Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 48-8 (2015) 795–800

Identification of Equation Error Models from Small Samples using Compressed Sensing Techniques

Satheesh K. Perepu * **Arun K. Tangirala** *

* *Department of Chemical Engineering, IIT Madras, Chennai, Tamilnadu 600036, India (e-mail: satheesh841@gmail.com & arunkt@iitm.ac.in).*

Abstract: System identification (SI), especially from small samples, is a challenging problem and of interest in several applications. Standard prediction-error minimization methods (PEM), under these conditions, generally result in estimates with higher variance. Moreover, in the identification of parametric models, one often needs prior knowledge of the input-output delay, obtaining estimates of which, is not possible using classical methods when the delay is either comparable or greater than the sample size. In this work, we develop a compressed sensing (CS)-based method for identifying *sparse* equation-error models that includes both *auto-regressive eXogenous* (ARX) and *AR moving average eXogenous* (ARMAX) structures with large delays, small orders and small delays with large orders, but with missing coefficients. The outcome is an iterative basis pursuit de noising (IBPDN) algorithm for solving non-linear CS problems. In addition, we propose a semi-rigorous method to lower the mutual coherence of the regressor matrix so as to obtain lower variance parameter estimates with the CS techniques. Errors in parameter estimates are computed using the bootstrapping method. Simulation studies on three diverse examples are presented to demonstrate the efficacy of the proposed methodology.

Keywords: system identification; ARX models; ARMAX models; non-linear compressed sensing; mutual coherence.

1. INTRODUCTION

Parametric system identification is concerned with developing models of a specific structure from input-output data. Most of the existing techniques including the widely used prediction-error minimization (PEM) methods (of which least squares (LS) and maximum likelihood (ML) are special cases) theoretically yield efficient and consistent estimates only under asymptotic (large sample) conditions (Ljung, 1999). Furthermore, identification of parametric models require prior specification of input-output delay and model polynomial orders. Time-delay estimation is typically carried out using impulse and frequency response methods (Bjorklund and Ljung, 2003; Selvanathan and Tangirala, 2010). Alternatively, delay may be also treated as an additional parameter and estimated simultaneously with model parameters. Regardless, both delay estimation techniques are known to work efficiently only in the presence of large samples, i.e., when the delay $d \ll M$, where M is the sample size. Order determination is usually carried out mostly using information-theoretic criteria such as Akaike information and Bayesian information criteria, respectively, with preliminary guesses generated using ideas from subspace identification (SSID). Once again the information-theoretic measures, which use the ML as their basic engine, and the SSID technieus are devised for large sample situations. However, in several applications, only data sets of *limited or small size* are available. Online estimation, set-point oriented process are a few examples of these situations. Isaksson (1991) used maximum likelihood estimation (MLE) to estimate parameters of ARX model using small number of samples. The complexity of these algorithms increase inversely with number of samples available (Bohlin, 1971). Yang et al. (2012) developed a method to identify a bioethanol plant using small number of samples. This technique is based on an orthogonal least squares algorithm and a new resampling method called output jittering. The complexity of this algorithm also increases inversely with number of samples available. Vanli and Castillo (2007) used pseudo-linear regression to estimate closed loop Box-Jenkins (BJ) models as ARMAX models from small samples. This method requires knowledge of delay and order of the process prior to estimation. To the best knowledge of authors there are no effective techniques for the estimation of parametric models, especially those that can also automatically estimate delay and order, from *small samples*. This work is concerned with a sub-class of such models, namely, the *equation-error* or the ARMAX models.

A regular ARMAX model with known model order and delay is described by the equation,

$$A(q^{-1})y[k] = B(q^{-1})u[k] + c(q^{-1})e[k] \tag{1a}$$

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a n_a q^{n_a} \tag{1b}$$

$$B(q^{-1}) = b_d q^{-d} + \cdots + b n_{b'} q^{-d-n_b} \tag{1c}$$

$$C(q^{-1}) = 1 + c_1 q^{-1} + c_2 q^{-2} + ... + c_{n_c} q^{n_c} \tag{1d}$$

where $u[k]$ and $y[k]$ are the input and measured output at sampling instant k respectively, $e[k] \sim \mathcal{N}(0, \sigma_e^2)$, d is the input-output delay and $n_{b'} = n_b + d$. An ARX model is a special case of ARMAX model with all the past terms of the innovations of $e[k]$ *i.e.* $c_1 = c_2 = \cdots = c_{n_c} = 0$ set to zero. ARX models give rise to linear predictors, thereby permitting the use of linear LS methods to generate unique parameter estimates. ARMAX models, on the other hand, as evident from (1a), result in non-linear predictors. A non-linear LS estimator has to be thus employed, wherein the optimum is searched numerically and one has to be usually content with a local optimum. As remarked earlier, these estimators, linear or non-linear, require the user to specify a prior the time-delay and order. More importantly, the non-linear estimator is efficient only under asymptotic conditions.

2405-8963 © 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.09.066

The objective of this work is to present an effective and practically elegant method for identification of *sparse* ARMAX models from *small samples* using the ideas of CS (Donoho, 2006), while automatically determining the delay and order of the model. The term "small" especially refers to the case of $d \gg M$, or in general, $\dim(\boldsymbol{\theta}) + d \gg M$ with the additional requirement that $\boldsymbol{\theta}$ is sparse. The *key idea* is to recognize that a regular ARMAX model structure with known delay and order manifests as a *sparse* ARMAX model when the same are unknown, especially when the delays are large. Referring to (1a), when the delays and orders are unknown, all inputs beginning from $u[k-1]$ and outputs up to a remote past have to be included, thereby increasing the size of the parameter vector. However, the augmented parameter vector, call it $\boldsymbol{\theta}_s$ (see (15)), is *sparse*, i.e., contains several zero-valued entries with only a few non-zero parameters. Thus, systems described by lower-order, large delay, ARMAX models with unknown delays and orders manifest as *sparse* ARMAX processes. Further mathematical formalities are presented in §3.

This paper presents an iterative BPDN (IBPDN) approach to estimate parameters of *sparse* ARMAX models, alternatively, those of regular ARMAX models without the prior knowledge of delay and order, from small samples, using the ideas of CS. It may be remarked that we are essentially dealing with an *underdetermined* problem. See §2 for a brief review of the linear CS optimization problem and the BPDN algorithm.

The idea of using CS techniques in system identification is relatively new, while that of estimating sparse parameter vectors has been around a little longer with both areas of research being highly active (Sanandaji et al., 2011; Ljung et al., 2011; Ozay et al., 2011). Sparsity requirements (on parameters) in system identification have been imposed usually from a regularization viewpoint, but still in the presence of large samples. The small sample case has been studied predominantly by Sanandaji et al. (2011) to estimate ARX models for linear time-invariant (LTI) and piecewise continuous linear time-varying (LTV) systems without the knowledge of delay and order, from few observations. However, the development is restricted to *deterministic* systems only. Moreover a block sparse structure of the signal is assumed in the algorithm. A few other limitations also exist, as explained below. Identification of ARMAX models using CS ideas can be viewed as solving non-linear CS problem. In non-linear CS problem, the measurements are assumed as non-linear projections of sparse vector. Blumensath (2013) linearised the non-linear function around the sparse vector at every iteration to arrive at the solution.

The success of the proposed IBPDN algorithm rests on two important factors, namely, the *mutual coherence* of the regressor matrix, which can be loosely interpreted as the *orthogonality* of the matrix and (ii) the *sparsity of the initial conditions*. The prominence of the first factor arises from a theoretical result in CS literature, which states that for perfect recovery of sparse signals from measurements, it is necessary to have the mutual coherence of the so-called *dictionary* (in system identification, this dictionary is the regressor matrix, see §3) remain below a certain bound. In this context, Sanandaji et al. (2011), through simulation studies, determine that the impact of high mutual coherence is not as serious as theory suggests it to be. However, it remains a fact that the error bounds decrease with the lowering of mutual coherence, and therefore it is imperative that the regressor matrix have as low as mutual coherence. Furthermore the observation by Sanandaji et al. (2011) is based on simulation studies and not rigorously conclusive. In this respect, Sanandaji et al. (2011) suggest a pre-filtering based technique for reducing mutual coherence. The method is, however, not effective in the sense that the coherence of the pre-filtered regressor matrix is much higher than the required theoretical bound. In the current work, a method based on an optimal scaling of the regressor matrix is proposed to lower the mutual coherence of the regressor matrix.

The initial conditions required for the identification of the sparse ARMAX model using the proposed algorithm are generated by solving a sparse pseudo-linear regression problem, followed by an iterative optimization of the same to finally generate a sparse set of initial conditions. The case studies in §5 demonstrate the efficacy of this approach. Finally, the convergence of the proposed algorithm is studied, albeit through simulations. A theoretical study of convergence involves a very rigorous study, which is beyond the scope of this paper. The error decay rates across iterations hold promise for the convergence of the proposed algorithm.

To summarize the novelty of this work, we present IBPDN algorithm for the estimation of ARMAX models from small samples with unknown delays and orders and an optimal scaling or transformation method for the reduction of mutual coherence. An added advantage of the proposed method over the method proposed in Blumensath (2013) is that the iterative update of initial guess to ensure sparser solution. The foregoing discussion establishes the significance of the novelty.

The rest of the paper is organized as follows. The necessary essentials of compressed sensing are reviewed in Section 2. Section 3 presents the proposed algorithm for identification of equation error models from small samples. The transformation method for reducing mutual coherence of the regressor matrix is discussed in Section 4. Results from simulation studies are presented in Section 5. The paper closes with a few concluding remarks in Section 6.

2. FOUNDATIONS OF COMPRESSED SENSING

One of the main problems of interest in the compressed sensing literature is to recover N observations of a signal $x[k]$ from its M random measurements ($M \ll N$), call it $y[k]$, under the premise that $x[k]$ is K-sparse (number of non-zeros) ($K < M$) in some basis space Donoho (2006). The measurement vector $y[k]$ can be in the same domain as that of $x[k]$, for e.g., randomly subsampled $x[k]$, or in some other transformed domain. Formally,

$$\mathbf{y} = \Pi\mathbf{x}, \quad \text{and} \quad \mathbf{x} = \mathbf{B}\mathbf{s} \tag{2}$$

$$\implies \mathbf{y} = \mathbf{A}\mathbf{s} \tag{3}$$

where Π is $M \times N$ measurement matrix (when $\mathbf{y}$ is subsampled $\mathbf{x}$, it consists of ones and zeros), $\mathbf{s} \in \mathbb{R}^{N\times 1}$ is the sparse representation of $\mathbf{x}$ in some *sparsifying* basis space defined by the matrix $\mathbf{B}$. In (3), the matrix $\mathbf{A} = \Pi\mathbf{B}$ is said to be the *dictionary* for $\mathbf{y}$, which may be known a priori (fixed) or unknown (adaptive). The goal in compressed sensing is to recover $\mathbf{x}$ by first estimating the sparse representation $\mathbf{s}$ given $\mathbf{A}$ or simultaneously along with $\mathbf{A}$ when it is unknown (Perepu and Tangirala, 2013).

An N-long vector $\mathbf{s}$ is said to be sparse if the number of non-zero elements in $\mathbf{s}$ is less than the number of zero-valued elements. There exist several measures of sparsity. See Hurley and Rickard (2009) for a comprehensive study of different sparsity measures. The zero-norm of a vector, given by

$$||\mathbf{s}||_0 = \dim\{i|s_i \neq 0\} \tag{4}$$

is frequently used in the CS literature as a measure of sparsity.

If the dictionary $\mathbf{A}$ is known a priori, the sparse representation $\mathbf{s}$ is recovered by solving the optimization problem

$$\min_{\mathbf{s}} ||\mathbf{s}||_0 \quad \text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{s} \tag{5}$$

However, the problem is non-convex and NP-hard. A widely used alternative is to replace with a convex 1-norm optimization to solve (3), as put forth by Donoho (2006):

$$\min_{\mathbf{s}} ||\mathbf{s}||_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{s} \tag{6}$$

Candès (2008) proved that the zero-norm solution equals 1-norm solution as long as $\mathbf{A}$ satisfies what is known as the Restrictive Isometric Property (RIP), given by

$$(1-\delta)\,||\mathbf{s}||_2^2 \leq ||\mathbf{A}\mathbf{s}||_2^2 \leq (1+\delta)\,||\mathbf{s}||_2^2 \tag{7}$$

It is difficult to verify RIP property in practice. Elad (2010) replaced RIP with a weaker condition based on *mutual coherence*. The mutual coherence of matrix $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_p]$ is given by

$$\mu(\mathbf{X}) = \frac{1}{p}\max_{i\neq j}\frac{\left|\mathbf{x}_i^T\,\mathbf{x}_j\right|}{||\mathbf{x}_i||_2\,||\mathbf{x}_j||_2} \tag{8}$$

where $||.||_2$ is the standard 2-norm. The mutual coherence of any matrix always satisfies $1/\sqrt{p} \leq \mu(X) \leq 1$; see (Romberg, J. and Wakin, 2007). In order to guarantee recovery of the sparse vector, the mutual coherence of the dictionary A should be as low as possible (Elad, 2010). The precise requirement is that at least K columns of $\mathbf{A}$ be orthogonal. In this respect mutual coherence is a conservative measure, *i.e.*, it need not be zero for guaranteed recovery but should be bounded above, see (24) in Section 4.

Note that the above definition of mutual coherence is valid for only deterministic matrices. In applications to (linear) system identification problems, the matrix $\mathbf{A}$ is the regressor matrix Φ while $\mathbf{s}$ is the sparse parameter vector (see Section 3). In most cases, the regressor matrix contains lagged measurements which have errors in them. Thus, strictly speaking, the definition of mutual coherence in (8) cannot be used as is. In this work, nevertheless with some abuse of definition, we apply it as is to the regressor matrix. A study concerning the extension of mutual coherence definition to the class of stochastic matrices is underway. For now, (8) may be treated as an estimate of the "true" mutual coherence matrix.

There exist two classes of algorithms to solve (6) (Elad, 2010). One class is that of greedy algorithms and other comprises iterative algorithms. Greedy algorithms include orthogonal matching pursuit (OMP), block-OMP, *etc*, while the iterative algorithms include least absolute shrinkage and selection operator (LASSO), basis pursuit (BP), basis pursuit de noising (BPDN), *etc*. Among these the widely used techniques to solve (6) are the LASSO and BPDN. We choose to employ the BPDN in this work due to its robustness (to noise) property. Another advantage is that even when $\mathbf{s}$ is not sufficiently sparse, $\mathbf{y}$ can be approximated with a small error. In BPDN (6) is solved as

$$\min_{\mathbf{s}} ||\mathbf{s}||_1 \quad \text{such that} \quad ||\mathbf{y} - \mathbf{A}\mathbf{s}||_2^2 \leq \zeta \tag{9}$$

where ζ is a small non-zero value. There are several flavours of the BPDN algorithm. The spectral gradient-based BPDN (SPG-BPDN) is used in this work.

3. PROPOSED IBPDN ALGORITHM FOR SI

The ARMAX model in (1a) can be written as

$$y[k] = \boldsymbol{\varphi}^T[k]\boldsymbol{\theta} + e[k] \tag{10}$$

where

$$\begin{aligned}\boldsymbol{\varphi}[k] = \big[&y[k-1]\ y[k-2]\ \cdots\ y[k-n_a]\\ &u[k-d]\ u[k-d-1]\ \cdots\ u[k-n_b-d]\\ &e[k-1]\ e[k-2]\ \cdots\ e[k-n_c]\big]^T \in \mathbb{R}^{n_a+n_b+n_c\times 1}\\ \boldsymbol{\theta} = \big[&a_1\ a_2\ \cdots\ a_{n_a}\ b_d\ b_{d+1}\ \cdots\ b_{n_b'}\\ &c_1\ c_2\ \cdots\ c_{n_c}\big]^T \in \mathbb{R}^{n_a+n_b+n_c\times 1}\end{aligned}$$

and $e[k] \sim \text{GWN}(0, \sigma_e^2)$. In the rest of this sequel, we shall refer to (10) as ARMAX($n_a, n_{b'}, n_c, d$) model where d is the unknown input-output delay the existence of which implies $b_0 = b_1 = \cdots = b_{d-1} = 0$.

The goal of system identification is to identify the $(n_a + n_{b'} + n_c) \times 1$ parameter vector $\boldsymbol{\theta}$ using M consecutive observations of output y for a specified n_a, n_c and d. Taking M consecutive measurements and putting them in the vector form gives

$$\mathbf{y} = \Phi\boldsymbol{\theta} + \mathbf{e} \tag{11}$$

where

$$\mathbf{y} = \big[y[k_0]\ y[k_0+1]\ \cdots\ y[k_0+M-1]\big]^T \tag{12a}$$

$$\Phi = \big[\boldsymbol{\varphi}^T[k_0]\ \boldsymbol{\varphi}^T[k_0+1]\ \cdots\ \boldsymbol{\varphi}^T[k_0+M-1]\big]^T \tag{12b}$$

$$\mathbf{e} = \big[e[k_0]\ e[k_0+1]\ \cdots\ e[k_0+M-1]\big]^T \tag{12c}$$

where $k_0 \geq 0$ is a suitable starting sample determined by the values of n_a, n_c and d.

When n_a,n_b and d are known / specified, the parameter vector $\boldsymbol{\theta}$ can be solved by well-established methods such as pseudo-linear regression, Gauss-Newton method, or a PEM method. As remarked in §1, it is possible to obtain these estimates efficiently from large samples, typically through non-parametric and information-theoretic methods. However, when the delays are large and/or that the orders are high, but with missing coefficients, classical methods fail. Thus, we are forced to jointly estimate the much longer (than $\boldsymbol{\theta}$) $P \times 1$ parameter vector $\boldsymbol{\theta}_s$,

$$\boldsymbol{\theta}_s = \big[a_1\ a_2\ \cdots\ a_{n_a}\ a_{n_a+1:N_a} \tag{13}$$

$$b_{0:d-1}\ b_d\ b_{d+1}\ \cdots\ b_{n_b'}\ b_{n_b'+1:N_b} \tag{14}$$

$$c_1\ c_2\ \cdots\ c_{n_c}\ c_{n_c:N_c}\big]^T \in \mathbb{R}^{N_a+N_b+N_c\times 1} \tag{15}$$

where $a_{n_a+1:N_d}$, $b_{0:d-1}$, $b_{n_b'+1:N_b}$ and $c_{n_c+1:N_c}$ are all *truly zero-valued* parameters. Therefore, the true $\boldsymbol{\theta}_s$ is *sparse*, as indicated below:

$$\boldsymbol{\theta}_s = \big[a_1\ a_2\ \cdots\ a_{n_a}\ \mathbf{0}\ b_d\ b_{d+1}\ \cdots\ b_{n_b'}\ \mathbf{0} \tag{16}$$

$$c_1\ c_2\ \cdots\ c_{n_c}\ \mathbf{0}\big]^T \in \mathbb{R}^{N_a+N_b+N_c\times 1} \tag{17}$$

In practice, this is ensured by choosing such that $(P - M) \gg M$, where $P = N_a + N_b + N_c$. It is important to note that there is a corresponding increase in the size of Φ. Denoting the new matrix by Φ_s, we have

$$\mathbf{y} = \Phi_s\boldsymbol{\theta}_s + \mathbf{e} \tag{18}$$

The problem statement is now as follows. Given M observations of input-output data, $M \ll P$, the goal is estimate the sparse vector $\boldsymbol{\theta}_s$. Once $\boldsymbol{\theta}_s$ is estimated, it is easy to extract $\boldsymbol{\theta}$. If the regressor matrix Φ_s is known, then the problem simplifies to a regular CS problem, and a BPDN algorithm can be employed to recover the sparse $\hat{\boldsymbol{\theta}}_s$. This is the case of estimating a sparse ARX model. Table 1 outlines the algorithm. Note that the algorithm involves the determination of a pre-filtering matrix $\mathbf{T}$ introduced for the purpose of reducing mutual coherence. Section 4 presents means of determining this matrix. In addition, the last step computes the significance levels for the

obtained parameter estimates using the bootstrapping method for dynamic processes (Efron and Tibshirani, 1993).

Turning to the focal problem of this work, which is that of estimating ARMAX models, we note that the regressor matrix in (18) is unknown - rather an implicit function of $\boldsymbol{\theta}_s$. Therefore, a non-linear optimization problem is involved, for which we propose the IBPDN algorithm, wherein at each iteration, we solve a BPDN algorithm to ensure sparse increments. Naturally this has to be complemented by a sparse initial guess, a method for which is described later.

The proposed IBPDN method sets up an iterative algorithm for the parameter estimate:

$$\hat{\boldsymbol{\theta}}_j^{(i+1)} = \hat{\boldsymbol{\theta}}_j^{(i)} + \gamma \triangle \boldsymbol{\theta} \tag{19}$$

where the user-specified parameter γ controls the convergence rate (Nocedal and Wright, 1999). In this work, we set $\gamma = 0.2$ based on experience. Note that for simplicity we have dropped the subscript $\mathbf{s}$ on $\boldsymbol{\theta}$, which shall be followed for the rest of this article.

The update $\triangle\boldsymbol{\theta}$ is now computed using the standard BPDN algorithm so as to obtain sparse increments. Assume an initial value for $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_j^{(i)}$, where j keeps track of the iteration for initial guess, while the superscripts keeps track of the iterations in (19) for a given guess. Thus, $j = 1$ corresponds to the *first* initial guess and $i = 1$ corresponds to the first iteration for (19). Compute prediction errors $\boldsymbol{\epsilon}$ using the initial model. For a fixed j, the following equation is solved using BPDN algorithm at each iteration of the algorithm:

$$\min_{\triangle\boldsymbol{\theta}} ||\triangle\boldsymbol{\theta}||_1 \quad \text{subject to} \quad ||\boldsymbol{\epsilon} - \boldsymbol{\Psi}\triangle\boldsymbol{\theta}||_2^2 \leq 0.001 \tag{20}$$

where $\boldsymbol{\Psi}$ is gradient of predictor equation *w.r.t* parameter vector. For ARMAX models, gradient of the predictor equation is the filtered regressor matrix $\boldsymbol{\Phi}$.

$$\boldsymbol{\Psi} = \frac{1}{C(q^{-1})}\boldsymbol{\Phi} \tag{21a}$$

$$\text{where } C(q^{-1}) = 1 + c_1 q^{-1} + c_2 q^{-2} + c_3 q^{-3} + \cdots + c_{n_c} q^{-n_c} \tag{21b}$$

Update $\boldsymbol{\theta}_j^{(i+1)}$ as per (19). The above mentioned steps are repeated until convergence criteria chosen for parameter vector. The convergence criteria in this paper is considered to be

$$\left|\left|\hat{\boldsymbol{\theta}}_j^{(i+1)} - \hat{\boldsymbol{\theta}}_j^{(i)}\right|\right|_2 < \zeta_1 \tag{22}$$

where ζ_1 is the standard user-defined parameter determining the trade-off between time taken for convergence and final error achieved. The value of ζ_1 is chosen to be 0.01 in this paper. Once the algorithm converges for a fixed j to $\hat{\boldsymbol{\theta}}_j$, the algorithm is again run for an another initial condition and $j \to j+1$.

$$\boldsymbol{\theta}_{j+1}^{(1)} = \hat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^{(1)}$$

The above described steps are repeated until the following criteria satisfied

$$\left|\left|\hat{\boldsymbol{\theta}}_{j+1}^{1} - \hat{\boldsymbol{\theta}}_j^{(1)}\right|\right|_2 < \zeta_2 \tag{23}$$

where ζ_2 is also user-defined parameter which is to be chosen on guidelines of ζ_1. The first initial guess is obtained using the pseudo-linear regression (PLR). To generate the prediction errors for the PLR method, a suitable sparse ARX model is fit to the data.

In passing, we may remark that we have not explicitly indicated the mutual coherence reduction (of the predictor gradient) step in the ARMAX model estimation, but it can be included in the same way as indicated for the ARX model.

Table 1. Algorithm for identification of ARX models from small samples

1. For a specified P, set up the regressor matrix (Φ) according to (18).
2. Compute the optimal transformation matrix $\mathbf{T}$ that minimizes the mutual coherence $\mu(\mathbf{T}\Phi_s)$ by solving (26).
3. Solve for sparse $\boldsymbol{\theta}$ using BPDN algorithm. The optimization problem is given by
$$\min_{\boldsymbol{\theta}} \ ||\boldsymbol{\theta}||_1 \quad \text{subject to} \quad ||\mathbf{T}\mathbf{y} - \mathbf{T}\Phi\boldsymbol{\theta}||_2^2 \leq \epsilon$$
4. Obtain the significance limits of $\hat{\boldsymbol{\theta}}$ using the bootstrapping method (Efron and Tibshirani, 1993).

Table 2. IBPDN algorithm for estimation of ARMAX models

Initialize $j = 1, i = 1$

1. Calculate the initial value for $\hat{\boldsymbol{\theta}}_j^{(i)}$ using PLR
2. Compute prediction errors $\boldsymbol{\epsilon}$ for the model having parameter vector $\hat{\boldsymbol{\theta}}_{\mathbf{j}}^{(\mathbf{i})}$
3. Solve for $\triangle\boldsymbol{\theta}$ using (20)
4. Update $\boldsymbol{\theta}_j^{(i+1)}$ using the following formula
$$\hat{\boldsymbol{\theta}}_j^{(i+1)} = \hat{\boldsymbol{\theta}}_j^{(i)} + \gamma\triangle\boldsymbol{\theta}$$
5. Repeat steps 2-5 untill convergence condition satisfied
$$\left|\left|\hat{\boldsymbol{\theta}}_j^{(i+1)} - \hat{\boldsymbol{\theta}}_j^{(i)}\right|\right|_2 < \zeta_1$$
6. Repeat the above steps for the new initial condition $\hat{\boldsymbol{\theta}}_{j+1}^{(1)}$ until it satisfies
$$\left|\left|\hat{\boldsymbol{\theta}}_{j+1}^{(1)} - \hat{\boldsymbol{\theta}}_j^{(1)}\right|\right|_2 < \zeta_2$$

4. REDUCING MUTUAL COHERENCE OF MATRIX Φ

As described in §2, the guaranteed unique solution of (6) depends on the mutual coherence of regressor matrix Φ. For recovery algorithms like OMP and BP the upper bound on mutual coherence is given by

$$\mu(\Phi) \leq \mu_U(\Phi) = \frac{1}{2K-1} \tag{24}$$

where K is the sparsity index of the sparse vector as before. When $K = 1$, it is sufficient to have $\mu < 1$, with $K = 2$, to have $\mu < 1/3$ and so on. Stated other way, a smaller coherence of the regressor matrix facilitates recovery of signals with lower sparsity, *i.e.*, more non-zero elements. The mutual coherence of a matrix $\mathbf{A}$ is a measure of the correlation between columns of $\mathbf{A}$. Therefore, when $\mathbf{A}$ is orthogonal $\mu(\mathbf{A}) = 0$ and vice versa. In SI applications, $\mathbf{A} = \Phi$ and as remarked earlier, the mutual coherence of Φ can be usually quite high due to its composition given in (12c).

In this paper, we propose a method to decrease mutual coherence of regressor matrix. The aim is to determine a pre-multiplying matrix $\mathbf{T}$ of order $M \times M$ such that obtained matrix $\mathbf{T}\Phi$ has a mutual coherence *as low* as possible. Then, (11) transforms to

$$\mathbf{T}^\star\mathbf{y} = \mathbf{T}^\star\Phi\boldsymbol{\theta} + \mathbf{T}^\star\mathbf{e} \tag{25}$$

where $\mathbf{T}^\star$ emerges from the optimization

$$\mathbf{T}^\star = \arg \min_{\mathbf{T}} f(\mathbf{T}) \quad \text{such that} \quad f(\mathbf{T}) \geq \mu(\mathbf{T}\Phi) \tag{26a}$$

$$\text{where} \quad f(\mathbf{T}) = ||\mathbf{T}'\mathbf{T} - \mathbf{I}_{m\times m}||_F^2 \tag{26b}$$

The above equation ensures the lowering of mutual coherence along with the orthogonalization of matrix $\mathbf{T}$. The advantage

is that the properties of vector $\mathbf{Te}$ in (25) is same as properties of $\mathbf{e}$ in (12c). Notice from (25) that the functional relationship is preserved, i.e., the same $\boldsymbol{\theta}$ relates the transformed variables. These observations are similar to those witnessed with the use of pre-filters in classical identification. In fact the approach followed in (25) is a generalization of the pre-filtering concept in identification, where it is necessary to apply the same pre-filter to both input and output to preserve the functional relationship. Sanandaji et al. (2011) apply the pre-filtering idea, but with no explicit control on mutual coherence and consequently not necessarily satisfying (24).

A numerical optimizer (such as *fmin* in MATLAB®) is used to determine the optimal $\mathbf{T}$. The *idea of using optimization here is not to find explicitly the minimum value of the function, but rather to obtain a local minimum that can lower the mutual coherence to a reasonable value.* In this sense, (26) is a non-convex problem that defeats the purpose of 1-norm minimization in the CS arena, but we are not truly interested in the global optimum.

With an optimal $\mathbf{T}$ in hand, the optimization problem in (9) is solved in the transformed space:

$$\min_{\boldsymbol{\theta}} ||\boldsymbol{\theta}||_1 \quad \text{such that } ||\mathbf{Ty} - \mathbf{T}\Phi\boldsymbol{\theta}||_2^2 \leq \epsilon \tag{27}$$

The estimate of matrix $\mathbf{T}$ depends on the initial guess. Hence the proposed algorithm is tried for different initial values of $\mathbf{T}$ and considerable reduction of mutual coherence is observed when $\mathbf{T}^{(0)} = \mathbf{I}_{M\times M}$ even when other choices of $\mathbf{T}^{(0)}$ produced lower initial values of mutual coherence. The causes underlying this observation are presently under study.

The proposed method for reducing mutual coherence is tested on an arbitrary 20×40 matrix Φ with a mutual coherence value of 0.842. Solving the optimization problem in (26) with the initial guess set to identity the mutual coherence decreased to 0.045 suggesting that this method is effective in reducing mutual coherence of the matrix.

5. SIMULATION RESULTS

The proposed algorithm is tested on three simulated examples as presented below. Signal to noise ratio (SNR) is maintained at 10 for all the examples.

5.1 *Example* 1

Consider a 2-parameter ARX$(40, 40)$ data generating process given in (28) below.

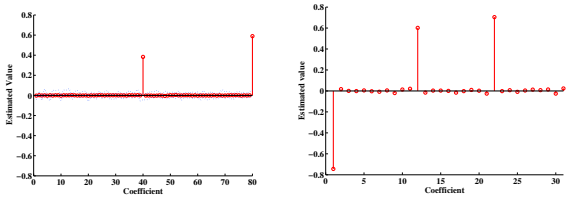
$$y[k] = 0.4y[k-40] + 0.6u[k-40] + e[k] \tag{28}$$

The objective is to identify a very high-order sparse system from 50 measurements using the proposed algorithm, without the knowledge of delay and order of the process. The regressor matrix Φ_s is constructed from 40 past outputs and 40 past inputs. The mutual coherence of Φ_s is lowered using the method discussed in §4. The true parameter vector should have non-zero values only at locations $l = 40, 80$.

A stem plot of the estimated parameter vector is shown in Figure 1(a). From the plot, only two parameter values are non-zero corresponding to the equation given in (28). Variability in parameter estimates are calculated from 1000 bootstrap simulations and are reported below each estimated value of the parameter.The identified system is given by

$$\hat{y}[k] = \underset{\pm 0.0341}{0.371}\, y[k-40] + \underset{\pm 0.014}{0.587}\, u[k-40]$$

which is in close agreement with the true process. To show the effect of mutual coherence on solution obtained by solving



(a) ARX$(40, 40)$ model (b) ARMAX$(1, 1, 1)$

Fig. 1. Parameter estimates obtained using proposed algorithm

(11), the model identified *without* lowering mutual coherence of the regressor matrix is reported below:

$$\hat{y}[k] = \underset{\pm 0.0514}{0.37}\, y[k-40] + \underset{\pm 0.022}{0.564}\, u[k-40]$$

For comparison purposes, estimates obtained from conventional LS technique with a pre-specification of the true model structure (order and delay) are reported below.

$$\hat{y}[k] = \underset{\pm 0.0541}{0.371}\, y[k-40] + \underset{\pm 0.026}{0.589}\, u[k-40]$$

The superiority of the proposed method is thus demonstrated for this example.

In the remaining two examples, the parameters of the model are obtained from 50 samples of data while *the regressor matrix consists of past ten outputs, current input and past ten inputs and past ten innovations*, *i.e.*, of dimension 19×31. Note that the number of past outputs / inputs / innovations (ten here) is not a rigid choice. The initial value for the proposed algorithm is obtained from PLR method described in §3. Owing to the difficulty associated with the variability of parameter estimates obtained using current algorithm, the variance has not been reported for the following examples.

5.2 *Example* 2

Consider a three-parameter ARMAX$(1, 1, 1)$ data generating process described by

$$A(q^{-1}) = 1 + 0.75q^{-1},\ B(q^{-1}) = 0.6q^{-1},\ C(q^{-1}) = 1 - 0.7q^{-1}$$

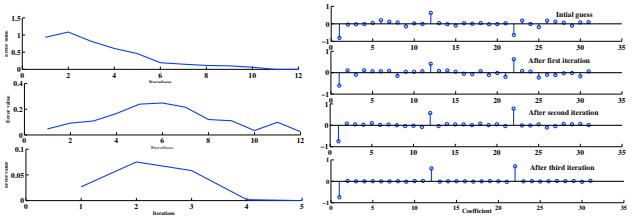
The model is simulated to generate 50 samples of data using a band-limited PRBS input. The proposed algorithm is used to estimate parameters of the simulated model. In addition the mutual coherence of the regressor matrix is lowered by using the method discussed in §4. For the sparse parameter vector $\boldsymbol{\theta}_s$, only three (the first, twelfth and twenty-second) values are non-zero. A stem plot of $\hat{\boldsymbol{\theta}}_s$ is shown in Figure 1(b).

From the plot, only three values are significant values, suggesting a 3-parameter ARMAX model. The model identified using proposed algorithm is thus,

$$\hat{A}(q^{-1}) = 1 + 0.748q^{-1},\ \hat{B}(q^{-1}) = 0.55q^{-1},\ \hat{C}(q^{-1}) = -0.675q^{-1}$$

A plot of the error obtained using proposed algorithm across iterations is shown in Figure 2(a). The plots are arranged from top to bottom with ascending order of iteration. From the plots it is evident that the algorithm converges quickly as the initial value is refined each time, which is shown in Figure 2(b). The graphs are arranged in same way as that of Figure 2(a) where the topmost one corresponds to the initial value obtained through the PLR method. From the plot, it is clear that the the parameter vector obtained after third iteration is close to the true parameter vector.

The model obtained using the proposed algorithm is in comparison with the true model. Hence the proposed algorithm can be



(a) Error in parameter estimates vs iterations (b) Initial values obtained for different iterations

Fig. 2. ARMAX$(1, 1, 1)$ model

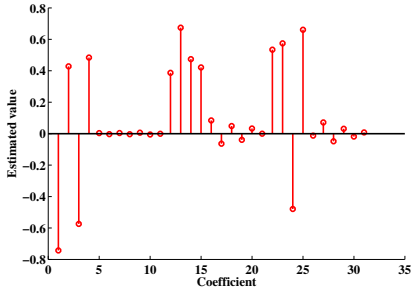


Fig. 3. Parameter estimates of the process in Example 3

used to identify ARMAX models from small samples without an estimate of delay and order of the process.

5.3 Example 3

The data generating process has a 12-parameter ARMAX$(4, 4, 1)$ structure described by

$$A(q^{-1}) = 1 + 0.75q^{-1} - 0.45q^{-2} + 0.6q^{-3} - 0.5q^{-4}$$
$$B(q^{-1}) = 0.4q^{-1} + 0.7q^{-2} - 0.5q^{-3} + 0.4q^{-4}$$
$$C(q^{-1}) = 1 + 0.55q^{-1} + 0.6q^{-2} - 0.5q^{-3} + 0.65q^{-4}$$

A band-limited PRBS is used to generate 50 samples of the output. Note that here the entries of parameter vector are non-zero only at locations $l \in [1\text{–}4, 12\text{–}15, 22\text{–}25]$.

The proposed algorithm is used to estimate the parameters of the simulated model. A stem plot of the estimated parameter vector is shown in Figure 3. From the plot it is evident that the parameter estimates are non-zero only at locations $l \in [1\text{–}4, 12\text{–}15, 22\text{–}25]$, clearly indicating that the process follows an ARMAX$(4, 4, 1)$ model. The model thus identified using the proposed algorithm is

$$\hat{A}(q^{-1}) = 1 + 0.743q^{-1} - 0.428q^{-2} + 0.574q^{-3} - 0.484q^{-4}$$
$$\hat{B}(q^{-1}) = 0.387q^{-1} + 0.674q^{-2} - 0.474q^{-3} + 0.421q^{-4}$$
$$\hat{C}(q^{-1}) = 0.534q^{-1} + 0.574q^{-2} - 0.479q^{-3} + 0.661q^{-4}$$

once again establishing the efficacy of the method.

All the above examples demonstrate that the proposed technique rightly identifies the delay, order and delivers satisfactory estimates of the parameters.

6. CONCLUSIONS

In this paper we presented two important contributions: (i) a method for estimation of ARMAX model from small number of samples without any prior knowledge of delay and order and (ii) a method for lowering the mutual coherence of regressor matrix for obtaining efficient estimates. The proposed IBPDN algorithm to identify ARMAX models is a modified version of regular BPDN algorithm used in CS techniques. A natural outcome of using the compressed sensing techniques is that it does not require the knowledge of delay and order of the process prior to estimation of parameters.

The method of lowering mutual coherence is demonstrated on a simulated ARX model and errors in parameter estimates are calculated using boot-strapping techniques. The errors obtained are lower than that of obtained using least square technique. The proposed algorithm for ARMAX model estimation is demonstrated using three simulated examples. All the case studies demonstrate the efficacy of the proposed method and show good convergence rates for the error. Future directions involve a rigorous study of the convergence of the algorithm and extensions to the estimation of BJ models.

REFERENCES

Bjorklund, S. and Ljung, L. (2003). A review of time-delay estimation techniques. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 3, 2502–2507 Vol.3.

Blumensath, T. (2013). Compressed sensing with nonlinear observations and related nonlinear optimization problems. *Information Theory, IEEE Transactions on*, 59(6), 3466–3474.

Bohlin, T. (1971). On the problem of ambiguities in maximum likelihood identification. *Automatica*, 7(2), 199–210.

Candès, E.J. (2008). The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9–10), 589 – 592.

Donoho, D. (2006). Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4), 1289–1306.

Efron, B. and Tibshirani, R.J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall, New York.

Elad, M. (2010). *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Publishing Company, Incorporated, 1st edition.

Hurley, N. and Rickard, S. (2009). Comparing measures of sparsity. *Information Theory, IEEE Transactions on*, 55(10), 4723–4741.

Isaksson, A. (1991). System identification subject to missing data. In *American Control Conference, 1991*, 693–698.

Ljung, L. (1999). *System Identification - A Theory for the User*. Prentice Hall International, New Jersey, USA.

Ljung, L., Hjalmarsson, H., and Ohlsson, H. (2011). Four encounters with system identification. *European Journal of Control*, 17(5-6), 449–471.

Nocedal, J. and Wright, S.J. (1999). *Numerical Optimization*. Springer New York.

Ozay, N., Sznaier, M., Lagoa, C., and Camps, O. (2011). A sparsification approach to set membership identification of switched affine systems. *IEEE Transactions on Automatic Control*, 57(3), 634–648.

Perepu, S. and Tangirala, A.K. (2013). An adaptive basis estimation method for compressed sensing with applications to missing data reconstruction. In *DYCOPS 2013*, 190–196. IFAC, Mumbai, India.

Romberg, J. and Wakin, M. (2007). Compressed Sensing: A Tutorial. In *IEEE Statistical Signal Processing Workshop,*.

Sanandaji, B., Vincent, T., Wakin, M., Toth, R., and Poolla, K. (2011). Compressive system identification of lti and ltv arx models. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 791–798.

Selvanathan, S. and Tangirala, A. (2010). Time-delay estimation in multivariate systems using hilbert transform relation and partial coherence functions. *Chemical Engineering Science*, 65(2), 660 – 674.

Tropp, J. (2006). Just relax: convex programming methods for identifying sparse signals in noise. *Information Theory, IEEE Transactions on*, 52(3), 1030–1051.

Vanli, O.A. and Castillo, E.D. (2007). Closed-loop system identification for small samples with constraints. *Technometrics*, 49(4), 382–394.

Yang, J., Wei, H.L., Kadirkamanathan, V., and Lin, X. (2012). System identification from small data sets using an output jittering method with application to model estimation of bioethanol production. In *ICMLC*, 949–955.