Recursive Subspace Identification using an Updating SVD and Recursive Matrix Least Squares

Sujay D. Kadam[1] and Harish J. Palanthandalam-Madapusi[2]

***Abstract*— This paper explores an idea for developing a recursive version of subspace identification techniques to estimate state space matrices *A*, *B*, *C*, and *D* from input and output data using recursive least squares for square systems. The state sequence is computed using a recursive projection and an updating SVD, making the overall identification exercise a recursive one. Results are shown for systems with dynamics changing abruptly as well as for systems that show gradual changes in the eigenvalues, which indicate a potential for online identification. Later, interpretations are provided to indicate the suitability of the suggested algorithm.**

***Index Terms*— Subspace identification, recursive least squares, singular value decomposition, system identification**

I. Introduction

Subspace state space identification methods have been actively developed and used for more than two decades [1], [2], [3], [4], [5], [6], after the pioneering work of de Moor [7]. The subspace methods are a class of system identification methods to identify linear state space models directly from input-output data. The input-output data is used to generate certain subspaces, from which the states and system matrices can be recovered.

The general approach followed in subspace identification is to gather input-output data, formulate Hankel matrices to take projection of their rowspaces and get a projection matrix. This projection matrix after decomposing using SVD can give system related matrices and states. A linear least squares optimization is then used to estimate the system matrices. An immediate advantage of the subspace methods is their applicability to MIMO systems. Also, only a single parameter, the model order is to be usually specified, which can allow model order reduction as well.

One challenge in subspace methods is that, if the dataset is large, operations like projections (oblique and orthogonal) and singular value decomposition become computationally heavy. This limits the applicability of subspace methods in real-time applications. A solution to this is to recursify the entire subspace identification process. Recently, several methods for recursive subspace identification have been proposed recently [8], [9], [6].

Another impediment in using the subspace methods is in using *a priori* information about the model under study. If the states (basis) or the sparsity (pattern) of the system matrices is known then it could be incorporated into the estimation process. However, this structured subspace identification or gray box modeling in the context of subspace methods is usually difficult.

In literature, recursive identification of systems in the subspaces framework has often been discussed. [10], [11] show how subspace identification can be seen as a optimal multi-step prediction. Further, [11] suggests the recursification of the oblique projection operation, that leads to the projection matrix. Several propogator based MOESP methods for recursive identification that make use of an observation vector and update the extended observability matrix online have been proposed in [4]. Output only recursive subspace identification is presented in [12]. Recursive subspace identification method that uses RLS formulation of the multi-step predictor as described in [10], [11] to find Markov parameter matrices, leading to the use of realization algorithm for finding systems matrices, has been discussed in [9]. [13] has suggested a recursive subspace scheme for systems with non uniform sampling.

Here, we suggest a recursive identification scheme for square systems (same number of inputs and outputs), that uses two recursive least squares steps and an updating SVD step in between those. The first RLS is used to get an estimate of the Markov parameter matrix H_i and a matrix L_w that can give the projected data matrix, when used with past input-output data. Using the updating SVD, Markov parameter matrix can be used to get the extended observability and controllability matrices. The extended observability matrix and the projected data matrix give the estimate of the state sequence. Once state estimate is available, the second RLS step is used to get the estimates of the system matrices. The main contribution of the paper is in the way states are calculated from the input-output data from sliding windows, using recursive projections and the updating SVD.

The paper is arranged in the following manner. Section III presents the problem statement. Later, in Section IV description of the suggested recursive scheme for subspace state space identification is provided in terms of the approach, techniques used and the summary. Results and interpretations of the suggested recursive scheme are presented in Section V. Conclusions and the scope of future work are put forth in Section VI.

II. Brief Overview of Subspace Identification

Subspace identification methods are used to compute system matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{l\times n}$, $D \in \mathbb{R}^{l\times p}$ and order n of state space models of the type,

$$x_{k+1} = Ax_k + Bu_k + w_k \quad (1)$$

[1] Sujay D. Kadam (kadam.sujay@iitgn.ac.in) is a Graduate Student at SysIDEA Lab, IIT Gandhinagar, India.

[2] Harish J. Palanthandalam-Madapusi (harish@iitgn.ac.in) is Assistant Professor in Mechanical Engineering at IIT Gandhinagar, India.

Preprint submitted to 2016 American Control Conference.
Received September 28, 2015.

$$y_k \; = \; Cx_k + Du_k + v_k \tag{2}$$

by using input-output data of s samples or measurements. Also in (1) and (2), $u \in \mathbb{R}^p$, $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^l$ denote inputs, states and outputs respectively. The states and measurements are assumed to be corrupted by process noise w_k and measurement noise v_k.

We recall here, the procedure generally followed in subspace identification of deterministic-stochastic systems as described in [5]. The first step in subspace state space identification (also called as 4SID) methods is the formation of block Hankel matrices given by, $Y = \begin{bmatrix} Y_p^T & Y_f^T \end{bmatrix}^T$ and $U = \begin{bmatrix} U_p^T & U_f^T \end{bmatrix}^T$. and,

$$U_p \triangleq \begin{bmatrix} u_0 & u_1 & \cdots & u_{j-1} \\ u_1 & u_2 & \cdots & u_j \\ \vdots & \vdots & \ldots & \vdots \\ u_{i-1} & u_i & \cdots & u_{i+j-2} \end{bmatrix};$$

$$U_f \triangleq \begin{bmatrix} u_i & u_{i+1} & \cdots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & \cdots & u_{i+j} \\ \vdots & \vdots & \ldots & \vdots \\ u_{2i-1} & u_{2i} & \cdots & u_{2i+j-2} \end{bmatrix} \tag{3}$$

and

$$Y_p \triangleq \begin{bmatrix} y_0 & y_1 & \cdots & y_{j-1} \\ y_1 & y_2 & \cdots & y_j \\ \vdots & \vdots & \ldots & \vdots \\ y_{i-1} & y_i & \cdots & y_{i+j-2} \end{bmatrix};$$

$$Y_f \triangleq \begin{bmatrix} y_i & u_{i+1} & \cdots & y_{i+j-1} \\ y_{i+1} & y_{i+2} & \cdots & y_{i+j} \\ \vdots & \vdots & \ldots & \vdots \\ y_{2i-1} & y_{2i} & \cdots & y_{2i+j-2} \end{bmatrix} \tag{4}$$

Subscripts p and f denote *'past'* and *'future'* data respectively. And j is a very large integer, such that $j \to \infty$ when dataset is assumed to be infinitely long. In practical cases, $j = s - 2i + 1$ and i is an integer indicating a guess of the maximum possible order of the system under identification.

The row spaces and column spaces of the Hankel matrices so formed contain the information of the system related matrices like the extended observability matrix (Γ_i) and the state sequence (X_i) which can be used for estimating the system matrices. Oblique or orthogonal projections can be used to find the projection data matrix $O_i = \Gamma_i X_i$ and this matrix will be same for any state space basis chosen. The projection operations are defined as follows. If α, β and γ are matrices, then α/β denotes orthogonal projection of row space of α on the row space of β and $\alpha/_\gamma\beta$ denotes oblique projection of row space of α on the row space of β in the direction of row space of γ. Also,

$$\alpha/\beta = \alpha\beta^T(\beta\beta^T)^{-1}\beta \tag{5}$$

and

$$\alpha/_\gamma\beta = [\alpha/\beta][\alpha/\beta]^\dagger\gamma \tag{6}$$

where, † indicates the Moore-Penrose pseudo-inverse.

The projection $Y_f/_{U_f}[W_p]$ yields the projection data matrix O_i, where, $W_p \triangleq \begin{bmatrix} Y_p \\ U_p \end{bmatrix}$. The O_i matrix so obtained, is pre-multiplied and post -multiplied by W_1 and W_2 weighting matrices respectively and then decomposed using SVD as, $W_1 O_i W_2 = M\Sigma N^T$. The SVD is partitioned to keep only the first n significant singular values and corresponding columns and rows of M and N^T as $\Sigma = \texttt{diag}(\Sigma_1, \Sigma_2)$, $M = [M_1 \; M_2]$ and $N^T = [N_1^T \; N_2^T]$ This is where order n can be determined. Γ_i and X_i can then be calculated using Σ_1, M_1 and N_1^T. Once these are obtained, estimates of A, B, C and D can be obtained by various algorithms/techniques presented in [5] that make use of decomposition of Γ_i and a least squares formulation using the state sequence X_i to determine the system matrices. Also, the choice of weighting matrices W_1 and W_2 determine the basis of the identified system. An interested reader is referred to [14] for more details on the choice of weighting matrices.

III. Problem Statement

Consider a system represented in an innovations form,

$$x_{k+1} \; = \; Ax_k + Bu_k + Ke_k \tag{7}$$

$$y_k \; = \; Cx_k + Du_k + e_k \tag{8}$$

where, e_k is an unknown innovation in the output, with covariance given by $R_e = \mathbb{E}[e_k e_k^T]$ and K is the steady state Kalman filter gain that brings innovations in states, corresponding to e_k. If number of available input-output data is assumed to be infinitely long, then K is steady state. If data available is limited, K is a non-steady state filter as described in [5]. (Note that $\mathbb{E}[\bullet]$ denotes an expectation operator.) Also, as per [3], the state space process form given by (1), (2) and the innovations form given by (7), (8) are equivalent as the represent the input-output data exactly.

If the recursive equations in (7) and (8) are converted into a non recursive formulation as in [5],

$$\begin{aligned} Y_p \; &= \; \Gamma_i X_p \; + \; H_i U_p \; + \; H_i^s E_p \\ Y_f \; &= \; \Gamma_i X_f \; + \; H_i U_f \; + \; H_i^s E_f \end{aligned} \tag{9}$$

where,

$$\Gamma_i \triangleq \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{bmatrix}; \; H_i \triangleq \begin{bmatrix} D & 0 & \cdots & 0 & 0 \\ CB & D & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{i-2}B & CA^{i-3}B & \cdots & D & 0 \\ CA^{i-1}B & CA^{i-2}B & \cdots & CB & D \end{bmatrix} \tag{10}$$

and,

$$H_i^s \triangleq \begin{bmatrix} I & 0 & \cdots & 0 & 0 \\ CK & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{i-2}K & CA^{i-3}K & \cdots & I & 0 \\ CA^{i-1}K & CA^{i-2}K & \cdots & CK & I \end{bmatrix} \tag{11}$$

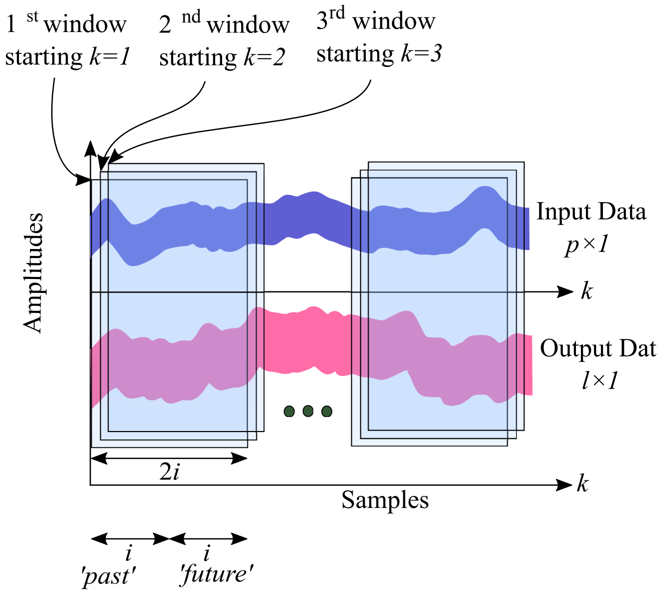

Fig. 1. Collection of data into sliding windows of size $2i$.

Γ_i is the extended observability matrix, and H_i and H_i^s are the Markov parameter matrices for the u_k and e_k inputs respectively. E_p and E_f are defined analogously as Y_p and Y_f. The problem considered here is to identify the parameters A, B, C and D recursively using subspace identification paradigms on online input-output data. Such an subspace state space identification scheme would be beneficial in cases where, the input-output data available are too large to be processed in batch. It will also be helpful for adaptive control algorithms which rely on the current estimate of the system matrices.

IV. Description of the Identification Scheme

A. Approach towards solving the identification problem

The idea suggested to identify the parameter matrices is as follows. Three major operations in the subspace state space identification methods are projections (oblique or orthogonal), singular value decomposition and least squares solution. The identification process therefore can be made recursive if the aforementioned operations are recursified. Instead of using the block Hankel matrices formed from the complete data to project subspaces on one another, updates to the projection matrices can be made through the data that becomes available every sample [11]. Decomposition of the projection matrix by recursive or updating SVD yields system related matrices like extended observability and controllability matrices and the state sequence. Once the unbiased estimate of the state sequence is available for the given sampling instant, equation (12) can be solved in a recursive least squares sense, thus making overall identification process a recursive one.

$$\begin{bmatrix} \hat{x}_{k+1} \\ y_k \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ u_k \end{bmatrix} \tag{12}$$

B. Recursive Projections

Refering to Fig. (1), assume that input-output data of respective dimensions are available at every sampling time instant from a system supposedly of the form given in (1) and (2). The data are collected in a moving sample window of size $2i$ for input as well as output. The window sizes for input and output respectively are $2i \times p$ and $2i \times l$. Here, i is an integer that would be a guess of maximum possible order of the system under identification. Incidentally, these windows form the columns of what would be block Hankel matrices of the complete data.

The equation,

$$\hat{Y}_f \; = \; \Gamma_i X_f \; + \; H_i U_f \tag{13}$$

can be seen as a predictor of Y_f [11]. But, X_f is unknown as the states are not known. Therefore, $\Gamma_i X_f$ is substituted by a term $L_w W_p$ that makes use of input-output data in lieu of X_f to achieve the same effect. $L_w \triangleq [(\Gamma_i A^i \Gamma_i^\dagger) \;\; \Gamma_i(\Delta_i \; - \; A^i \Gamma_i^\dagger H_i)]$ [10], [9], and pages 41 and 110 of [5].

The equation (13) can be rewritten as,

$$\left[\hat{Y}_f\right]_{li \times j} \; = \; \left[\left[L_w\right]_{li \times 2pi} \quad \left[H_i\right]_{li \times li} \right] \begin{bmatrix} W_p \\ U_f \end{bmatrix}_{(2p+l)i \times j} \tag{14}$$

The equation (14) can be seen as a least squares problem and can be solved by using recursive least squares formulation [11]. In the least squares formulation, data from sliding windows i.e. from successive columns of the block Hankel matrices are used to update the parameter $\Theta_{\text{pr}}, k = \begin{bmatrix} L_w & H_i \end{bmatrix}$. Getting the updates of this parameter is analogous to obtaining the projected data matrix in subspace based methods. The recursive estimate of Θ_{pr}, k can be computed at every sampling instant by,

$$\underbrace{\hat{\Theta}_{\text{pr},k}}_{New\ estimate} \; = \; \underbrace{\hat{\Theta}_{\text{pr},k-1}}_{Old\ estimate} + \underbrace{\left(\mathbf{y}_k - \hat{\Theta}_{\text{pr},k-1}\mathbf{d}_k\right)}_{Correction} \underbrace{\mathbf{d}_k^T P_{\text{pr},k}}_{Gain} \tag{15}$$

where,

$$\mathbf{y}_k \; = \; \begin{bmatrix} \mathbf{y}_{p,k} \\ \mathbf{y}_{f,k} \end{bmatrix}; \; \mathbf{d}_k \; = \; \begin{bmatrix} \mathbf{y}_{p,k} \\ \mathbf{u}_{p,k} \\ \mathbf{u}_{f,k} \end{bmatrix} \tag{16}$$

and

$$P_{\text{pr},k} = P_{\text{pr},k-1} - P_{\text{pr},k-1}\mathbf{d}_k \left(I + \mathbf{d}_k^T P_{\text{pr},k-1}\mathbf{d}_k\right)^{-1} \mathbf{d}_k^T P_{\text{pr},k-1} \tag{17}$$

Note that, $k > 2i$. The recursive estimation computation can be commenced once the data of a complete sliding window are available. Vectors $\mathbf{u} \in \mathbb{R}^{2i \times p}$ and $\mathbf{y} \in \mathbb{R}^{2i \times l}$ are the columns at the sampling instant k of the input and output block Hankel matrices respectively, those would exist if entire data were available. And, the subscripts p and f indicate the *'past'* and *'future'* samples of the sliding windows for input-output data. $P_{\text{pr},k}$ is the covariance of the estimation error. It is important to note that the concatenated matrix $\mathbf{d}_k$ in (16) will be valid only for $p = l$. i.e for square systems only.

Let, $\mathbf{w}_{p,k} \triangleq \begin{bmatrix} \mathbf{y}_{p,k} \\ \mathbf{u}_{p,k} \end{bmatrix}$, then $L_w \mathbf{w}_{p,k} = \Gamma_i x_{k-i+1}$. This means, as $\mathbf{w}_{p,k}$ is a single column vector, it can provide information about the state only at one sampling instant $k-i+1$. An i step ahead prediction of the state x starting from $k-i+1$ can be obtained as described here-forth. A block Hankel matrix $\bar{\mathbf{w}}_{p,k}$ given by,

$$\bar{\mathbf{w}}_{p,k} = \begin{bmatrix} \bar{\mathbf{y}}_{p,k} \\ \bar{\mathbf{u}}_{p,k} \end{bmatrix} \tag{18}$$

is constructed by concatenating two block Hankel matrices $\bar{\mathbf{u}}_{p,k}$ and $\bar{\mathbf{y}}_{p,k}$ formed by arranging into block Hankel structure, the input-output data from a single sliding window. Therefore,

$$\begin{aligned} L_w \bar{\mathbf{w}}_{p,k} &= \Gamma_i \bar{\mathbf{x}}_{f,k} \\ \therefore \bar{\mathbf{x}}_{f,k} &= \Gamma_i^{\dagger} L_w \bar{\mathbf{w}}_{p,k} \end{aligned} \tag{19}$$

Where, $\bar{\mathbf{x}}_{f,k}$, the state sequence corresponding to the block Hankel matrices $\bar{\mathbf{y}}_{f,k}$ and $\bar{\mathbf{u}}_{f,k}$. The first part of (19) indicates that $L_w \bar{\mathbf{w}}_{p,k}$ (the *'past'* inputs and outputs) will have a same effect as that of $\Gamma_i \bar{\mathbf{x}}_{f,k}$ (*'future'* states). In analogy to second part of the equation (9), we may write,

$$\bar{\mathbf{y}}_{f,k} = L_w \bar{\mathbf{w}}_{p,k} + H_i \bar{\mathbf{u}}_{f,k} + H_i^s \bar{\mathbf{e}}_{f,k} \tag{20}$$

where, $\bar{\mathbf{e}}_{f,k}$ is defined analogously to $\bar{\mathbf{u}}_{f,k}$ and $\bar{\mathbf{y}}_{f,k}$ and contains white residuals. From equations (19) and (20), we may say the following,

$$\hat{\mathbf{x}}_{f,k} = \Gamma_i^{\dagger} \bar{\mathbf{y}}_{f,k} = \bar{\mathbf{x}}_{f,k} + \Gamma_i^{\dagger} H_i \bar{\mathbf{u}}_{f,k} + \Gamma_i^{\dagger} H_i^s \bar{\mathbf{e}}_{f,k} \tag{21}$$

$\hat{\mathbf{x}}_{f,k}$ is an unbiased estimate of $\bar{\mathbf{x}}_{f,k}$, the state sequence of length i starting from sampling time index $k-i+1$. Let $\kappa = k-i+1$. Then, $\hat{x}_{\kappa}$ and $\hat{x}_{\kappa+1}$ from the estimated state sequence $\hat{\mathbf{x}}_{f,k}$, can be used in (12) along with y and u having appropriate sampling time indices, to recursively update the parameter matrices A, B, C and D.

On similar lines, *'past'* state sequence can be computed as,

$$\hat{\mathbf{x}}_{p,k} = \Gamma_i^{\dagger} \bar{\mathbf{y}}_{p,k} = \bar{\mathbf{x}}_{p,k} + \Gamma_i^{\dagger} H_i \bar{\mathbf{u}}_{p,k} + \Gamma_i^{\dagger} H_i^s \bar{\mathbf{e}}_{p,k} \tag{22}$$

The index that is to be used with states $\hat{x}_{\kappa}$ and $\hat{x}_{\kappa+1}$ from the *'past'* is $\kappa = k-2i+1$.

C. Recursive matrix least squares for parameters A, B, C, and D

The recursive least squares solution for solving the least squares problem can be obtained by,

$$\begin{aligned} \underbrace{\hat{\Theta}_{\kappa}}_{\textit{New estimate}} &= \underbrace{\hat{\Theta}_{\kappa-1}}_{\textit{Old estimate}} + \underbrace{\left(\mathscr{Y}_{\kappa}^T - \hat{\Theta}_{\kappa-1}\Phi_{\kappa}^T\right)}_{\textit{Correction}} \underbrace{\Phi_{\kappa} P_{\kappa}}_{\textit{Gain}} \\ P_{\kappa} &= P_{\kappa-1} - P_{\kappa-1}\Phi_{\kappa}^T \left(I + \Phi_{\kappa} P_{\kappa-1} \Phi_{\kappa}^T\right)^{-1} \Phi_{\kappa} P_{\kappa-1} \end{aligned} \tag{23}$$

where, $\Theta = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ $\mathscr{Y}_{\kappa} = \begin{bmatrix} x_{\kappa+1} \\ y_{\kappa} \end{bmatrix}$ and $\Phi_{\kappa} = \begin{bmatrix} x_{\kappa} \\ u_{\kappa} \end{bmatrix}$

D. SVD update to obtain Γ_i

In the equation (21), in order to compute the estimate of the state sequence, a term $\Gamma_i^{\dagger}$ is required. This term can be computed by performing an updating SVD of the H_i Markov parameter term as defined in (10). The term H_i can be split as $H_i = \Gamma_i \Delta_i$ as suggested in [7], [11]. Where, $\Delta_i \triangleq \begin{bmatrix} B & AB & A^2B & \cdots & A^{i-1}B \end{bmatrix}$. It must be noted that H_i contains D terms whereas Γ_i and Δ_i do not. Besides, H_i is lower triangular and therefore a sparse matrix. The product $\Gamma_i \Delta_i$, however, may not be sparse. Hence, a new matrix $\bar{H}_i$ which is void of any D terms and symmetric is defined as,

$$\bar{H}_i \triangleq \begin{bmatrix} CB & CAB & \cdots & CA^{i-2}B & CA^{i-1}B \\ CAB & CB & \cdots & CA^{i-3}B & CA^{i-2}B \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{i-2}B & CA^{i-3}B & \cdots & CB & CAB \\ CA^{i-1}B & CA^{i-2}B & \cdots & CAB & CB \end{bmatrix} \tag{24}$$

As H_i is computed every sampling instant, $\bar{H}_i$ can also be computed every time instant. $\bar{H}_i$ can be decomposed using an updating SVD as suggested in [15]. Therefore, $\bar{H}_i = USV^T$ is available every sampling instant and $\Gamma_i = US^{\frac{1}{2}}$ [7] can be used in finding the unbiased estimates of the states using (21).

The updating algorithm suggested in [15], however, may return S matrix with negative diagonal elements. As a correction to this, the following changes are suggested: $U = Q_B \times \texttt{diag}(\texttt{sign}(\texttt{diag}(R_B)))$, and $S = R_B \times \texttt{diag}(\texttt{sign}(\texttt{diag}(R_B)))$. ($Q_B$ and R_B are matrices as suggested in [15]).

It is important to note that, since $\bar{H}_i$ is constructed from H_i matrix without the D blocks, Γ_{i-1} is obtained instead of Γ_i. Hence, while constructing the matrix given in (18), this fact must be taken into account.

E. Summary of the Algorithm

1) Start gathering input-output data and form sliding windows of length $2i$ for both input and output. First sliding windows for input and output data are obtained for sampling instants $k > 2i$. Thereafter, for every sampling instant new sliding windows are available.
2) Sliding windows so obtained after $k > 2i$ sampling instants can be used to find parameter $\Theta_{\mathrm{pr}}, k = \begin{bmatrix} L_w & H_i \end{bmatrix}$ using (15).
3) Perform an updating SVD on $\bar{H}_i$ to get Γ_i.
4) Obtain an estimate of the state sequence using (21) or (22).
5) Using appropriate indices κ, update the system matrices A, B, C and D as indicated in (23).

V. Results

A. Numerical Example

Assume that input-output data from a continuous time system with transfer function given by,

$$G(s) = \frac{0.5}{(s+2)(s+0.2 \pm 0.5j)} \tag{25}$$

is sampled at a rate of 1 Hz, and used for the identification exercise. It is assumed that the input and output are both corrupted by white-gaussian noises $\mathscr{N}(0, 0.01)$. The input is a combination of pseudo-random binary signal and a random gaussian signal to have a persistent excitation. The samples arriving every time instant are used in the algorithm suggested in the previous section. An assumption is made

Preprint submitted to 2016 American Control Conference.
Received September 28, 2015.

that the order of the system $n = 3$ is known. i is chosen to be 5. After using $k = 1000$ samples of data, the system identification results are obtained as shown in Fig. (2), for one realization of the estimates of A, B, C and D. If the identification exercise is repeated for numerous times, it creates an ensemble of estimates. Here, result of some realization from the ensemble of the recursive identification exercises is presented.

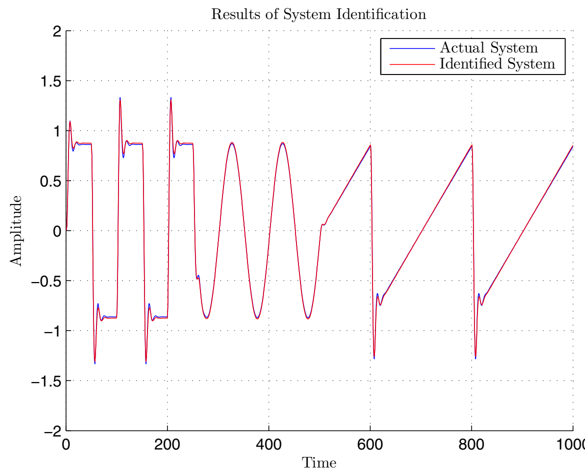

Fig. 2. Responses of actual and identified systems to square, sine and sawtooth inputs

The comparison of impulse responses for actual and identified system in the same realization is shown in Fig. (3). It can be seen that the impulse responses closely follow each other. It is also suggestive of the fact that Markov parameter matrix H of the actual system and its the estimate H_i calculated in an updating manner using (15) are in good accordance. In fact, the second RLS step identifies the system matrices such that they conform to the impulse response given by the Markov parameter matrix H_i.

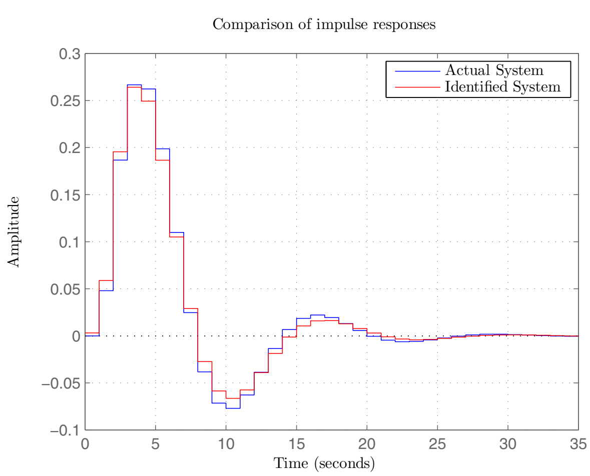

Fig. 3. Impulse responses for the actual and identified system

An ensemble of identified parameters is generated by performing the identification exercise for a 100 realizations. The eigenvalues of the system matrices from these realizations are plotted as shown in Fig. (4). Even though the actual and identified eigenvalues may seem very different in the eigenvalue plot for different realizations, the input-output responses are close enough.

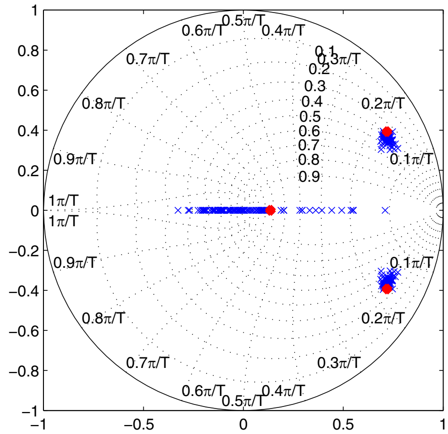

Fig. 4. Plot of eigenvalues of the identified system matrix for 100 realizations. Red indicates the actual system's eigenvalues and blue are the identified system's eigenvalues.

If a certain system undergoes a sudden change, its eigenvalues also can be seen to relocate suddenly. This single shot change in the eigenvalues can be detected by the suggested algorithm, when forgetting factors are used in the recursive least squares steps. Fig. (5) shows the plots for the system dynamics changing as a single shot event and the responses of the suggested algorithm.

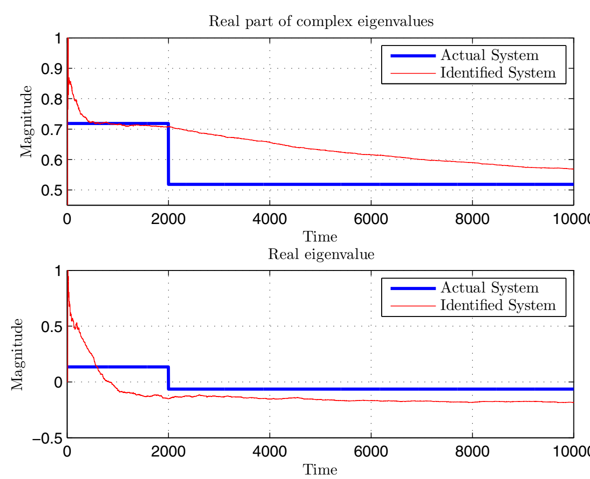

Fig. 5. Plot of eigenvalues for a single shot change in the system dynamics. Forgetting factor is set to 0.7.

For systems with slowly changing dynamics, when forgetting factors are used, the algorithm gives results as shown in Fig. (6).

Preprint submitted to 2016 American Control Conference.
Received September 28, 2015.

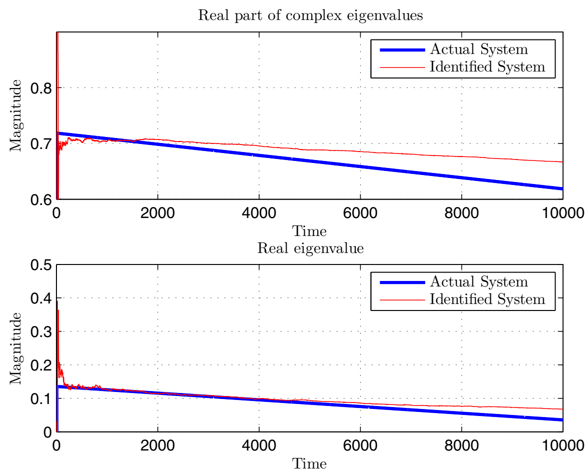

Fig. 6. Plot of eigenvalues for a gradual shift in the system eigenvalues. Forgetting factor is set to 0.7.

B. Interpretations

The algorithm uses two recursive least squares operations, one for the projection update and the other for recursive matrix least squares that finds the system matrices. Besides, an updating SVD step takes place in between the RLS operations. The updating SVD factorizes the H_i matrix to get the extended observability and controllability matrices. This means that the extended observability matrix is updated with every new data received. This further ensures that the basis of the identified system remains fixed. Also, the second RLS step ensures that the estimates of the system matrices conform to the impulse response given by the Markov parameter matrix.The algorithm so presented in the previous section is still a black box identification method as the basis chosen for each realization by the updating SVD operation may not be the same. Thus, the algorithm is recursive/updating in a sense that system parameter matrix updates are obtained at every sampling instant. The suggested algorithm to perform well, relies heavily on the unbiased estimates of the states. If the estimates of the states are poor, the algorithm will not perform well. Also, if there is a time delay in the input-output data, the algorithm will not provide expected results. Inherent property of the subspace state space identification methods is that the are applicable for MIMO systems without much modification. This method is no exception, however with the constraint that it is applicable only for square systems. This is due to definition of $\mathbf{d}_k$ in (16). The sliding windows to be concatenated must have agreeable dimensions, which is possible only in square cases.

VI. Conclusion

The paper has put forth an idea of completely recursifying the subspace state space identification algorithm, by the use of state estimates and the input and output data. States are estimated by recursive projections and updating SVD methods existing in literature. The knowledge of the system's order n is required, however. Changes in the updating SVD method have been suggested to ensure that the singular values are positive/non-negative. The proposed scheme suggests the use of SVD instead of circumventing it, as many other recursive methods suggest. Parameter matrices are then updated using a recursive formulation of a matrix least squares problem given in (23). This is different from using Markov parameters to estimate system matrices using various realization algorithms. The suggested method is favorable over realization algorithm based methods as those do not have a scope to apply corrections once Markov parameter matrices are found. The suggested method uses a correction mechanism in terms of a RLS step that computes the system matrices. Markov parameters are however used to find extended observability matrix. Results using forgetting factors for RLS steps indicate that the method has a potential for a truly online subspace identification that could be applicable for time varying systems or systems that change by a single shot event.

References

[1] M. Verhaegen, "Identification of the deterministic part of mimo state space models given in innovations form from input-output data," *Automatica*, vol. 30, no. 1, pp. 61–74, 1994.

[2] W. Favoreel, B. De Moor, and P. Van Overschee, "Subspace state space system identification for industrial processes," *Journal of process control*, vol. 10, no. 2, pp. 149–155, 2000.

[3] S. J. Qin, "An overview of subspace identification," *Computers & chemical engineering*, vol. 30, no. 10, pp. 1502–1513, 2006.

[4] G. Mercère, L. Bako, and S. Lecœuche, "Propagator-based methods for recursive subspace model identification," *Signal Processing*, vol. 88, no. 3, pp. 468–491, 2008.

[5] P. van Overschee and B. de Moor, *Subspace Identification for Linear Systems: Theory - Implementation - Applications*. Kluwer Academic Publishers, 2012.

[6] L. Shang, J. Liu, K. Turksoy, Q. M. Shao, and A. Cinar, "Stable recursive canonical variate state space modeling for time-varying processes," *Control Engineering Practice*, vol. 36, pp. 113–119, 2015.

[7] B. de Moor, *Mathematical Concepts and Techniques for Modelling of Static and Dynamic Systems*. PhD thesis, Katholieke Univerisiet, Leuven, 1988.

[8] L. Zhuang, F. Pan, and F. Ding, "Parameter and state estimation algorithm for single-input single-output linear systems using the canonical state space models," *Applied Mathematical Modelling*, vol. 36, no. 8, pp. 3454 – 3463, 2012.

[9] A. Alenany and H. Shang, "Recursive subspace identification with prior information using the constrained least squares approach," *Computers & Chemical Engineering*, vol. 54, pp. 174 – 180, 2013.

[10] P. Trnka and V. Havlena, "Subspace identification as multi-step predictions optimization," in *Proceedings of the Fifth IASTED International Conference on Modelling, Simulation and Optimization*, pp. 223–228, 2005.

[11] P. Trnka and V. Havlena, "Recursive subspace identification in the least squares framework," in *7th International Scientific-Technical Conference-Process Control 2006*, pp. R068C1–R068C9, 2006.

[12] I. Goethals, L. Mevel, A. Benveniste, and B. De Moor, "Recursive output-only subspace identification for in-flight flutter monitoring," in *Proceedings of the 22nd International Modal Analysis Conference (IMACXXII), Dearborn, Michigan*, 2004.

[13] H. Wang and T. Liu, "Recursive state-space model identification of non-uniformly sampled systems using singular value decomposition," *Chinese Journal of Chemical Engineering*, vol. 22, no. 11, pp. 1268–1273, 2014.

[14] P. Van Overschee and B. De Moor, "A unifying theorem for three subspace system identification algorithms," *Automatica*, vol. 31, no. 12, pp. 1853–1864, 1995.

[15] R. Badeau, G. Richard, and B. David, "Sliding window adaptive SVD algorithms," *Signal Processing, IEEE Transactions on*, vol. 52, no. 1, pp. 1–10, 2004.

Preprint submitted to 2016 American Control Conference.
Received September 28, 2015.