Solution of an ARMA-Representation via its Boundary Mapping Equation.

J.Jones¹, N.P.Karampetakis¹ and A.C.Pugh¹

¹Department of Mathematical Sciences
Loughborough University of Technology
Loughborough, Leics
England, U.K.

¹Department of Mathematics
Aristotle University of Thessaloniki
54006 GREECE

ABSTRACT

Background

Consider the regular, discrete-time, AutoRegressive Moving Average (ARMA)-Representation described by the matrix equation

\[ A(\sigma)y(k) = B(\sigma)u(k) \]  

(1)

where \( \sigma \) denotes the backwards shift operator (i.e. \( \sigma^i y(k) = y(k + i) \)). \( A(\sigma) = A_0 + A_1 \sigma + \ldots + A_p \sigma^p \in \mathbb{R}[\sigma]^{r \times r} \), \( B(\sigma) = B_0 + B_1 \sigma + \ldots + B_q \sigma^q \in \mathbb{R}[\sigma]^{r \times m} \)

where at least one of \( A_q, B_q \) is non-zero, \( y(k) : \mathbb{Z}^+ \rightarrow \mathbb{R}^r \) defines the output and \( u(k) : \mathbb{Z}^+ \rightarrow \mathbb{R}^m \) defines the input of the system.

In the case where \( A(\sigma) = \sigma E - A \in \mathbb{R}[\sigma]^{r \times r} \) and \( B(\sigma) = B \in \mathbb{R}^{r \times m} \) the ARMA-Representation (1) reduces to the Generalized State Space (GSS)-Representation

\[ E y(k + 1) = A y(k) + B u(k) \]  

(2)

Further if \( E \) is non-singular (i.e. \( |E| \neq 0 \)) then (2) reduces to the State Space Representation.

Consider a solution to the system (1) (resp. (2)), over the index set \([0, N]\). Such a solution is represented in the form \([y(0), y(1), \ldots, y(N - 1), y(N)]\).

The solution of (2) has been considered by many authors [3]–[7]. In Luemberger[6] the conditions under which (2) possessed a solution were discussed for both the time-invariant and time-variant cases and the notions of solvability and conditionability were introduced. Following [6] we say,

**Definition 1 (Solvability)**

The system (1) (resp. (2)) is solvable in case for any input sequence \([u(0), u(1), \ldots, u(N)]\) there exists at least one sequence \([y(0), y(1), \ldots, y(N)]\) which solves the system (1) (resp. (2)).

In general if a system is solvable its solution will not be unique i.e. there will exist a set of solutions. To specify a unique solution additional constraints must be specified and applied to the system. It is natural to apply these additional constraints at the boundary points of the interval concerned. If these additional constraints are specified at intermediate points in the interval, and not at the boundary points, then they can be regarded as being independent of the conditions at either end. In this respect they can be considered as partly redundant. We therefore have the following definition[6].
Definition 2 (Conditionability)

The system (1) (resp. (2)) is conditionable in case a unique solution 
\( y(0), y(1), \ldots, y(N) \) exists for any input sequence \( u(0), u(1), \ldots, u(N) \) by the specification of additional restrictions at the boundary points in the interval. 

It is noted in [6] that for the regular case of (2) (i.e. \( \text{rank} E - A \neq 0 \)) the system (2) is always solvable and conditionable. This can be shown to be true for the ARMA-Representation (1) if \( A(\sigma) \) is regular and this will be shown in the paper.

To obtain a solution \( y(0), y(1), \ldots, y(N) \) to (2) Luenberger [6] noted that the system can be completely characterized by all the pairs \( (y(0), y(N)) \) which satisfy (2). These can be regarded as the admissible boundary conditions of (2) and are represented in the form

\[ Z_0 y(0) + Z_N y(N) = C \]  

for some matrices \( Z_0, Z_N, C \). (3) is referred to as the boundary mapping equation for (2).

In cases where (3) does not give a unique boundary pair \( (y(0), y(N)) \), and hence not a unique solution, additional constraints can be applied to the system in the form of an auxiliary equation

\[ Z_0 \dot{y}(0) + Z_N \dot{y}(N) = \dot{C} \]  

Combining (3) and (4) results in

\[ \begin{pmatrix} Z_0 & Z_N \\ \dot{Z}_0 & \dot{Z}_N \end{pmatrix} \begin{pmatrix} y(0) \\ \dot{y}(N) \end{pmatrix} = \begin{pmatrix} C \\ \dot{C} \end{pmatrix} \]  

Hence, a unique boundary pair \( (y(0), y(N)) \), and hence a unique solution, will be produced iff \( \mathcal{Z} \mathcal{Z}^\dagger C = C \) where \( \mathcal{Z}^\dagger \) denotes the generalized inverse of \( \mathcal{Z} \).

Luenberger [7] gave a recursive procedure to solve (3) from which a solution to (2) was directly found. Another approach to form the boundary mapping equation, and hence obtain a solution, was presented by Karamanlioglu [3]. This method can be seen to be superior to that of [7] in that it easily extends to the case of 2-D systems.

Contribution

In this paper we extend the method of [3] to the case of regular ARMA-Representations as given in (1). For this case we compute a boundary mapping equation of the form

\[ Z_0 Y_{in} + Z_N Y_{fiva} = C \]  

where \( Y_{in}^T = (y(0)^T, \ldots, y(q-1)^T)^T \in \mathbb{R}^{vq}, Y_{fiva}^T = (y(N-q+1)^T, \ldots, y(N)^T)^T \in \mathbb{R}^{vq} \).

This is done as follows. Consider the system (1) where we assume the input is known throughout the range \( [0, N] \). Taking \( \mathcal{Z} \)-transforms of (1), assuming \( \mathcal{Y}(z) = \mathcal{Z}[y(k)] = \sum_{i=0}^{N} y(i)z^{-i}, \dot{u}(z) = \mathcal{Z}[\dot{u}(k)] = \sum_{i=0}^{N} u(i)z^{-i} \), it can be shown that the following relation
holds

\[
A(z)\hat{y}(z) = [z^q I_r \ldots z I_r] \begin{pmatrix}
A_q & \cdots & 0 \\
\vdots & \ddots & \vdots \\
A_1 & \cdots & A_q
\end{pmatrix} \begin{pmatrix}
y(0) \\
\vdots \\
y(q-1)
\end{pmatrix}
\begin{pmatrix}
\hat{A} \\
\hat{Y}
\end{pmatrix}
\]

\[
+ [z^{-N} I_r \ldots z^{-N+q-1} I_r] \begin{pmatrix}
A_0 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
A_{q-1} & \cdots & A_0
\end{pmatrix} \begin{pmatrix}
y(N) \\
\vdots \\
y(N-q+1)
\end{pmatrix}
\begin{pmatrix}
\hat{A} \\
\hat{Y}
\end{pmatrix}
\]

\[
+ [I_r \ldots z^{-N+q} I_r] \begin{pmatrix}
B_0 & \cdots & B_q & 0 \\
\vdots & \ddots & \vdots & \vdots \\
0 & B_0 & \cdots & B_q
\end{pmatrix} \begin{pmatrix}
u(0) \\
\vdots \\
u(N)
\end{pmatrix} \overset{\text{def}}{=} Q(z)
\begin{pmatrix}
\hat{U}
\end{pmatrix}
\]

(7)

where \( I_r \) denotes the \((r \times r)\) identity matrix.

From [3]

\[
A(z)^{-1} = \frac{H(z)}{d(z)}
\]

(8)

where

\[
H(z) = \sum_{i=0}^{n-1} H_i z^i \in \mathbb{R}[z]^{r \times r}, \quad d(z) = \sum_{i=0}^{n} d_i z^i \in \mathbb{R}
\]

(9)

where at least one of \( H_{n-1}, d_n \) is non-zero. This inversion can be easily implemented through the numerical algorithm proposed by [2] and consequently the coefficient matrices \( H_i \) and scalars \( d_i \) may be readily obtained. Combining (7) and (8)

\[
\hat{y}(z) = \frac{H(z)}{d(z)} Q(z) \Rightarrow [d(z) I_r] \hat{y}(z) = H(z) Q(z)
\]

(10)

Expanding both sides of (10) and equating powers of \( z \) we can obtain the matrix equation

\[
[D \otimes I_r] Y = (\hat{R} \hat{A} \hat{Y} + \hat{R} \hat{A} \hat{Y} + R_w B_w \hat{U})
\]

(11)

where \( \otimes \) denotes the Kronecker product, the matrices \( \hat{A}, \hat{A}, B_w \) and vectors \( \hat{Y}, \hat{Y}, \hat{U} \) are as defined in (7) and

\[
\hat{R} = \begin{pmatrix}
O_{(N+1)r, qr} \\
H_{n-1} \\
\vdots \\
H_0
\end{pmatrix}, \quad R_w = \begin{pmatrix}
O_{q r, (N-q+1)r} \\
H_{n-1} \\
\vdots \\
H_0
\end{pmatrix}
\]

3
\[ D = \begin{pmatrix} O_{q-1,N+1} \\ d_0 & \cdots & d_n \\ \vdots & & \vdots \\ d_s & \cdots & d_n \\ \vdots & & \vdots \\ d_n & \cdots & d_0 \end{pmatrix} \]  
\[ \hat{R} = \begin{pmatrix} H_{n-1} \\ \vdots \\ H_0 \\ \vdots \\ H_{n-1} \\ O_{(N+1)p,qr} \end{pmatrix} \]

where \( d_s \) is defined as the first non-zero element in the first column of the matrix \( D \) as represented in (12). Consider a non-singular matrix \( P \in \mathbb{R}^{(N+u+q) \times (N+u+q)} \) such that

\[ PD = \begin{pmatrix} 0_{(q+n+q-1),(N+1)} \\ I_{(N+1)} \\ 0_{b,(N+1)} \end{pmatrix} \]  

Such a matrix \( P \) can be selected to have the form

\[ P = \begin{pmatrix} I_{\beta,\beta} & O_{\beta,\alpha} \\ \alpha_{\alpha,\beta} & \vdots \\ \vdots & \vdots \\ \alpha_{N+\beta,\alpha} & \alpha_1 & \alpha_0 \end{pmatrix} \]

where \( \alpha = (N + s + 1) \), \( \beta = (n + q - s + 1) \) and

\[ a_k = \begin{cases} \frac{1}{d_s} & \text{, } k = 0 \\ \frac{1}{d_s} \sum_{i=1}^{k} \left[ -a_{(k-1)d_i} \right] & , 1 \leq k \leq s \\ \frac{1}{d_s} \sum_{i=1}^{k} \left[ -a_{(k-1)d_i} \right] & , s + 1 \leq k \leq N + s \end{cases} \]

By multiplying both sides of (11) by the matrix \((P \otimes \mathcal{I}_r)\) results in

\[ \begin{pmatrix} 0_{(q+n+q-1)p,1} \\ y(0) \\ \vdots \\ y(N) \\ 0_{s+1,r,1} \end{pmatrix} = (P \otimes \mathcal{I}_r) \left( \hat{R} \hat{A} \hat{Y} + \tilde{R} \hat{A} \tilde{Y} + R_sB_xL \right) \in \mathbb{R}^{(N+u+q)} \]  

(16)
i.e. the l.h.s of (16) is explicitly in terms of the solution vectors \( y(i), i = 0, \ldots, N \). The selection of the matrix \( P \) in (14) will be further discussed in the paper.

From (16) the boundary mapping equation can be immediately realized and an admissible set of boundary conditions found as follows. It is clear that the first \((2q + n - s - 1)r\) and last \((s + q)r\) equations only involve the boundary conditions \( \underline{y}_{in} \) and \( \underline{y}_{fin} \). Hence we can extract these \((3q + n - 1)r\) equations to form the boundary mapping equation as given in (6) where \( Z_0, Z_N \in \mathbb{R}^{(3q+n-1)r\times q} \) and \( C \in \mathbb{R}^{(N+n+q)} \).

To define a unique solution an auxiliary equation, of the form (4), can be appended to the boundary mapping equation formed. This is always the case for a regular ARMA-Representation as it is always conditionable (see paper).

Once an admissible set of boundary conditions have been computed, via (6), the r.h.s of (14) is known. Hence the intermediate states \( y(q), \ldots, y(N - q) \) can be immediately determined in any order.

Subsequently algorithms for the construction of the boundary mapping equation and solving of the system are presented in the paper. These algorithms are especially attractive for implementation via a language such as MAPLE [1] due to their high dependence on matrix manipulation techniques. MAPLE is a symbolic computational package which contains such inbuilt matrix procedures. These combine to form the linear algebra package linalg. Subsequently, in the paper, we present such computer procedures, based on the aforementioned algorithms, for the computation of the boundary mapping equation and the corresponding solution of the system.

References


