - x'_i]} = x'_i - \frac{g_i(x_i) - x'_i}{g'_i(x'_i) - 1} \]

For

\[ A_i = g'_i(\xi) - 1, \]

we have

\[ x'_{i+1} = x'_i - \frac{g_i(x_i) - x'_i}{A_i}. \]  

(A.4)

Consider the limit

\[ \lim_{i \to \infty} x'_{i+1} - x^* \]

(A.5)

We know that \([x_i]\) converges to \(x^*\). From equations (A.4) and (A.5) we obtain

\[ x'_i - \frac{g_i(x_i) - x'_i}{A_i} - x^* \]

(A.6)

Suppose expression (A.6) \( \neq 0 \). Then,

\[ \forall \epsilon > 0, \exists M_1, \epsilon \in \mathbb{N}, \text{ such that } \frac{|A_i x'_i - [g_i(x'_i) - x'_i] - A_i x^*|}{A_i [g_i(x'_i) - x^*]} < \epsilon \text{ } 2. \text{ } \forall i \geq M_1. \]

so

\[ \forall \epsilon > 0, \exists M_1, \epsilon \in \mathbb{N}, \text{ such that } |A_i (x'_i - x^*) - [g_i(x'_i) - x'_i]| < \epsilon/2 |A_i||g_i(x'_i) - x^*|. \text{ } \forall i \geq M_1. \]

Since

\[ |g_i(x'_i) - x'_i| - |A_i||x'_i - x^*| \leq |A_i(x'_i - x^*) - [g_i(x'_i) - x'_i]|. \]

we have

\[ |g_i(x'_i) - x'_i| - |A_i||x'_i - x^*| < \epsilon/2 |A_i||g_i(x'_i) - x^*|. \text{ } \forall i > M_1. \]

So

\[ \forall \epsilon > 0, \exists M_1, \epsilon \in \mathbb{N}, \text{ such that } |g_i(x'_i) - x'_i| - |A_i||x'_i - x^*| < \epsilon/2 |A_i||g_i(x'_i) - x^*| - |A_i||x'_i - x^*|. \text{ } \forall i \geq M_1. \]

The sequence \([x_i] \) converges to \(x^*.\) This fact implies

\[ \forall K > 0, \exists M'_1, \epsilon \in \mathbb{N}, \text{ such that } |x'_i - x^*| < \epsilon/2 K. \text{ } \forall i \geq M'_1. \]

\[ \exists M^*_1, \epsilon \in \mathbb{N}, \text{ such that } |g_i(x'_i) - x^*| < \epsilon K. \text{ } \forall i \geq M^*_1. \]

Let \( M'_1 = \max(M'_1, M^*_1). \) With this value of \( M'_1, \) the two expressions above will be verified, this is also true with \( M = \max(M'_1, M^*_1). \) However, \( M_1 \) does not exist, so neither does \( M. \)
\[ |g_{i}(X_{i}) - X_{i}| < \varepsilon \quad \forall \varepsilon > 0, \quad \forall M \in \mathbb{N} \]

If \(|A| = 1\) then \(|A| = K\). \(\forall \varepsilon > 0, \forall M \in \mathbb{N}\) such that
\[ |g_{i}(X_{i}) - X_{i}| < \varepsilon \quad \forall \varepsilon > 0, \forall M \in \mathbb{N}\].

This implies that \(|A|\) is bounded from a certain value of \(\varepsilon\). Under these conditions we have:
\[ \forall \varepsilon > 0, \forall M \in \mathbb{N} \quad \text{such that} \quad |g_{i}(X_{i}) - X_{i}| < \varepsilon \quad \forall \varepsilon > 0, \forall M \in \mathbb{N}\].

This expression contradicts the fact that \(\{X,\Omega\}\) converges because it does not verify Cauchy's criterion of convergence [2]. So,
\[ \lim_{\varepsilon \to 0} X_{i+1} - X^* = 0. \quad (A.7) \]

**APPENDIX B**

Considering the GRC algorithm [4], we saw that
\[ g_{i+1}(\delta_{i+1}) = \delta_{i+1} + \frac{a_i \phi_{i+1}(y_i - \Phi_{i+1} \delta_{i+1})}{c_i + \Phi_{i+1} \phi_{i+1}}. \quad (B.1) \]

Hence
\[ g_{i+1}(g_{i+1}(\delta_{i+1})) = g_{i+1}\left[ \delta_{i+1} + \frac{a_i \phi_{i+1}(y_i - \Phi_{i+1} \delta_{i+1})}{c_i + \Phi_{i+1} \phi_{i+1}} \right] \]
\[ = \delta_{i+1} + \frac{a_i \phi_{i+1}(y_i - \Phi_{i+1} \delta_{i+1})}{c_i + \Phi_{i+1} \phi_{i+1}} \]
\[ + \frac{a_i \phi_{i+1} \left[ y_i - \Phi_{i+1} \delta_{i+1} \right]}{c_i + \Phi_{i+1} \phi_{i+1}} \]
\[ = \delta_{i+1} + \frac{a_i \phi_{i+1}(y_i - \Phi_{i+1} \delta_{i+1})}{c_i + \Phi_{i+1} \phi_{i+1}}. \]

Since
\[ \Phi_{i+1} \phi_{i+1} = (0, \ldots, 0, \phi_{i+1} \phi_{i+1}, 0, \ldots, 0), \]
\[ g_{i+1}(g_{i+1}(\delta_{i+1})) = \delta_{i+1} + \frac{2a_i \phi_{i+1}(y_i - \Phi_{i+1} \delta_{i+1})}{c_i + \Phi_{i+1} \phi_{i+1}}. \quad (B.2) \]

Development of \([g_{i+1}(\delta_{i+1})]^2\) leads to
\[ [g_{i+1}(\delta_{i+1})]^2 = \delta_{i+1} + \frac{2a_i \phi_{i+1} \left[ y_i - \Phi_{i+1} \delta_{i+1} \right]}{c_i + \Phi_{i+1} \phi_{i+1}} + \frac{[a_i \phi_{i+1}(y_i - \Phi_{i+1} \delta_{i+1})]^2}{(c_i + \Phi_{i+1} \phi_{i+1})^2}. \quad (B.3) \]

From equations (B.2) and (B.3) we obtain
\[ \delta_{i+1} - 2g_{i+1}(\delta_{i+1}) + g_{i+1}(g_{i+1}(\delta_{i+1})) = \frac{a_i \phi_{i+1}(y_i - \Phi_{i+1} \delta_{i+1})}{(c_i + \Phi_{i+1} \phi_{i+1})^2} \cdot \delta_{i+1} - 2g_{i+1}(\delta_{i+1}). \quad (B.4) \]

and
\[ \delta_{i+1} - 2g_{i+1}(\delta_{i+1}) + g_{i+1}(g_{i+1}(\delta_{i+1})) = \frac{a_i \phi_{i+1}(y_i - \Phi_{i+1} \delta_{i+1})}{(c_i + \Phi_{i+1} \phi_{i+1})^2}. \quad (B.5) \]

Assuming \(\delta_{i+1} \neq 0\), we obtain from equations (B.4) and (B.5)
\[ \delta_{i+1} = \frac{\delta_{i+1} - 2g_{i+1}(\delta_{i+1}) + g_{i+1}(g_{i+1}(\delta_{i+1}))}{\delta_{i+1} - 2g_{i+1}(\delta_{i+1}) + g_{i+1}(g_{i+1}(\delta_{i+1}))}. \quad (B.6) \]