Linear Dynamic Model Identification and Data Reconciliation using Dynamic Iterative PCA (DIPCA)

Vipul Mann, Arun K. Tangirala, Shankar Narasimhan

Department of Chemical Engineering
Indian Institute of Technology Madras, Chennai, Tamil Nadu, India

Abstract

Identification of input-output models from data is of utmost relevance in chemical process industries and has applications in process monitoring, control and fault diagnosis. Input-output data used in such identification exercises often has measurement errors in both the variables. Model identification under such conditions translates to solving an errors-in-variables (EIV) problem which is difficult to solve using classical system identification techniques. A recently proposed method - Dynamic Iterative Principal Component Analysis (DIPCA) uses PCA framework to identify the process order, delay, model parameters, and error variances. DIPCA, however, has certain shortcomings under small sample conditions which limit its practical applications.

In this work, we address these shortcomings, namely ambiguity in order determination under small sample cases and arbitrary selection of stacking lag which leads to sub-optimal parameter estimates. We define a metric called 'd-selective eigenvalue ratio', or *d-SEVR* that sharply identifies the true order even for small sample cases. We also demonstrate the existence of an optimal stacking lag corresponding to the lowest error in estimation of error-covariance matrix. Finally, we use the identified model to obtain reconciled estimates of variables using Kalman Filter.

Keywords: Identification; Principal Component Analysis; Errors-in-Variables; Kalman Filter

1. Introduction

While performing data generation exercises in a plant, measurements are often corrupted with errors which could incur due to instrumentation error, observational error, situational error, or any other unknown source of error. The variances of such errors across both input and output variables are usually unequal i.e. the errors are heteroskedastic in nature. In the process industry, such data is predominantly used for optimizing control loop performances, making predictions about a given process variable, and monitoring process operations - all of which have model identification at their core. Therefore, it becomes extremely important to identify the underlying data generating process from erroneous measurements of the plant variables without any prior knowledge about the process.

Due to the presence of errors in both the variables, model identification under these conditions translates to solving an errors-in-variables (EIV) problem and it gets difficult to apply classical system identification techniques which are based on the assumption of error-free inputs. There exist different techniques for solving the EIV problem but the total least squares (TLS) formulation gives the most efficient solution. In [1], a way to use the instrumental variable techniques is proposed to solve EIV regression problem. On the other hand, multivariable statistical data analysis techniques have proven themselves to be highly successful in finding out linear relations between variables corrupted with measurement errors. Principal

Component Analysis (PCA) is one of the most widely used techniques for solving the TLS problem because of its simplicity, numerical robustness and computational efficiency. While PCA has widespread use in dimensional reduction, fault detection, and feature extraction, we focus on its application to solving the regression problem.

The generic formulation of PCA, which we call static PCA, is suitable for identifying steady-state or instantaneous linear relationships between variables. Dynamic PCA or DPCA was proposed in [2] as a natural extension of PCA by constructing the data matrix using stacking lagged versions of input-output variables and applying PCA treating them as separate variables. DPCA requires the order to be known exactly which is seldom known in reality. The general practice is to use heuristics based methods for order determinations, for example [3] and [4]. But, the major issue with DPCA is that it gives biased estimates of model parameters under heteroskedastic error conditions. Therefore, in order to get unbiased estimates of model parameters, the data has to be scaled appropriately using the error-covariance matrix as proposed in [5]. But again, the error-covariance matrix is rarely available in reality and has to be estimated from data. In [6], an iterative procedure called iterative PCA or IPCA is proposed for estimating the error-covariance matrix by solving a maximum likelihood problem in the static case. IPCA also estimates the number of constraints simultaneously. An extension of IPCA to the dynamic case is recently proposed in [7], which estimates the true model order, delay, and the underlying model parameters along with the input-output error-variances under large sample conditions.

In this work, we focus on DIPCA and attempt to study this algorithm in detail, address some of its shortcomings that arise due to small sample conditions, and explore its applications in data reconciliation.

The rest of the paper is organised in the following manner - In Section 2, we formally define our problem and the framework that we will be working under. In section 3, we review some of the algorithms necessary to gain a better understanding of this algorithm, namely static PCA, dynamic PCA, and iterative PCA. In Section 4, we explain the underlying algorithm in DIPCA and motivate it with an example. In Section 5, we highlight the shortcomings of this algorithm and propose techniques subsequently to address them in Section 6. In Section 7, we demonstrate the application of DIPCA in solving a data reconciliation problem, and present our concluding remarks in Section 8.

2. Problem formulation and objectives

Given a linear time-invariant (LTI), single input single output (SISO) system,

$$y^\star[k] + \sum_{i=1}^{n_a} a_i y^\star[k-i] = \sum_{j=D}^{n_b} b_j u^\star[k-j], \tag{1}$$

and input-output data corrupted with white-noise errors as

$$u = u^\star + e_u \quad e_u \sim \mathcal{N}(0, \sigma_{e_u}^2) \tag{2}$$

$$y = y^\star + e_y \quad e_y \sim \mathcal{N}(0, \sigma_{e_y}^2) \tag{3}$$

where $u^*,\ y^*$ are noise free versions of the input and output variables, respectively, the objective is to identify the above dynamic difference equation model by estimating the process order $\eta \triangleq \max(n_a, n_b)$ (n_a is order of the output & n_b is order of the input), delay D, parameters $a_1, a_2, \ldots a_{n_a},\ b_D$
$, b_{D+1}, \ldots b_{n_b}$, and the error variances $\sigma_{e_y}^2$ & $\sigma_{e_u}^2$.

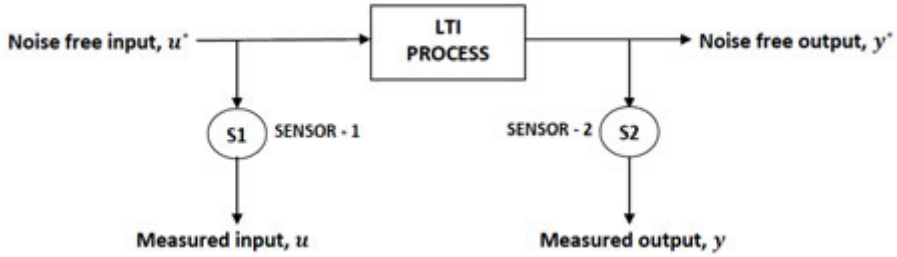

Figure 1: Process flow diagram depicting the data collection process

3. Foundations

The following subsections provide a brief overview of static PCA, dynamic PCA and iterative PCA from the viewpoint of model identification under the TLS framework.

3.1. Steady-state model identification using PCA

Suppose $\mathbf{x}[k] \in \mathbb{R}^{M\times 1}$, at any instant k, are related through d linear constraints as

$$\mathbf{A}\mathbf{x}[k] = \mathbf{0} \qquad A \in \mathbb{R}^{d\times M},\ \text{rank}(\mathbf{A}) = d \tag{4}$$

We can, therefore, say that the constraint matrix, $\mathbf{A}$, lies in the null-space of $\mathbf{X}$, and the model identification problem is equivalent to finding a basis for $\mathbf{A}$. $\mathbf{A}$ can be obtained through eigenvalue decomposition of the sample covariance matrix $\frac{1}{N}\mathbf{X}\mathbf{X}^\mathbf{T}$, of $\mathbf{X}$ where $\mathbf{X} \in \mathbb{R}^{M\times N}$ corresponds to N observations of $\mathbf{x}[k]$. For the noise free case, the last d eigenvalues will be identically 0 and the corresponding eigenvectors provide a basis for the null space of $\mathbf{X}$ i.e.

$$\sigma_{M-d+1} = \sigma_{M-d+2} = \ldots = \sigma_M = 0$$
$$\mathbf{A} = [\mathbf{v}_{M-d+1} \quad \mathbf{v}_{M-d+2} \quad \ldots \quad \mathbf{v}_M]^T$$

For the noisy case, however, the last d eigenvalues does not exactly equal 0 and the extent of deviation from 0 will depend upon the error-variances in the variables. Under such conditions, heuristic based are used in order to determine the value of d. Under homoskedastic error conditions, we have

$$\sigma_{M-d+1} = \sigma_{M-d+2} = \ldots = \sigma_M = \sigma^2_{e_u} = \sigma^2_{e_y}$$
$$\mathbf{A} = [\mathbf{v}_{M-d+1} \quad \mathbf{v}_{M-d+2} \quad \ldots \quad \mathbf{v}_M]^T$$

But, when the error variances are heteroskedastic in nature with $\sigma^2_{e_u} \neq \sigma^2_{e_y}$, PCA gives sub-optimal results. Also, satic PCA is limited only to identifying static relationships between variables and cannot discover dynamic relationships.

3.2. Dynamic model identification using Dynamic PCA

Dynamic PCA or DPCA is an extension of static PCA to the dynamic case which discovers dynamic relationships between a given set of input-output variables. The idea is to stack lagged versions of the variables up to a lag L in order to construct the data matrix $\mathbf{Z}$ as:

$$\mathbf{Z} = \begin{bmatrix} y[k] & y[k-1] & \ldots y[k-L] & u[k] & u[k-1]\ldots & u[k-L] \\ y[k+1] & y[k] & \ldots y[k-L+1] & u[k+1] & u[k]\ldots & u[k-L+1] \\ \vdots & & & & & \vdots \\ y[N-L] & y[N-L+1] & \ldots y[N] & u[N-L] & u[N-L+1]\ldots & u[N] \end{bmatrix}$$

Dynamic relationships between the independent and dependent variables can be obtained by modelling static constraints between these lagged variables.

Now, suppose $z = \begin{bmatrix} z_D & z_I \end{bmatrix}^T$ where, $z_D \in \mathbb{R}^{n_y \times 1}$ corresponds to the dependent variable y, and z_I $\mathbb{R}^{n_u \times 1}$ corresponds to the independent variable, u. Correspondingly, assume that $\mathbf{A} = \begin{bmatrix} \mathbf{A}_D & \mathbf{A}_I \end{bmatrix}$. Then, applying PCA on $\mathbf{Z}$ and considering the eigenvector corresponding to the smallest eigenvalue, the model is given by

$$\mathbf{A}\mathbf{Z}^T = 0 \implies \mathbf{z}_D = \mathbf{B}z_I \qquad \text{where, } \mathbf{B} = -\mathbf{A}_D^{-1}\mathbf{A}_I$$

However, in order to apply DPCA, the order η of the process has to be known a priori as DPCA involves exact stacking with $L = \eta$. Also, DPCA gives optimal parameter estimates only under homoskedastic-error conditions.

3.3. Iterative PCA for error-variance estimation

The underlying idea in IPCA is to perform PCA on a scaled version of original data with a scaling factor of $\mathbf{\Sigma}_e^{-1/2}$, i.e.

$$z[k] :\rightarrow \mathbf{\Sigma}_e^{-1/2} z[k] \tag{5}$$

where, $\mathbf{\Sigma_e}$ is the error-covariance matrix. Scaling the data matrix $\mathbf{Z}$ with $\Sigma_e^{-1/2}$ resolves this issue because it transforms the heteroskedastic-error problem into a homoskedastic-error problem. The homoskedastic-error problem can then be solved easily using dynamic PCA if the system order (η) is known. However, as a result of scaling, eigenvalues of the sample covariance matrix $\mathbf{S_z}$ are shifted by unity, i.e.

$$\lambda(\mathbf{S}_{\mathbf{z}_{scaled}}) = \lambda(\mathbf{S_z}) + 1$$

Hence, instead of zero eigenvalues, the eigenvectors corresponding to unity eigenvalues will give us the linear relations $\hat{\mathbf{A}}$ between variables. Also, the number of such unity eigenvalues gives us the number of constraints, d.

Iterative PCA iteratively estimates the noise covariance matrix $\mathbf{\Sigma}_e$ by solving the following maximum likelihood estimation problem,

$$\min_{\mathbf{\Sigma_e}} \quad N \log |\hat{\mathbf{A}}^{(\mathbf{i})} \mathbf{\Sigma_e} (\hat{\mathbf{A}}^{(\mathbf{i})})^{\mathbf{T}}| + \sum_{i=1}^{N} (\mathbf{r}_i^T[k] (\hat{\mathbf{A}}^{(\mathbf{i})} \mathbf{\Sigma_e} (\hat{\mathbf{A}}^{(\mathbf{i})})^{\mathbf{T}})^{-\mathbf{1}} \mathbf{r}_i[k]) \tag{6}$$

which is based on the assumption that the residuals generated as

$$\mathbf{r} = \hat{\mathbf{A}}\mathbf{Z} \tag{7}$$

are Gaussian-distributed with zero mean and variance $\mathbf{\Sigma_e}$. Therefore, IPCA estimates $\mathbf{A}$ and $\mathbf{\Sigma_e}$ alternatively in an iterative manner given an identifiability constraint is satisfied.

4. Dynamic Iterative PCA

Dynamic Iterative PCA or DIPCA, proposed in [7] is an extension of IPCA for the dynamic case where dynamic relationships between the input and output variables have to be identified. DIPCA is to IPCA what DPCA is to PCA.

4.1. DIPCA algorithm

The idea behind DIPCA is to use IPCA for estimating the error-covariance matrix $\mathbf{\Sigma_e}$ with a modification that $\mathbf{\Sigma_e}$ is a diagonal matrix comprising of σ_u^2 and σ_y^2. The order η is derived from the number of linear relations d as $\eta = L - d + 1$ since stacking in excess results in increased number of constraints. Once we know these quantities, we can then estimate the difference equation model $\mathbf{A}$ using PCA.

4.2. Motivating example for DIPCA

Consider a data generating process,

$$\text{P1:} \quad y^\star[k] + 0.4y^\star[k-1] + 0.6y^\star[k-2] = 1.2u^\star[k-1]$$

Order η, delay η_u, error-covariance matrix $\mathbf{\Sigma_e}$ are all assumed to be unknown and L is arbitrarily chosen to be equal to 5. The error-variances in $\mathbf{y}$ and $\mathbf{u}$ are 0.2477 and 0.1103, respectively. The sample size is $N = 1023$ and the system is excited with a full-band white PRBS signal.

Following DIPCA procedure, we guess a value for the number of constraints d, and check whether the last d eigenvalues are unity. If the last d eigenvalues exactly equal unity, the guessed d equals true number of constraints d^*. If not, we refine our guess of d to a higher or lower value. For the above example, eigenvalues obtained from guessed $d = 4$ gives

$$\Lambda = [28.3 \;\; 26.5 \;\; 15.6 \;\; 14.4 \;\; 11.2 \;\; 10.0 \;\; 9.3 \;\; 2.9 \;\; 1.0458 \;\; 1.0031 \;\; 0.9801 \;\; 0.9710]$$

Therefore, $d^\star = 4$. Order is given by $\eta = L - d^\star + 1 = 5 - 4 + 1 = 2$. Estimated noise variances are $[0.2460 \quad 0.1144]$. Monte-Carlo simulation of 200 runs for non-parametric estimates of confidence intervals yields:

$$y[k] + \underset{\pm(0.0453)}{0.4007}\, y[k-1] + \underset{\pm(0.025)}{0.5997} y[k-2] = \underset{\pm(0.0458)}{0.0012}\, u[k] + \underset{\pm(0.0654)}{1.2025}\, u[k-1] - \underset{\pm(0.0865)}{0.0013}\, u[k-2]$$

It is evident from the above example that DIPCA has correctly identified the process order, delay and the error-variances. Also, the parameter estimates obtained from DIPCA are unbiased.

5. Shortcomings of DIPCA

Although DIPCA in itself is a powerful algorithm that estimates all the unknown quantities, there are certain shortcomings in the current version of DIPCA which we study in the following subsections.

5.1. Ambiguity in order determination for finite sample conditions

For high order systems under finite sample conditions, it is observed that exact unity eigenvalues are not obtained even for a correct guess of the number of constraints d^*. In addition to this, the deviation in unity eigenvalues increases as the order η of the process increases, or the sample size N decreases.

For example, consider a 5^{th} order process with unit delay, excited with a full-band white PRBS input signal as

$$\text{P2:} \quad y^\star[k] + 0.2y^\star[k-1] - 0.6y^\star[k-5] = 1.2u^\star[k-1] + 1.6u^\star[k-2]$$

The error-variances are such that the signal-to-noise ratio (SNR) is maintained at 10. Stacking lag L is arbitrarily chosen to be 15. The true number of linear relations will, therefore, be given

by $d^* = L - \eta + 1 = 15 - 5 + 1 = 11$. However, for a guess of $d = 11$, DIPCA does not give exact unity eigenvalues for the smallest 11 eigenvalues, as depicted in the following plots:

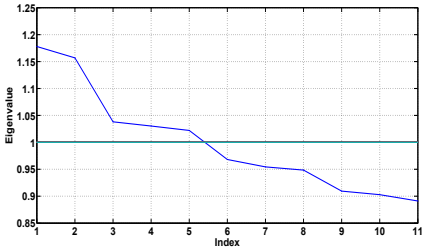

Figure 2: Last $d^* = 11$ λ's for $N = 1023$

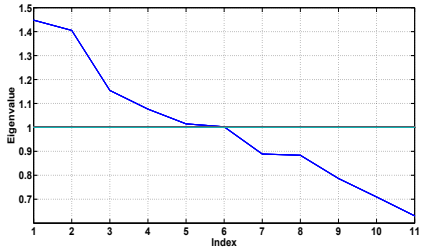

Figure 3: Last $d^* = 11$ λ's for $N = 127$

This deviation from unity leads to ambiguity in order determination since order determination is based on identifying the true number of linear relations d^* by counting the number of unity eigenvalues and then determining the order using the relation $\eta = L - d^* + 1$.

5.2. Effect of stacking in excess and existence of optimal stacking lag

Construction of the data matrix $\mathbf{Z}$ requires stacking lagged version of the input and output variables up to a lag L which is arbitrarily chosen to be large enough to accommodate higher order processes. However, there is a trade-off associated with choosing the value of L - higher L leads to higher redundancy, i.e. the number of equations that go into solving overdetermined set of equations for $\sigma^2_{e_u}$ and $\sigma^2_{e_y}$ increases which leads to a better least squares solution, but at the same time, the effective number of degrees of freedom $N - L - 2$ decreases leading increased variability of estimates.

To demonstrate this trade-off, again consider the process P2. $\mathbf{\Sigma_e}$ is estimated at different stacking lags, L and the mean absolute error (MAE),

$$\text{MAE} = ||\text{diag}(\mathbf{\Sigma_e^0} - \hat{\mathbf{\Sigma}}_\mathbf{e})||_1/2 \tag{8}$$

is computed for different L. For a given L, MAE is averaged over 100 different realizations of error-sequences maintaining the SNR at 10. The following plots for MAE vs. L are obtained:

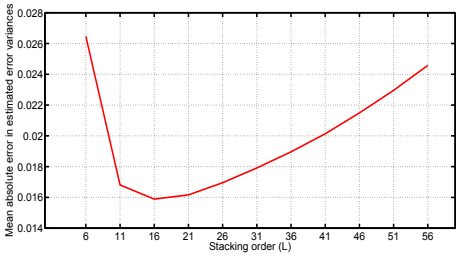

Figure 4: $N = 500$

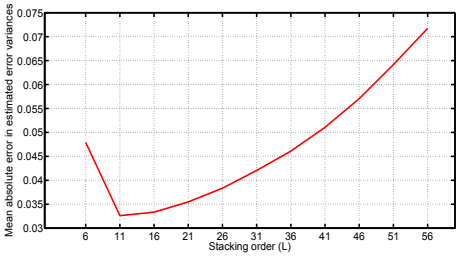

Figure 5: $N = 200$

$\hat{\mathbf{\Sigma}}_\mathbf{e}$ for $N = 500$ and $L = 16$ is $[0.2201 \quad 0.0957]$ when the true values are $[0.2322 \quad 0.0937]$. $\hat{\mathbf{\Sigma}}_\mathbf{e}$ for $N = 200$ and $L = 11$ is $[0.2280 \quad 0.0816]$ when the true values are $[0.2320 \quad 0.0952]$. Note how optimal L changes with the sample size - the optimal L decreases with decrease in N and increases with increase in N.

6. Proposed metric for resolving order ambiguity

The deviations in eigenvalues are observed since the results of IPCA are derived under asymptotic conditions i.e. as $N \to \infty$. As the sample size decreases, this conditions is invalidated and the estimates of eigenvalues are distorted. The distortion happens in both the directions i.e. some of the eigenvalues are over-estimated and some are under-estimated. It is, however, observed that the net deviation in the smallest d eigenvalues equals 0 until the point when $d \leq d^*$. Building upon these properties, we propose a technique to resolve this issue.

6.1. 'd-SEVR' procedure for order determination:

We propose the following systematic procedure to identify the true number of constraints d^* and to subsequently identify the true order η from diverging eigenvalues.

Table 1: Proposed 'd-selective eigenvalue ratio procedure' for order determination

1. For given guessed number of constraints d, calculate the ratio of the maximum (λ^{max}) and minimum (λ^{min}) eigenvalues among the last d eigenvalues, i.e.
$$R_d = \lambda_d^{max}/\lambda_d^{min}$$
2. Calculate R_d for different guesses of d, preferably increasing in steps of 1.
3. The value of d after which there is an abrupt jump (inflection point) in the magnitude of R_d gives us the true number of constraints d^*.
4. The order is given by $\eta^* = L - d^* + 1$.

6.2. Illustrating example

For the same 5^{th} order process P2, we employ the *d-SEVR* procedure to identify d^* for datasets with sample sizes $N = 1023$ (Fig. 6) and $N = 127$ (Fig. 7).

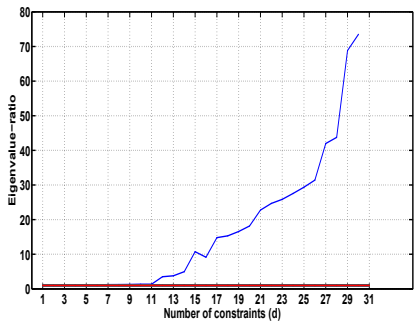

Figure 6: R_d vs. d for $N = 1023$

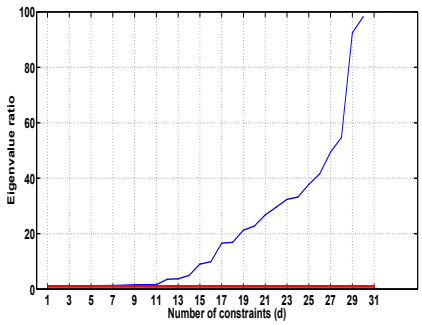

Figure 7: R_d vs. d for $N = 127$

From the above simulation results, it is clear that the proposed metric R_d is robust towards the number of samples N and at the same time is sensitive to an overestimated guess of d.

7. Applications in data reconciliation

A major end goal of any model building exercise is to generate noise free estimates of the noisy data coming out of sensor measurements. In this section, we discuss the application of DIPCA in data reconciliation from a practical viewpoint.

7.1. PCA for dynamic data reconciliation

Since DIPCA is based on the PCA framework, one could consider generating noise-free estimates of the input-output variables using PCA de-noising technique. However, this results in non-unique estimates even for the same variable because in our data matrix,

$$\mathbf{Z} = \begin{bmatrix} y[k] & y[k-1] & u[k-L] \\ y[k+1] & y[k] & u[k-1+1] \\ \vdots & \vdots & \vdots \\ y[N-L] & y[N-L+1] & u[N] \end{bmatrix}$$

we have stacked lagged versions of the same input and output variables. PCA considers all these $2 \times (L+1)$ variables as separate variables independent from each other. As a result of this, we get $L+1$ different reconciled estimates each of the two variables, $\mathbf{y}$ and $\mathbf{u}$.

The following plot corresponding to two different estimates of $\hat{\mathbf{y}}$ for the process P2 depicts this:

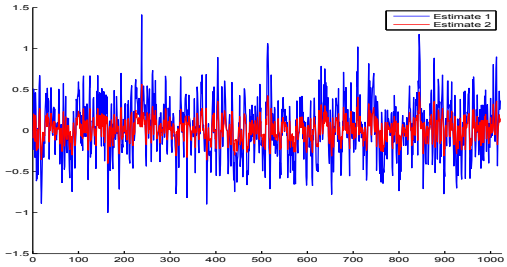

Figure 8: Estimated (filtered) values $\hat{\mathbf{y}}$ from first two columns in $\hat{Z}$

Therefore, we need a more rigorous and technically correct way of generating reconciled values of the input-output variables, and hence, we turn to a Kalman Filter based approach to solve our data reconciliation problem.

7.2. EIV Kalman Filter formulation

In order to get unique, noise free estimates of the input and output variables, we switch to a Kalman Filter based approach since for a linear process, Kalman Filter generates noise free estimates in an optimal manner. However, since the problem setting is that in the errors-in-variables (EIV) framework, we cannot use the usual Kalman Filter as it is. Instead, we use the EIV Kalman Filter proposed in [8] which is formulated as follows:
Consider the state-space form of any process given by

$$\begin{aligned} x[k+1] &= Ax[k] + Bu^*[k] + Gw[k] \\ y^*[k] &= Cx[k] + Du^*[k] \end{aligned} \tag{9}$$

where, $u[k] = u^*[k] + e_1[k]$ and $y[k] = y^*[k] + e_2[k]$ are the input and output variables with measurement errors. In terms of measurements $u[k]$ and $y[k]$, the above model can be re-written as

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] - Be_1[k] + Gw[k] \\ y[k] &= Cx[k] + Du[k] - De_2[k] + e_2[k] \end{aligned} \tag{10}$$

Introducing auxiliary white noise processes,

$$\eta_x[k] = Gw[k] - Be_1[k] \tag{11}$$

$$\eta_y[k] = e_2[k] - De_1[k] \tag{12}$$

the above model can again be re-written as

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] + \eta_x[k] \\ z[k] &= Cx[k] + \eta_y[k] \end{aligned} \tag{13}$$

where,

$$z[k] = y[k] - Du[k] \tag{14}$$

From the equations given above, the update rules for state estimation can be derived by minimizing the trace of the estimated states' error-covariance matrix as given in [8].

7.3. EIV Kalman Filtering for data reconciliation

Table 2: Proposed Methodology

1. Convert the estimated difference equation model into its equivalent state-space form.
2. Estimate unknown states of the state-space model using EIV Kalman Filter update rules.
3. Compute the filtered $\mathbf{y}$ and $\mathbf{u}$ from the estimated states using equations (13) and (14).

7.4. Illustrating example

Consider the data generating process,

$$\text{P3:} \quad y^*[k] + 0.4y^*[k-1] + 0.6y^*[k-2] = 1.2u^*[k-1] + 1.6u^*[k-2]$$

The error variances are such that the SNR is 10 with error variances $\sigma^2_{e_u} = 0.0930$ & $\sigma^2_{e_y} = 0.2336$. Model identified using DIPCA with a stacking lag $L = 5$, averaged over 10 different noise realizations is

$$\bar{\mathbf{A}} = [1 \;\; 0.3955 \;\; 0.5938 \;\; 0.0030 \;\; -1.2035 \;\; -1.5867]$$

or, $$y(k) + 0.3955y(k-1) + 0.5938y(k-1) = 1.2035u(k-1) + 1.5867y(k-2)$$

The estimated error-variances are $\sigma^2_{e_u} = 0.0881$ & $\sigma^2_{e_y} = 0.2436$. Using *tf2ss()* in MATLAB, the equivalent state space representation of the above model is obtained as

$$x[k+1] = \begin{bmatrix} -0.3955 & -0.5938 \\ 1.000 & 0 \end{bmatrix} x[k] + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u[k]$$
$$z[k] = \begin{bmatrix} 1.2035 & 1.5867 \end{bmatrix} x[k]$$

The state estimation error-covariance matrix for this model is $\mathbf{P} = \begin{bmatrix} 0.1060 & -0.1027 \\ -0.1027 & 0.0855 \end{bmatrix}$. The reconciled values for the output variables are obtained and plotted against their measured values below in Figure 9. Also, since the goodness of fit gives us an idea of the amount of de-noising that has happened, Table 3 shows the goodness of fit between the noise free output variable y^* and the measured y & filtered values $\hat{y}$. We can see that there is a 3% increase in the goodness of fit for $\hat{y}$ generated using EIV Kalman Filter.

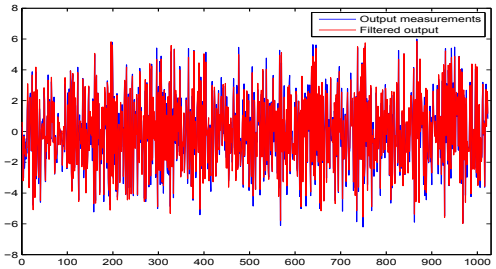

Figure 9: Measurement and reconciled output variables

Table 3: Goodness of fit NMSE values

Goodness of Fit	
Between $\mathbf{y}^*$ and $\mathbf{y}$	0.9348
Between $\mathbf{y}^*$ and $\hat{\mathbf{y}}$	0.9639

8. Conclusions

In this paper, DIPCA algorithm is studied from a practical viewpoint and its shortcomings under small sample conditions are highlighted. A new metric for determining the model order is proposed which, through simulations, is shown to be robust towards the sample size and sensitive to the true model order. Existence of an optimal stacking lag is also shown and its determination from data is the problem that we will address in the future. Kalman Filtering for the errors-in-variables setting is used to generate reconciled values of the input and output variables in an optimal manner. Since multiple input multiple ouput (MIMO) systems also have huge prevalence in industry, extension of DIPCA algorithm for identification of such systems will be our future direction of research.

References

[1] Soderstrom, T. (2007). "Errors-in-variables methods in system identification". *Automatica*, 43(6), 939-958.

[2] Ku, W., Storer, R., and Georgakis, C. (1995). "Disturbance detection and isolation by dynamic principal component analysis". *Chemometrics and Intelligent Laboratory Systems*, 30, 179-196.

[3] Guerfel, M., Othman, K. B., & Benrejeb, M. (2009). "On the structure determination of a dynamic PCA model using sensitivity of fault detection". IFAC Proceedings Volumes, 42(11), 958-963.

[4] Rato, T. J., & Reis, M. S. (2013). "Defining the structure of DPCA models and its impact on process monitoring and prediction activities". Chemometrics and Intelligent Laboratory Systems, 125, 74-86.

[5] Vijaysai, P., Gudi, R., and Lakshminarayanan, S. (2005). "Errors-in-variables-based modeling using augmented principal components". *Industrial & Engineering Chemistry Research*, 44, 368-380.

[6] Narasimhan, S., & Shah, S. L. (2008). "Model identification and error covariance matrix estimation from noisy data using PCA". Control Engineering Practice, 16(1), 146-155.

[7] Maurya, D., Tangirala, A. K., & Narasimhan, S. (2016). "Identification of Linear Dynamic Systems using Dynamic Iterative Principal Component Analysis". IFAC-PapersOnLine, 49(7), 1014-1019.

[8] Diversi, R., Guidorzi, R., & Soverini, U. (2005). "Kalman filtering in extended noise environments". IEEE Transactions on Automatic Control, 50(9), 1396-1402.