Maximum Likelihood Estimation of Stable ARX Models using Randomized Coordinate Descent

Ozmen Erkin Kokten and Raviv Raich
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331-5501
{kokteno,raich}@oregonstate.edu

***Abstract*—Autoregressive models play an important role in a variety of applications including finance, engineering, sciences, and agriculture. While for some models (e.g., physics-based models) parameters are known, in other domains the parameters may not be available. This paper deals with the estimation of the parameters of an autoregressive model with exogenous variables. A significant body of literature has explored autoregressive model estimation across different estimation criteria, data availability, and parameterization; however, limited attention has been given to the estimation problem under stability constraints. The incorporation of stability constraints often results in increased computational complexity. As an efficient alternative, we propose to estimate stable ARX parameters using randomized coordinate descent. To demonstrate the efficiency of the proposed approach, we present an empirical convergence study and compare our approach to a state-of-the-art alternative.**

***Index Terms*—autoregressive with exogenous variables, stable, coordinate descent, parameter estimation**

I. Introduction

Autoregressive moving average (ARMA) and autoregressive with exogenous variables (ARX) models are pivotal in various fields including finance, engineering, sciences, and agriculture [1],[2]. One of the key uses of such models is forecasting. To achieve accurate forecasts, the model parameters must be known or estimated from data. Maximum likelihood estimation under Gaussian noise assumption results in least squares (LS) formulation, for which a simple and well-known solution exists. However, in the past few decades, many alternative formulations to the estimation problem have been proposed including changes to the objective, constraints, or optimization type with each alternative resulting in a more challenging problem. One such alternative, of interest to this paper, is parameter estimation under stability constraints. Stability-constrained ARX/ARMA models can improve long-term forecasts but introduce a layer of complexity to the problem, namely, non-trivial characterization of the constraints.

Our paper focuses on the problem of parameter estimation of ARX models under stability constraints. Since stability for ARMA and ARX models is enforced through the common autoregressive (AR) component, some of the underlying approaches for parameter estimation cross over. Hence, we proceed with a review of stability-constrained parameter estimation for both ARMA and ARX models. Due to the non-trivial nature of stability constraints, most methods for estimating stable ARMA/ARX employ a reparameterization approach based on the Levinson mapping introduced in the Levinson [3] - Durbin [4] algorithm. For ARMA parameter estimation, Jones [5] combines Levinson-Durbin recursion with an additional mapping to ensure stability by restricting the parameters to $(-1, 1)$. Beadle and Djuric combine the same recursion and Jacobians to efficiently generate stable parameters [6]. Similarly, Combettes and Trussel use the Levinson mapping and hypercube stability set to estimate stable parameters [7]. Di Gangi et al. use Levinson transformation of partial autocorrelations and partial moving average coefficients in optimization to obtain a stable system [8]. Furthermore, White et al. stabilize the system by constructing the AR matrix as a slowly decaying permutation matrix using the randomly generated eigenvectors, eigenvalues, and their conjugates [9]. Nallasivam et al. enforce stability by incorporating modified Jury stability in bilinear optimization [10]. [11] links the AR polynomial's stability conditions to the eigenvalues of the state matrix, while [12] applies this by ensuring stability via joint spectral radius constraints. In cases of missing data, Horner et al. and Isaksson used Kalman filter-based reconstruction alongside expectation maximization, maximum likelihood estimation, and LS techniques [13],[14]. Although they do not directly estimate stable parameters, [15] stabilizes given unstable polynomials, while [16] uses regularized LS to find a stable system matrix in state-space problems.

In this paper, we present a novel approach for parameter estimation of stability-constrained ARX models. In particular, we derive a version of randomized coordinate descent (CD) that relies on alternative parametrization based on pairing the roots of the AR polynomial, allowing stability constraints to be efficiently incorporated. Parameter updates are provided in closed form. Additionally, we provide a computational complexity analysis to assess the per-iteration complexity and the rate of convergence of the algorithm via experiments.

The paper is organized as follows. Section II reviews ARX models and their stability. Section III formulates the parameter estimation problem for the ARX model. Section IV presents our approach for stability-constrained/unconstrained ARX model parameter estimation using CD. Section V includes the algorithm implementation and comparison. Finally, a summary of the work is provided in Section VI.

II. Background

In this section, we review the ARX model, an alternative representation of the model, and the definition of stability of the model.

A. The ARX model

This paper focuses on ARX models. Given a collection of known input sequence $x_j[n]$ for $j = 1, 2, \dots, q$, the output of the model is given by

$$y[n] = \sum_{i=1}^{p} a_i y[n-i] + \sum_{j=1}^{q} b_j x_j[n] + e[n] \tag{1}$$

where a_i for $i = 1, \dots, p$ are the real-valued autoregressive coefficients, b_j for $j = 1, \dots, q$ are real-valued coefficients, and $e[n]$ are sampled $i.i.d.$ from $\mathcal{N}(0, \sigma^2)$. This model is commonly used for characterizing a system that maps multiple input sequences to a scalar output sequence.

B. Alternative parametrization for ARX

Since the ARX model is a linear time-invariant (LTI) model, it can be characterized in the Z-domain. If we let $u[n] = \sum_{j=0}^{q} b_j x_j[n]$, then following Z-domain equation can be used to describe the model:

$$A(z)Y(z) = U(z) + E(z)$$

This research was supported by USDA NIFA award No. 2021-67021-35344 (AgAID AI Institute).

where $Y(z)$, $U(z)$, and $E(z)$ denote the Z-transform of $y[n]$, $u[n]$, and $e[n]$, respectively, and $A(z)$ is the AR coefficient polynomial given via its additive form as

$$A(z) = 1 - a_1 z^{-1} - a_2 z^{-2} - \ldots - a_p z^{-p}.$$

The transfer function of the model for both $U(z)$ and $E(z)$ is given by $\frac{1}{A(z)}$. In a variety of system analysis steps (e.g., stability tests or partial fraction expansion), it is common to consider the multiplicative form for representing the polynomial $A(z)$:

$$A(z) = \prod_{k=1}^{p} (1 - \alpha_k z^{-1}), \tag{2}$$

where α_k's are the roots of the polynomial. By comparison of the additive form to the multiplicative form, a_k for $k = 1, 2, \ldots, p$ can be written as a function of $\boldsymbol{\alpha} \in \mathbb{R}^p$ as follows

$$a_k(\boldsymbol{\alpha}) = (-1)^{k-1} e_k(\boldsymbol{\alpha}) \tag{3}$$
$$e_k(\boldsymbol{\alpha}) = \sum_{1 \le i_1 < \ldots < i_k \le p} \alpha_{i_1} \alpha_{i_2} \cdots \alpha_{i_k}, \tag{4}$$

where $e_k(\boldsymbol{\alpha})$ is an elementary symmetric polynomial [17, p. 547-552]. Brute-force calculation of the symmetric polynomials is computationally expensive, however, Newton identities can be used to efficiently calculate these polynomials (see Section IV-B3) [18, p. 278-279]. The vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_p]^T$ may be complex-valued to ensure the existence of the multiplicative representation. However, some restriction on $\boldsymbol{\alpha}$ must be applied to ensure that $\boldsymbol{a}$ is a real-valued vector, i.e., $\boldsymbol{\alpha} \in S_p$, where

$$S_p = \{\boldsymbol{\alpha} \in \mathbb{C}^p | \ \operatorname{Im}\{\textstyle\sum \alpha_{i_1} \ldots \alpha_{i_k}\} = 0, \text{ for } 1 \le k \le p\}.$$

The parameterization of the multiplicative form is not unique: any permutation of $\boldsymbol{\alpha}$ results in the same $A(z)$. Since any permutation of valid $\boldsymbol{\alpha}$ is also valid, S_p is redundant and some of the ambiguity can be removed as follows. The α_is can be organized further in pairs and the feasibility set for $\boldsymbol{\alpha}$ can be reduced to S_p^e and S_p^o for the even and odd p, respectively,

$$S_p^e = \big\{\boldsymbol{\alpha} \in \mathbb{C}^p | \ (\alpha_{2k-1}, \alpha_{2k}) \in S_2 \text{ for } 1 \le k \le \frac{p}{2}\big\}$$
$$S_p^o = \big\{\boldsymbol{\alpha} \in \mathbb{C}^p | \ (\alpha_{2k-1}, \alpha_{2k}) \in S_2 \text{ for } 1 \le k \le \frac{p-1}{2}, \ \alpha_p \in \mathbb{R}\big\}.$$

This is because α_i's can be paired so that each pair is either a complex conjugate pair or a real-valued pair.

C. Stable ARX Processes

To define the stability of the ARX model in (1), the transfer function $\frac{1}{A(z)}$ is considered. The condition for BIBO stability (known as Jury stability [19]) requires that all roots of the denominator $A(z)$ lie inside the unit circle [20, p. 182-185]. This can be verified by the multiplicative form of $A(z)$ in (2). With roots $(\alpha_1, \ldots, \alpha_p)$, stability is ensured by $|\alpha_i| < 1$ for all i. The set of all stable ARX models defined via their $\boldsymbol{\alpha}$ parameters is given by

$$G_p = S_p \cap \Big\{\boldsymbol{\alpha} \in \mathbb{C}^p \ \Big| \ |\alpha_i| < 1 \text{ for } 1 \le i \le p\Big\},$$

where S_p^e or S_p^o can be used in place of S_p to characterize the same set of stable models. In the literature, it is acknowledged that applying constraints to $\boldsymbol{a}$ directly is a complicated problem. As a solution, parameter transformation techniques that enforce stability constraints within the process like Jones reparameterization, Levinson transformation, and Barndorff et al.'s parameterization techniques are available [5], [3], [21].

III. Problem Formulation

Here, we formulate the ARX parameter estimation with/without stability constraints as a constrained/unconstrained LS problem.

A. ARX parameter estimation via MLE

To estimate the ARX model parameters, the maximization of the conditional log-likelihood is considered. The negative log-likelihood objective, up to a constant, is given by

$$\frac{1}{2\sigma^2} \sum_{i=p+1}^{n} \Big(y[i] - \sum_{k=1}^{p} y[i-k] a_k - \sum_{k=1}^{q} x_k[i] b_k\Big)^2. \tag{5}$$

where $(y[i], x_1[i], x_2[i] \ldots, x_q[i])$ for $i = 1, 2, \ldots, n$ are the available observations, and $(a_1, a_2, \ldots, a_p, b_1, b_2, \ldots, b_q)$ are the unknown parameters. The minimization of the negative log-likelihood objective is the following LS problem:

$$\min_{\boldsymbol{a}, \boldsymbol{b}} \|\boldsymbol{y} - (\boldsymbol{Y}\boldsymbol{a} + \boldsymbol{X}\boldsymbol{b})\|^2, \tag{6}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q]$, $\boldsymbol{Y} = [\boldsymbol{y}_{n-1}, \boldsymbol{y}_{n-2}, \ldots \boldsymbol{y}_{n-p}]$, $\boldsymbol{y} = [y[n], y[n-1], \ldots, y[p+1]]^T$, $\boldsymbol{y}_i = [y[i], y[i-1], \ldots y[p+i-(n-1)]]^T$, $\boldsymbol{x}_i = [x_i[n], x_i[n-1], \ldots x_i[p+1]]^T$, $\boldsymbol{a} = [a_1, \ldots, a_p]^T$ and $\boldsymbol{b} = [b_1, \ldots, b_q]^T$. We refer to this LS optimization as the unconstrained MLE for ARX.

B. ARX parameter estimation LS forms

Unconstrained, the minimization in (6) can be trivially solved with the LS solution as

$$[\boldsymbol{a}_*^T, \boldsymbol{b}_*^T]^T = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{y}$$

where $\boldsymbol{H} = [\boldsymbol{Y}, \boldsymbol{X}]$. To include stability constraints on $\boldsymbol{a}$, we consider a two-step implementation of the optimization as follows. Since $\boldsymbol{b}$ is unconstrained, we start by finding an optimal $\boldsymbol{b}$ as a function of $\boldsymbol{a}$: $\boldsymbol{b}_*(\boldsymbol{a}) = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{Y}\boldsymbol{a})$. Then, substituting this value of $\boldsymbol{b}$ back into the objective in (6), the optimization reduces to $\min_{\boldsymbol{a}} \|\boldsymbol{P}_X^{\perp}(\boldsymbol{y} - \boldsymbol{Y}\boldsymbol{a})\|^2$ where $\boldsymbol{P}_X^{\perp} = \boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$. This can be written as the following quadratic optimization:

$$\min_{\boldsymbol{a}} \boldsymbol{a}^T\boldsymbol{M}\boldsymbol{a} - 2\boldsymbol{c}^T\boldsymbol{a} + const \tag{7}$$

where $\boldsymbol{M} = \boldsymbol{Y}^T\boldsymbol{P}_X^{\perp}\boldsymbol{Y}$ and $\boldsymbol{c} = \boldsymbol{Y}^T\boldsymbol{P}_X^{\perp}\boldsymbol{y}$. Note that this optimization focuses directly on the estimation of the AR parameters $\boldsymbol{a}$. Without additional constraints, the problem is a quadratic minimization and its solution can be obtained in closed-form (by differentiating and setting to zero) as $\boldsymbol{a}_* = \boldsymbol{M}^{-1}\boldsymbol{c}$. Under stability constraints on $\boldsymbol{a}$, the problem becomes a challenging constrained quadratic minimization as stated in II-C. Hence, we proceed by reparametrizing the problem in terms of the poles $\boldsymbol{\alpha}$ and performing the optimize directly on $\boldsymbol{\alpha}$.

IV. CD for ARX parameter estimation

In this section, we present a CD approach for MLE of the ARX model parameters. For an easy exposition, we begin with the solution for the problem without stability constraints and proceed with the solution with stability constraints.

A. CD without stability constraints

We start by rewriting the optimization over $\boldsymbol{\alpha}$, without stability constraints

$$\min_{\boldsymbol{\alpha} \in S_p} \boldsymbol{a}(\boldsymbol{\alpha})^T\boldsymbol{M}\boldsymbol{a}(\boldsymbol{\alpha}) - 2\boldsymbol{c}^T\boldsymbol{a}(\boldsymbol{\alpha}) \tag{8}$$

where $a_k(\boldsymbol{\alpha})$ for $k = 1, 2, \ldots, p$ is given in (3). This problem is no longer quadratic or convex in its objective and its feasibility set is nontrivial. However, when optimized with respect to one entry or a pair of entries, the problem becomes tractable.

1) Single real-valued parameter update: WLOG, we assume that a real-valued α_1 is selected to be updated at iteration t. We set α_i for $i = 2, \ldots, p$ fixed to their current iteration value $\alpha_i = \alpha_i^t$ and keep α_1 as a variable. The dependence of $\boldsymbol{a}$ on α_1 is given by

$$\boldsymbol{a}(\alpha_1, \alpha_2^t, \ldots, \alpha_p^t) = \boldsymbol{u}_0 + \boldsymbol{u}_1 \alpha_1 \tag{9}$$

where $\boldsymbol{u}_0 = \boldsymbol{a}(\boldsymbol{\alpha})|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^t}$, $\boldsymbol{u}_1 = \frac{\partial \boldsymbol{a}(\boldsymbol{\alpha})}{\partial \alpha_1}|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^t}$, and $\boldsymbol{\alpha}^t = [0, \alpha_2^t, \ldots, \alpha_p^t]^T$. Note that both $\boldsymbol{u}_0$ and $\boldsymbol{u}_1$ consist of elementary symmetric polynomials in $\alpha_2^t, \ldots, \alpha_p^t$. Using (9), the optimization problem for α_1 can be written as

$$\min_{\alpha_1 \in \mathbb{R}} (\boldsymbol{u}_0 + \alpha_1 \boldsymbol{u}_1)^T \boldsymbol{M} (\boldsymbol{u}_0 + \alpha_1 \boldsymbol{u}_1) - 2\boldsymbol{c}^T (\boldsymbol{u}_0 + \alpha_1 \boldsymbol{u}_1)$$

where no further constraints are needed to ensure that $[\alpha_1, \alpha_2^t, \ldots, \alpha_p^t]^T \in S_p$ since the real-valued α_1^t is replaced with another real-valued number. Since the problem is a convex quadratic minimization, the optimal solution can be obtained by differentiating with respect to α_1 and setting it to zero. The resulting optimal α_1 is given in closed-form as

$$\alpha_1^{t+1} = \frac{\boldsymbol{u}_1^T (\boldsymbol{c} - \boldsymbol{M}\boldsymbol{u}_0)}{\boldsymbol{u}_1^T \boldsymbol{M} \boldsymbol{u}_1}.$$

This illustrates the advantage of the re-parameterization using $\boldsymbol{\alpha}$. When using a CD approach, the update of a single parameter at a time yields a simple unconstrained real-valued quadratic optimization problem and as shown in Section IV-B1, the enforcement of constraints can be directly implemented.

2) Pair update: Here, we consider the update for any pair $(\alpha_{2k-1}, \alpha_{2k}) \in S_2$. WLOG, we consider the CD update of the pair (α_1, α_2) when $\alpha_3, \ldots, \alpha_p$ are held fixed. Hence, at the tth iteration

$$\boldsymbol{a}(\alpha_1, \alpha_2, \alpha_3^t, \ldots, \alpha_p^t) = \boldsymbol{v}_0 + \boldsymbol{v}_1(\alpha_1 + \alpha_2) + \boldsymbol{v}_2 \alpha_1 \alpha_2 \tag{10}$$

where $\boldsymbol{v}_0 = \boldsymbol{a}(\boldsymbol{\alpha})|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^t}$, $\boldsymbol{v}_1 = \frac{\partial \boldsymbol{a}(\boldsymbol{\alpha})}{\partial \alpha_1}|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^t}$, and $\boldsymbol{v}_2 = \frac{\partial^2 \boldsymbol{a}(\boldsymbol{\alpha})}{\partial \alpha_1 \partial \alpha_2}|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^t}$ where with a slight abuse of notation, we redefine $\boldsymbol{\alpha}^t$ as $\boldsymbol{\alpha}^t = [0, 0, \alpha_3^t, \ldots, \alpha_p^t]^T$. Note that $\boldsymbol{v}_0$, $\boldsymbol{v}_1$, and $\boldsymbol{v}_2$ consist of elementary symmetric polynomials in $\alpha_3^t, \ldots, \alpha_p^t$. The optimization problem for α_1 and α_2 can be written by substituting (10) into (8) and minimizing (8) with respect to $(\alpha_1, \alpha_2) \in S_2$. In this case, the objective as a function of (α_1, α_2) is not quadratic and hence the minimization does not appear trivial. To remedy this issue, a new parametrization is considered: $s = \frac{1}{2}(\alpha_1 + \alpha_2)$ and $d = \alpha_1 \alpha_2$. Using this parametrization and requiring $s, d \in \mathbb{R}$ ensures that $(\alpha_1, \alpha_2) \in S_2$. Using this reparametrization, we have

$$\min_{(s,d) \in \mathbb{R}^2} f(s, d) \qquad \text{where}$$

$$\begin{aligned} f(s, d) = &(\boldsymbol{v}_0 + 2s\boldsymbol{v}_1 + d\boldsymbol{v}_2)^T \boldsymbol{M} (\boldsymbol{v}_0 + 2s\boldsymbol{v}_1 + d\boldsymbol{v}_2) \\ &-2\boldsymbol{c}^T (\boldsymbol{v}_0 + 2s\boldsymbol{v}_1 + d\boldsymbol{v}_2) \end{aligned} \tag{11}$$

This formulation involves the optimization of a convex quadratic over an unconstrained space $\mathbb{R}^2$. The optimal solution is given by

$$[s, d]^T = ([2\boldsymbol{v}_1, \boldsymbol{v}_2]^T \boldsymbol{M} [2\boldsymbol{v}_1, \boldsymbol{v}_2])^{-1} [2\boldsymbol{v}_1, \boldsymbol{v}_2]^T (\boldsymbol{c} - \boldsymbol{M}\boldsymbol{v}_0) \tag{12}$$

and $\alpha_1^{t+1}, \alpha_2^{t+1}$ are given by

$$\alpha_{1,2}^{t+1} = s \pm \sqrt{s^2 - d}. \tag{13}$$

B. CD with stability constraints

To enforce stability, the constraint $\boldsymbol{\alpha} \in S_p$ in (8) is replaced with $\boldsymbol{\alpha} \in \bar{G}_p$, where $\bar{G}_p$ denotes the closure of the set of stable solutions G_p. Replacing G_p with its closure ensures the existence of a solution to the constrained minimization. Effectively, our new constraint set $\bar{G}_p$ represents the set of stable and critically stable systems.

1) Single real-valued parameter update: Substituting $\boldsymbol{a}(\boldsymbol{\alpha})$ in (9) into (8) and applying the stability constraint, the optimization problem for α_1 can be written as

$$\min_{\alpha_1 \in \bar{G}_1} (\boldsymbol{u}_0 + \alpha_1 \boldsymbol{u}_1)^T \boldsymbol{M} (\boldsymbol{u}_0 + \alpha_1 \boldsymbol{u}_1) - 2\boldsymbol{c}^T (\boldsymbol{u}_0 + \alpha_1 \boldsymbol{u}_1).$$

Since $\bar{G}_1 = [-1, 1]$, the optimal α_1 is given in closed-form as

$$\alpha_1^{t+1} = P_{[-1,1]}\left(\frac{\boldsymbol{u}_1^T (\boldsymbol{c} - \boldsymbol{M}\boldsymbol{u}_0)}{\boldsymbol{u}_1^T \boldsymbol{M} \boldsymbol{u}_1} \right),$$

where the projection $P_{[a,b]}$ is

$$P_{[a,b]}(x) = aI(x < a) + xI(a \leq x \leq b) + bI(x > b)$$

for scalars and applied element-wise for vectors.

2) Pair update: To update a pair, we follow the approach of Section IV-A2 to obtain the following optimization

$$\min_{(s,d) \in \mathbb{T}} f(s, d)$$

where $\mathbb{T}$ is the triangle with the vertices $(1, 1)$, $(-1, 1)$, and $(0, -1)$:

$$\mathbb{T} = \{(s, d) \mid 1 \geq d \geq -1 + 2|s|\}.$$

Mapping $(s, d) \in \mathbb{T}$ back to (α_1, α_2) ensures that $(\alpha_1, \alpha_2) \in \bar{G}_2$. Similar triangle-like stability constraints have been introduced in [6] and [7]. The solution in this case starts with the following steps:

1) Find (s, d) using (12). If $(s, d) \in \mathbb{T}$, then the solution satisfies the constraints and hence is the optimum. Exit.
2) Else, the solution must be on the boundary of $\mathbb{T}$: either (1) $d = 1$ $|s| \leq 1$ or (2) $d = -1 + 2s$, $0 \leq s \leq 1$ or (3) $d = -1 - 2s$, $-1 \leq s \leq 0$. The solution for each case is given by

   Case (1):

$$s_1 = P_{[-1,1]}\left(\frac{\boldsymbol{v}_1^T (\boldsymbol{c} - \boldsymbol{M}(\boldsymbol{v}_0 + \boldsymbol{v}_2))}{2\boldsymbol{v}_1^T \boldsymbol{M} \boldsymbol{v}_1} \right), \; d_1 = 1 \tag{14}$$

   Case (2):

$$s_2 = P_{[0,1]}\left(\frac{(\boldsymbol{v}_1 + \boldsymbol{v}_2)^T (\boldsymbol{c} - \boldsymbol{M}(\boldsymbol{v}_0 - \boldsymbol{v}_2))}{2(\boldsymbol{v}_1 + \boldsymbol{v}_2)^T \boldsymbol{M} (\boldsymbol{v}_1 + \boldsymbol{v}_2)} \right), \; d_2 = -1 + 2s_2 \tag{15}$$

   Case (3):

$$s_3 = P_{[-1,0]}\left(\frac{(\boldsymbol{v}_1 - \boldsymbol{v}_2)^T (\boldsymbol{c} - \boldsymbol{M}(\boldsymbol{v}_0 - \boldsymbol{v}_2))}{2(\boldsymbol{v}_1 - \boldsymbol{v}_2)^T \boldsymbol{M} (\boldsymbol{v}_1 - \boldsymbol{v}_2)} \right), \; d_3 = -1 - 2s_3 \tag{16}$$

Set (s, d) to (s_l, d_l) for $l = 1, 2, 3$ that minimizes the objective in (11). After determining (s, d), obtain $(\alpha_1^{t+1}, \alpha_2^{t+1})$ using (13).

3) Efficient computation of elementary symmetric polynomials: Here, the goal is to compute all the symmetric elementary polynomials $e_k(\alpha_1, \ldots, \alpha_p)$ in (4). This process is needed for the computation of $\boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{v}_0, \boldsymbol{v}_1$, and $\boldsymbol{v}_2$. These polynomials are computed using the well-known Newton-Girard Formulas [18]. First, the power sums of $\boldsymbol{\alpha}$'s are computed for $k = 1, \ldots, p$ as follows

$$g_k(\boldsymbol{\alpha}) = g_k(\alpha_1, \ldots, \alpha_p) = \sum_{i=1}^{p} \alpha_i^k.$$

Note that this sum is $\mathcal{O}(p)$ per polynomial and a total of $\mathcal{O}(p^2)$. Then, initializing $e_0(\boldsymbol{\alpha}) = 1$, the elementary polynomials $e_k(\boldsymbol{\alpha})$ for $k = 1, \ldots, p$ are computed recursively using

$$e_k(\boldsymbol{\alpha}) = -\frac{1}{k} \sum_{j=1}^{k} e_{k-j}(\boldsymbol{\alpha})(-1)^j g_j(\boldsymbol{\alpha}) \tag{17}$$

where computing all e_k is $\mathcal{O}(p^2)$.

4) Computational Complexity: Algorithm 1 summarizes the CD approach for ARX parameter estimation for the stability-un/constrained cases. The computational complexity outside the iterative process of the algorithm is $\mathcal{O}(q^3 + n(p^2 + q^2))$ due to $\boldsymbol{M}$ and $\boldsymbol{c}$ computation. The per-iteration computation complexity provided in lines 8, 9, 13, 14, and 16, which either follow the matrix multiplication of the form $\boldsymbol{M}\boldsymbol{r}_i$ where $\boldsymbol{M} \in \mathbb{R}^{p\times p}$ and $\boldsymbol{r}_i \in \mathbb{R}^{p\times 1}$, or the computation of elementary symmetric polynomials, is $\mathcal{O}(p^2)$.

Algorithm 1 Stable/Unstable ARX Parameter Estimation (Multiple α Update)

1: p, q, n_{iter}, $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{y}$, and **stable** are given.
2: Initialize $\boldsymbol{\alpha} = [\alpha_1, \alpha_2 \dots, \alpha_p]^T$ vector from $N(0, 1)$
3: $\boldsymbol{M} = \boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$ ▷ $\mathcal{O}(q^3 + n(p^2 + q^2))$
4: $\boldsymbol{c} = \boldsymbol{Y}^T\boldsymbol{y} - \boldsymbol{Y}^T\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$ ▷ $\mathcal{O}(pq + n(p + q))$
5: **for** $i = 1, 2 \dots n_{iter}$ **do**
6: Sample j uniformly from $\{1, 2, \dots, p\}$. ▷ $\mathcal{O}(1)$
7: **if** p is odd AND $j = p$ **then**
8: Calculate $\boldsymbol{u}_0$ and $\boldsymbol{u}_1$ in IV-A ▷ $\mathcal{O}(p^2)$
9: $\alpha_j = \frac{\boldsymbol{u}_1^T(\boldsymbol{c}-\boldsymbol{M}\boldsymbol{u}_0)}{\boldsymbol{u}_1^T\boldsymbol{M}\boldsymbol{u}_1}$ ▷ $\mathcal{O}(p^2)$
10: **if** **stable** = True **then**
11: $\alpha_j \leftarrow P_{[-1,1]}(\alpha_j)$ ▷ $\mathcal{O}(1)$
12: **else**
13: Calculate $\boldsymbol{v}_0$, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ in IV-A2 for $k = \lceil \frac{j}{2} \rceil$ ▷ $\mathcal{O}(p^2)$
14: $[s, d]^T = ([2\boldsymbol{v}_1, \boldsymbol{v}_2]^T\boldsymbol{M}[2\boldsymbol{v}_1, \boldsymbol{v}_2])^{-1}[2\boldsymbol{v}_1, \boldsymbol{v}_2]^T(\boldsymbol{c} - \boldsymbol{M}\boldsymbol{v}_0)$ ▷ $\mathcal{O}(p^2)$
15: **if** $(s, d) \notin \mathbb{T}$ AND **stable** = True **then**
16: Calculate $\{(s_i, d_i)\}_{i=1}^3$ using (14)-(16) ▷ $\mathcal{O}(p^2)$
17: $[s, d] = \arg\min_{(s,d)\in\{(s_i,d_i)\}_{i=1}^3}$ (Eq. 11) ▷ $\mathcal{O}(1)$
18: $\alpha_{2\lceil \frac{j}{2}\rceil-1} = s + \sqrt{s^2 - d}$, $\alpha_{2\lceil \frac{j}{2}\rceil} = s - \sqrt{s^2 - d}$ ▷ $\mathcal{O}(1)$
19: Compute $e_1(\boldsymbol{\alpha}), \dots, e_p(\boldsymbol{\alpha})$ using section IV-B3 ▷ $\mathcal{O}(p^2)$
20: Compute $\boldsymbol{a}(\boldsymbol{\alpha})$ using (3) ▷ $\mathcal{O}(p)$
21: **return** $\boldsymbol{a}(\boldsymbol{\alpha})$

V. Experiments

In this section, numerical experiments are used to: verify that the proposed CD can be used to minimize the unconstrained LS objective; verify that the proposed CD can be used to minimize the stability-constrained LS objective; assess via empirical observation the nature of convergence of the algorithm; and compare to the Levinson Mapping based projected gradient descent (LM-PGD) by Combettes and Trussell [7].

A. Data Generation and Parameter Initialization

We start by setting $p = 6$ and $q = 2$ and generate $\boldsymbol{a}$ values from $\boldsymbol{\alpha} = [0.8 + 0.7j, 0.8 - 0.7j, -0.8 + 0.7j, -0.8 - 0.7j, -1.1, 1.1]$ using equations (3) and (17). We randomly sample the entries of $\boldsymbol{b} = [b_0, b_1]^T$ and of $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2]$ where $\boldsymbol{x}_i \in \mathbb{R}^{n-p}$ for $n = 100$ *i.i.d.* from a standard normal distribution. We set the $y[1], \dots, y[p]$ to zero and generate the remaining entries $y[i]$ for $i = p+1, \dots, n$ using (1). The standard deviation of the noise, σ, was set to 0.01. From the sequence $y[i]$, vector $\boldsymbol{y}$ and matrix $\boldsymbol{Y}$ were constructed using (6).

B. Algorithms Evaluated

We evaluate the stability- unconstrained and constrained CD for the ARX model (ARXCD and ARXCD-S) from Algorithm 1 and LM-PGD. In our experiments, we ran ARXCD and compared its iterations to the solution of the unconstrained problem (i.e., the LS solution).

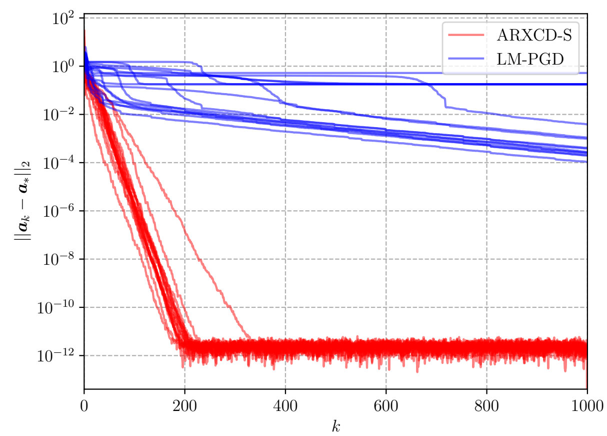

Fig. 1. Sum-of-squared-parameter-errors vs. epoch number k for ARXCD-S, and [7] using multiple runs with different initializations.

Separately, we compared ARXCD-S and LM-PGD that solve the stability-constrained LS in (8). For LM-PGD, we set the parameters as $\beta = 0.55$, $\epsilon = 10^{-14}$, $\delta = 0$, and $\mathbf{k}_0$ is uniformly sampled from $(-1, 1)^p$. Their original gradient-descent backtracking criterion was replaced with the backtracking for PGD: $\boldsymbol{\Phi}(\mathbf{k}_{n+1}) - \boldsymbol{\Phi}(\mathbf{k}_n) + 0.5\nabla\boldsymbol{\Phi}(\mathbf{k}_n)(\mathbf{k}_n - \mathbf{k}_{n+1}) > 0$ where $\mathbf{k}_{n+1} = \mathcal{P}(\mathbf{k}_n - \lambda_n\nabla\boldsymbol{\Phi}(\mathbf{k}_n))$. LM-PGD calculates the Jacobian of the Levinson transformation in every iteration, leading to a complexity of $\mathcal{O}(p^3)$ per iteration.

C. Results

Under stability constraints, multiple local solutions may exist. We jointly cluster all the solutions obtained from multiple runs of ARXCD-S and LM-PGD and present the error associated with trajectories that converge to one of the solutions. Figure 1 shows the L2 norm of the difference between $\boldsymbol{a}_k$, the AR coefficients after k epochs, and the exact solution $\boldsymbol{a}_*$ for multiple runs of ARXCD-S and LM-PGD. Each run of the same algorithm differs only by the initialization. We observe that most runs of ARXCD-S appear consistent sharing a similar linear convergence rate. We also observe that for some runs of LM-PGD a significant delay is present before reaching the linear convergence region. Note that this figure is based on one data generation of the problem. From multiple data generations, we noticed more variation in rates of convergence and time to reach linear convergence for LM-PGD than for ARXCD-S. While Fig. 1 shows a faster convergence rate for the proposed method, the opposite relation may be observed in other runs. Moreover, while not presented due to space limitation, we observe that ARXCD successfully converges to LS solution linearly and reaches an error for 10^{-12} at around 25 steps. Our overall assessment is that ARXCD-S offers a consistent and robust alternative to LM-PGD for solving the stability constraint MLE for the ARX model.

VI. Conclusion

In this study, we presented a novel approach for estimating stable ARX models using CD. We introduced a parameterization of the problem, allowing for an efficient closed-form update iteration. Numerical experiments conducted in this paper demonstrate a linear convergence rate that is competitive with the baseline, highlighting the efficiency of the proposed approach.

REFERENCES

[1] J. Janczura and A. Puć, "ARX-GARCH probabilistic price forecasts for diversification of trade in electricity markets—variance stabilizing transformation and financial risk-minimizing portfolio allocation," *Energies*, vol. 16, no. 2, pp. 1–28, 2023.

[2] M. Yang, Y. Qiu, J. Dong, L. Wu, and M. Shen, "Modeling of soil moisture data by arma time series," *Journal of Physics: Conference Series*, vol. 2650, no. 1, pp. 1–7, Nov 2023.

[3] N. Levinson, "The wiener (root mean square) error criterion in filter design and prediction," *Journal of Mathematics and Physics*, vol. 25, no. 1-4, pp. 261–278, 1946.

[4] J. Durbin, "The fitting of time-series models," *Revue de l'Institut International de Statistique / Review of the International Statistical Institute*, vol. 28, no. 3, pp. 233–244, 1960.

[5] R. H. Jones, "Maximum likelihood fitting of ARMA models to time series with missing observations," *Technometrics*, vol. 22, no. 3, pp. 389–395, 1980.

[6] E. Beadle and P. Djuric, "Uniform random parameter generation of stable minimum-phase real ARMA (p,q) processes," *IEEE Signal Processing Letters*, vol. 4, no. 9, pp. 259–261, 1997.

[7] P. Combettes and H. Trussell, "Best stable and invertible approximations for ARMA systems," *IEEE Transactions on Signal Processing*, vol. 40, no. 12, pp. 3066–3069, 1992.

[8] L. Di Gangi, M. Lapucci, F. Schoen, and A. Sortino, "Improved maximum likelihood estimation of arma models," *Lobachevskii Journal of Mathematics*, vol. 43, no. 9, pp. 2433–2443, Sep 2022.

[9] M. White, J. Wen, M. Bowling, and D. Schuurmans, "Optimal estimation of multivariate ARMA models," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, pp. 3080–3086, Feb. 2015.

[10] U. Nallasivam, B. Srinivasan, V. Kuppuraj, M. N. Karim, and R. Rengaswamy, "Computationally efficient identification of global ARX parameters with guaranteed stability," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1406–1411, 2011.

[11] P. Whittle, "On the fitting of multivariate autoregressions, and the approximate canonical factorization of a spectral density matrix," *Biometrika*, vol. 50, no. 1-2, pp. 129–134, 06 1963.

[12] V. D. Iuliis, F. Smarra, C. Manes, and A. D'Innocenzo, "On the stability of switched ARX models, with an application to learning via regression trees," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 61–66, 2021, 7th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2021.

[13] M. Horner, S. N. Pakzad, and N. S. Gulgec, "Parameter estimation of autoregressive-exogenous and autoregressive models subject to missing data using expectation maximization," *Frontiers in Built Environment*, vol. 5, pp. 1–18, 2019.

[14] A. Isaksson, "Identification of ARX-models subject to missing data," *IEEE Transactions on Automatic Control*, vol. 38, no. 5, pp. 813–819, 1993.

[15] R. Moses and D. Liu, "Determining the closest stable polynomial to an unstable one," *IEEE Transactions on Signal Processing*, vol. 39, no. 4, pp. 901–906, 1991.

[16] T. Van Gestel, J. Suykens, P. Van Dooren, and B. De Moor, "Identification of stable models in subspace identification by using regularization," *IEEE Transactions on Automatic Control*, vol. 46, no. 9, pp. 1416–1420, 2001.

[17] M. Artin, "Algebra," Boston, MA, 2011.

[18] R. Seroul and D. O'Shea, "Programming for mathematicians," Berlin, 2000.

[19] E. I. Jury, "A simplified stability criterion for linear discrete systems," *Proceedings of the IRE*, vol. 50, no. 6, pp. 1493–1500, 1962.

[20] K. Ogata, *Discrete-time control systems (2nd ed.)*. USA: Prentice-Hall, Inc., 1995.

[21] O. Barndorff-Nielsen and G. Schou, "On the parametrization of autoregressive models by partial autocorrelations," *Journal of Multivariate Analysis*, vol. 3, no. 4, pp. 408–419, 1973.