Preliminary results for ARIMA adaptation algorithms with time-varying matrix adaptation gain
Ioan Doré Landau, Tudor-Bogdan Airimitoaie, Bernard Vau, Gabriel Buche

To cite this version:
Ioan Doré Landau, Tudor-Bogdan Airimitoaie, Bernard Vau, Gabriel Buche. Preliminary results for ARIMA adaptation algorithms with time-varying matrix adaptation gain. 2023. hal-04045870

HAL Id: hal-04045870
https://hal.science/hal-04045870
Preprint submitted on 25 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract

This note introduces two ARIMA type adaptation/learning algorithms that use a time-varying matrix adaptation gain. Simulations illustrate the performance of one of the algorithms.

1 Algorithm I

The algorithm of Eq. (16) from [1] is replaced by:

\[
\hat{\theta}(t+1) = H_{PAA}(q^{-1})[F(t)\phi(t)\epsilon(t+1)] = \hat{\theta}(t) + H_{DAC}(q^{-1})[F(t)\phi(t)\epsilon(t+1)]
\]  

(1)

The explicit form of the adaptation algorithm becomes:

\[
\hat{\theta}(t+1) = d_1\hat{\theta}(t) + d_2\hat{\theta}(t-1) + \ldots + F(t)\phi(t)\epsilon(t+1) + c_1F(t-1)\phi(t-1)\epsilon(t) + c_2F(t-2)\phi(t-2)\epsilon(t-1) + \ldots
\]  

(2)

The expression of the matrix adaptation gain is given by:

\[
F(t+1) = \frac{1}{\lambda_1(t)} \left[ F(t) - \frac{F(t)\phi(t)\phi^T(t)F(t)}{\lambda_1(t) + \phi^T(t)F(t)\phi(t)} \right]
\]  

(3)

where \(\lambda_1(t)\) and \(\lambda_2(t)\) allow to obtain various profiles for the adaptation gain \(F(t)\) (see [2]). \(\phi(t)\) is given by Eq. (3) from [1] and \(\epsilon(t+1)\) is given by:

\[
\epsilon(t+1) = \frac{\epsilon^0(t+1)}{1 + \phi^T(t)F(t)\phi(t)}
\]  

(4)

where \(\epsilon^0(t+1)\) is given by:

\[
\epsilon^0(t+1) = y(t+1) - \hat{\theta}_0^T(t)\phi(t)
\]  

(5)

and \(\hat{\theta}_0(t)\) is given by:

\[
\hat{\theta}_0(t) = d_1\hat{\theta}(t) + d_2\hat{\theta}(t-1) + \ldots + c_1F(t-1)\phi(t-1)\epsilon(t) + c_2F(t-2)\phi(t-2)\epsilon(t-1) + \ldots
\]  

(6)
2 Simulation results

The system considered in [1] for simulation is used also for the case of time-varying matrix adaptation gain. A pseudo random binary sequence (PRBS) is used as excitation signal. The least squares algorithm is used for updating the adaptation gain \( F(t) (\lambda_1(t) = \lambda_1 = 1, \lambda_2(t) = \lambda_2 = 1) \). The initial parameter estimates are set to 0 (initial squared parameter distance = 4). Figure 1 shows the results of the simulation (zoom) for the classical recursive least square algorithm \( H_{\text{DAG}}^1 = 1 \), i.e. \( c_1 = c_2 = d'_1 = 0 \) and the ARIMA2 algorithm \( (c_1 = 0.5, c_2 = 0.05, d'_1 = 0.5) \) using recursive least squares algorithm for updating the matrix adaptation gain \( F(t) \). The case without measurement noise and the case when a white noise is added on the output (signal/noise ratio (standard deviation) = 33 dB) have been considered. In both situations the ARIMA2 algorithm provides better adaptation transients.

![Figure 1: Evolution of the squared parametric distance (zoom) using ARIMA2 algorithm with least squares adaptation gain and the standard least squares algorithm \( (c_1 = c_2 = d'_1 = 0) \)](image)

3 Algorithm II

The algorithm given in Eq. (7) is replaced by:

\[
\hat{\theta}(t+1) = \hat{\theta}(t) + F(t)H_{\text{DAG}}(q^{-1})[\phi(t)e(t+1)]
\]  

(7)
i.e. the dynamic adaptation gain acts only on the gradient (or its approximation). The explicit form of the adaptation algorithm becomes:

\[
\hat{\theta}(t+1) = d_1\hat{\theta}(t) + d_2\hat{\theta}(t-1) + \ldots \\
+ F(t)[\phi(t)\epsilon(t + 1) + c_1\phi(t - 1)\epsilon(t) + c_2\phi(t - 2)\epsilon(t - 1) + \ldots] 
\] (8)

where \( F(t) \) and \( \epsilon(t + 1) \) are given respectively by Eqs. (3) and (4). \( \epsilon^o \) is given by Eq. (5) where \( \hat{\theta}_0(t) \) is given now by:

\[
\hat{\theta}_0(t) = d_1\hat{\theta}(t) + d_2\hat{\theta}(t - 1) + \ldots \\
+ F(t)[c_1\phi(t - 1)\epsilon(t) + c_2\phi(t - 2)\epsilon(t - 1) + \ldots] 
\] (9)

References
