Maximum-Likelihood Parameter Estimation of Bilinear Systems

Stuart Gibson, Adrian Wills, and Brett Ninness

***Abstract*—This paper addresses the problem of estimating the parameters in a multivariable bilinear model on the basis of observed input-output data. The main contribution is to develop, analyze, and empirically study new techniques for computing a maximum-likelihood based solution. In particular, the emphasis here is on developing practical methods that are illustrated to be numerically reliable, robust to choice of initialization point, and numerically efficient in terms of how computation and memory requirements scale relative to problem size. This results in new methods that can be reliably deployed on systems of nontrivial state, input and output dimension. Underlying these developments is a new approach (in this context) of employing the expectation-maximization method as a means for robust and gradient free computation of the maximum-likelihood solution.**

***Index Terms*—Bilinear systems, maximum likelihood (ML), parameter estimation, system identification.**

I. Introduction

BILINEAR systems are nonlinear descriptions which are distinguished by the fact that exogenous inputs may enter in a manner which is multiplicative with the system state. The importance and utility of these sort of models is now well accepted with a history, at least within the control community, spanning more than three decades [6].

This is due in part to their relative simplicity within the broad class of nonlinear systems. However, it also arises via their ability to characterize a very wide range of chemical, biological, robotic and manufacturing processes for which any linear approximation is very far from satisfactory [11], [47].

For example, as explained in [11], for chemical processes it is common that exogenous inputs are flow rates. Natural choices of system state, such as temperature or concentration then evolve and affect the process output in a manner which is multiplicative with the input, according to mass and heat balance considerations.

Additionally, bilinear system models are also useful as approximators, or alternate representations for a range of other nonlinear system descriptions [25]. For instance, any discrete-time finite Volterra series expansion with time-invariant separable kernels can be realized compactly as a discrete-time bilinear system [37].

Manuscript received June 1, 2004; revised May 25, 2005. Recommended by Guest Editor A. Vicino. This work was supported by the Australian Research Council in the case of all the authors involved.

S. Gibson is with the Lehman Brothers, London WE1, U.K. (e-mail: stuart.gibson@lehman.com).

A. Wills and B. Ninness are with the School of Electrical Engineering and Computer Science, The University of Newcastle, NSW 2308, Australia (e-mail: onyx@ee.newcastle.edu.au; brett@ee.newcastle.edu.au).

Digital Object Identifier 10.1109/TAC.2005.856664

Motivated by their importance and utility, this paper studies the problem of estimating the parameters of bilinear descriptions on the basis of observed input-output data. Given the significance of this issue, it has an extensive history of previous study, which can be briefly surveyed by division into themes.

In [1], [21], and [43] and related work cited in those papers, the input is assumed to be a stationary time series with known properties (sometimes known densities), and second or higher order moments of certain signals are computed and then employed to find estimates via a correlation analysis, or via a stochastic approximation approach. This involves a Volterra kernel description of the bilinear system, which can imply very high dimension quantities, with attendant computation difficulties for systems of appreciable state or input–output dimension.

In [12], [13], [45], and [46] the use of a state–space formulation of the bilinear models, together with gradient based search for a maximum-likelihood (ML) solution is studied. In particular [12], [13] consider single-input–single-output (SISO) systems together with a canonical parametrization of the system matrices, while [45], [46] address the multiple-input–multiple-output (MIMO) case via a combination of full parametrization of system matrices coupled with a gradient search strategy that ignores associated directions of rank deficiency.

The work [8], [11], [45], [48] also employs fully parametrized state space descriptions, but explores the use of subspace-based methods for the purposes of computing estimates. The advantage of this approach is the avoidance of iterative search and associated concerns with local minima. The disadvantage is the exponential growth in the size of certain Hankel matrices with respect to state and input-output dimension, although the recent contribution [49] has proposed a strategy for ameliorating this.

One outcome of this previous study is that, as is usual for ML methods, they are established to provide statistically optimal and hence accurate estimates. Furthermore, the employment of a fully parametrized state space description has proven useful not only in terms of numerical robustness, but also in terms of catering for multivariable systems. However, despite this progress, solutions that can reliably deal with systems of appreciable state and input–output dimension are not completely developed.

For example, as these dimensions grow, it is well recognized that the associated exponential growth in associated Hankel matrix dimension renders subspace methods too costly in terms of memory and processing speed requirements [45], [49]. Furthermore, as will be established here, while gradient based search methods can be highly effective on smaller size problems, they do not scale well in terms of how the same processing requirements grow with problem size. The system dimensions empir-

ically profiled in all the previously cited work are limited to a state order of three, and an input-output dimension of two.

Related to this, the work here develops methods for bilinear system estimation that do scale well with problem dimension, and hence function reliably for systems of significant size. For example, the methods developed here will be illustrated as being effective on a twentieth order, four input four output problem.

In light of the benefits established in the previous work [29], [45], [49] of using a ML criterion coupled with a fully parametrized state space description, this strategy will also be pursued here. However, this paper will examine the new idea of replacing the gradient based search proposed elsewhere with a nongradient based one. In particular, the use of the expectation-maximization (EM) algorithm [9] for computing ML estimates will be developed, analyzed, and profiled here.

The EM algorithm enjoys wide popularity and acceptance in a broad variety of fields of applied statistics. For example, areas as disparate as signal processing and dairy science routinely use the method [34], [4]. Despite this acceptance and success in other fields, it could be argued that in systems and control settings, the EM algorithm is not as well understood, accepted and utilized as it may deserve. The same phenomenon has been observed in econometrics [38].

Perhaps this is due to the fact that, while the EM algorithm does provide a general structure for the solution of estimation problems, if employed naïvely, it will generally fail on all but trivially sized problems due to effects of finite precision computation. This paper therefore develops a robust implementation whereby it is made explicit how customised methods should be employed in both the expectation and maximization steps in order to deliver a highly reliable algorithm.

Although this paper has a prime focus on EM based methods, it also examines the gradient based search methods proposed in [45], [46], since they are a highly effective technique. In relation to this, the work here establishes and illustrates the apparently new result that the local-coordinate approach developed in [45], [46] is precisely equivalent to ignoring the effects of overparametrization, and then performing standard Gauss–Newton optimization that employs a pseudo-inverse [based on singular value decomposition (SVD)] of an associated rank deficient Hessian approximation.

Recognizing this permits further performance tuning of the attractive methods developed in [45], [46]. Indeed, as illustrated here, the gradient based search involved with these techniques can often converge significantly faster than the EM methods developed here. However, again as will be illustrated, this best case performance involves important tradeoffs. For example, as the model state and input–output dimension grow, the computational load associated with gradient based search methods grows very much faster than for EM based methods. As a result, on systems of significant size (twenty state, four inputs and four outputs) where gradient based methods become impractical, EM techniques provide an effective alternative.

Another tradeoff is that EM based methods are more reliable, in that while their best case convergence rate is slower than the best case for gradient based methods, the variability in their performance is quite small. In particular, as will be shown here, they are quite robust against termination in local minima. Considering the difficulties of finding initial models for bilinear system estimates, this turns out to be a very important advantage.

Finally, this paper builds on earlier work in [14] that established the utility of EM-based methods for the estimation of linear and time invariant systems. Wherever possible, the derivation of the extended results here have been shortened by reference to that previous work. However, the work [14] contains a much fuller and more tutorial presentation of the principles and properties underlying the EM methods used here, and hence readers seeking more details on those topics are referred there.

II. Bilinear System Modelling

One of the most general models for the input-output behavior of a nonlinear system is the Volterra description

$$y_t = \sum_{i=1}^{n_d} \sum_{\tau_1=0}^{\infty} \sum_{\tau_2=0}^{\infty} \cdots \sum_{\tau_k=0}^{\infty} h_i(\tau_1, \tau_2, \ldots, \tau_k) \times u_{t-\tau_1} u_{t-\tau_2} \cdots u_{t-\tau_k} \quad (1)$$

which has a very long history [37]. Here, the terms $\{h_i(\tau_1, \tau_2, \ldots, \tau_k)\}$ are referred to as the Volterra kernels. In the special case of $n_d = k = 1$, the representation (1) becomes the impulse response description of a time invariant linear system as a special case.

A key difficulty with employing this model is that of determining the possibly large number of Volterra kernels. An obvious way to address this problem is to reduce complexity by constraining the kernels to be time-invariant and separable. In this case the Volterra series is realisable by a finite-order bilinear system [37], which in the single-input–single-output (SISO) case can be expressed as

$$y_t = \sum_{i=1}^{n_1} a_i y_{t-i} + \sum_{i=0}^{n_2} b_i u_{t-i} + \sum_{i=1}^{n_3} \sum_{j=1}^{n_3} c_{i,j} u_{t-i} y_{t-j} + \varepsilon_t. \quad (2)$$

Here, $\{y_t \in \mathbf{R}\}$ and $\{u_t \in \mathbf{R}\}$ are system output and input (respectively) and $\{\varepsilon_t\}$ is a zero mean stochastic process that accounts for measurement corruption. For the purposes of system identification, this model structure was used in [32], and in a slightly varied form in [12] and [13].

However, when developing models for multivariable data, the following state–space bilinear description, as employed in [11], [12], [45]–[47] is more tractable:

$$\begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} = \begin{bmatrix} A & F & B \\ C & G & D \end{bmatrix} \begin{bmatrix} x_t \\ u_t \otimes x_t \\ u_t \end{bmatrix} + \begin{bmatrix} w_t \\ v_t \end{bmatrix}. \quad (3)$$

Here, the vector sequences $\{x_t \in \mathbf{R}^n\}, \{u_t \in \mathbf{R}^m\}$, and $\{y_t \in \mathbf{R}^p\}$ represent the evolution of the system's state, input and output, and the quantities $A \in \mathbf{R}^{n\times n}, B \in \mathbf{R}^{n\times m}, F \in \mathbf{R}^{n\times mn}, C \in \mathbf{R}^{p\times n}, D \in \mathbf{R}^{p\times m}, G \in \mathbf{R}^{p\times mn}$ are constant matrices which, in the sequel, this paper will seek to estimate. Measurement and modeling corruptions are accounted for by

the zero mean i.i.d. processes $\{w_t\}$ and $\{v_t\}$. The symbol $\otimes$ is the Kronecker tensor product of matrices [5].

As noted in [45], input–output bilinear representations of the form shown in (2) neither subsume, nor are subsumed by, those with a state–space description (3).

III. Maximum Likelihood Estimation

For the purpose of estimating the parameters describing the bilinear model (3) on the basis of N observations of input $\{u_t\}$ and output $\{y_t\}$ the latter will be denoted as

$$U \triangleq \{u_1, u_2, \ldots, u_N\} \quad \text{and} \quad Y \triangleq \{y_1, y_2, \ldots, y_N\}. \tag{4}$$

As a solution strategy, the work here will employ an ML approach wherein a stochastic model for the modeling and measurement corruptions $\{w_t\}$ and $\{v_t\}$ is required. Here, it will be taken as the i.i.d. Gaussian one

$$\begin{bmatrix} w_t \\ v_t \end{bmatrix} \sim \mathcal{N}(0, \Pi), \quad \Pi \triangleq \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} > 0 \tag{5}$$

where, in the above, Π is partitioned conformally with respect to w_t and v_t. To allow for estimation from data records that have not attained steady-state operation, the initial state will be estimated via the parameters of a further Gaussian model

$$x_1 \sim \mathcal{N}(\mu, P_1). \tag{6}$$

The model (3), (5), (6) is therefore completely described by the elements of the parameter vector θ defined as

$$\theta^T \triangleq [\beta^T, \eta^T] \tag{7}$$

$$\beta^T \triangleq [\mathrm{vec}\{\Gamma\}^T, \mu^T] \quad \eta^T \triangleq [\mathrm{vec}\{\Pi\}^T, \mathrm{vec}\{P_1\}^T] \tag{8}$$

where the $\mathrm{vec}\{\cdot\}$ operator creates a vector from a matrix by stacking its columns on top of one another and

$$\Gamma \triangleq \begin{bmatrix} A & F & B \\ C & G & D \end{bmatrix}. \tag{9}$$

In order to estimate the vector θ parametrizing an underlying bilinear model (3), this paper examines the ML approach of using a value $\hat{\theta}_{\mathrm{ML}}$ defined as

$$\hat{\theta}_{\mathrm{ML}} \in \{\theta \in \Theta : p_\theta(Y) \geq p_{\bar{\theta}}(Y), \bar{\theta} \in \Theta\}. \tag{10}$$

Here, $p_\theta(Y)$ is the probability density function of the observed data Y conditioned on the system parameters being θ, which are assumed to lie within a compact set $\Theta \subset \mathbf{R}^d$ of candidate parameter vectors.

The use and analysis of the ML method for the general estimation problem is classical [20], [27], [42], [7]. A main attraction is the general (but not universal [26]) feature that ML estimators achieve optimal accuracy, in that they are asymptotically (in data length N) consistent, and achieve the Cramér–Rao lower bound on estimate variability [26], [27].

Despite these advantages, an important obstacle to employing the method is the difficulty of computing a value $\hat{\theta}_{\mathrm{ML}}$ that satisfies the criterion (10), since $p_\theta(Y)$ is typically non convex with respect to θ, and is also nonlinearly parametrized by θ. For example, in the bilinear system estimation case considered here [19]

$$L(\theta) \triangleq \log p_\theta(Y)$$

$$= -\frac{1}{2}\sum_{t=1}^{N} \log\det(C_t P_{t\,|\,t-1} C_t^T + R) - \tag{11}$$

$$\frac{1}{2}\sum_{t=1}^{N} (y_t - \hat{y}_{t\,|\,t-1})^T \left[C_t P_{t\,|\,t-1} C_t^T + R\right]^{-1} (y_t - \hat{y}_{t|t-1}) \tag{12}$$

$$C_t \triangleq C + G(u_t \otimes I_n) \tag{13}$$

where $\{\hat{y}_{t|t-1}\}$ is the one step ahead mean square optimal predictor, and $\{P_{t\,|\,t-1}\}$ the associated state estimate covariance which are both computed via an appropriate Kalman predictor that depends upon θ.

Note that in forming (12), constant terms that do not affect the maximizer of $L(\theta)$ have been neglected, and the so-called "log likelihood" $L(\theta)$ as opposed to $p_\theta(Y)$ is considered since it has the same maximizer as $p_\theta(Y)$, but is more convenient to work with.

IV. Bilinear Estimation Via the EM Algorithm

The EM algorithm is an iterative technique for obtaining ML estimates. It has its origins, in specific cases such as discrete state and measurement hidden Markov model estimation, stretching at least as far back as [2]. It first appeared in the general form which will be employed here in [9]. While it enjoys a history of success in areas of mathematical statistics, signal processing, and even dairy science [51], [4], [34], [31], [33], [41], it has only been employed in the control literature in certain specialised applications, such as that of estimation with censored data [16], [22].

To the authors' knowledge, it has not been previously employed for the purposes of bilinear system estimation. However, the current authors have recently made a detailed study of the method in the case of multivariable LTI system estimation [14], and this paper will draw on, and then extend certain results and techniques developed in that work.

A. The EM Algorithm

The key principle underlying the EM algorithm is the postulate of a so-called complete data set $Z = (Y, X)$, which consists not only of the actual observations Y, but also another set of data X, termed "missing data," that was not observed, and is a key design variable chosen by the user. The essential point of the EM algorithm is to choose X such that if it were available, the computation of an ML estimate with respect to Z would be straightforward.

These ideas are developed by first applying Bayes' rule to the joint density $p_\theta(Z)$ to obtain

$$p_\theta(Z) = p_\theta(Z|Y)p_\theta(Y)$$

and, therefore

$$L(\theta) \triangleq \log p_\theta(Y) = \log p_\theta(X,Y) - \log p_\theta(X\,|\,Y). \quad (14)$$

In this case, with $\mathbf{E}_{\theta'}\{\cdot\,|\,Y\}$ denoting expectation with respect to a probability density function determined by θ', and conditional upon data Y, then operating on both sides of (14) leads to

$$\begin{aligned} L(\theta) \triangleq \log p_\theta(Y) &= \mathbf{E}_{\theta'}\{\log p_\theta(Y)\,|\,Y\} \\ &= \mathbf{E}_{\theta'}\{\log p_\theta(X,Y)\,|\,Y\} \\ &\quad - \mathbf{E}_{\theta'}\{\log p_\theta(X\,|\,Y)\,|\,Y\} \\ &= \mathcal{Q}(\theta,\theta') - \mathcal{V}(\theta,\theta') \end{aligned} \quad (15)$$

where

$$\mathcal{Q}(\theta,\theta') \triangleq \mathbf{E}_{\theta'}\{\log p_\theta(X,Y)\,|\,Y\} \quad (16)$$

and

$$\mathcal{V}(\theta,\theta') \triangleq \mathbf{E}_{\theta'}\{\log p_\theta(X\,|\,Y)\,|\,Y\}. \quad (17)$$

The EM algorithm then proceeds by maximizing $\mathcal{Q}(\theta,\theta')$ with respect to θ in the hope of delivering a new estimate which is an improvement relative to θ'. An intuition behind this strategy is that although, by appropriate choice of X, maximization of $\log p_\theta(X,Y)$ is straightforward, since X is not available, an alternate strategy of maximizing an approximation of $\log p_\theta(X,Y)$ given as

$$\log p_\theta(X,Y) \approx \mathbf{E}_{\theta'}\{\log p_\theta(X,Y)\,|\,Y\} \triangleq \mathcal{Q}(\theta,\theta') \quad (18)$$

is used. To understand the utility of this approach, note that via the decomposition (15), the difference between the likelihoods associated with any two elements of Θ, say θ and $\hat\theta_k$, can be written

$$\begin{aligned} L(\theta) - L(\hat\theta_k) = [\mathcal{Q}(\theta,\hat\theta_k) - \mathcal{Q}(\hat\theta_k,\hat\theta_k)] \\ +[\mathcal{V}(\hat\theta_k,\hat\theta_k) - \mathcal{V}(\theta,\hat\theta_k)]. \end{aligned} \quad (19)$$

The second term on the right-hand side, by dint of the definition of $\mathcal{V}(\cdot,\cdot)$ may be identified as the Kullback–Leibler divergence between $p_\theta(X\,|\,Y)$ and $p_{\hat\theta_k}(X\,|\,Y)$ which has the property [24]

$$\mathcal{V}(\hat\theta_k,\hat\theta_k) - \mathcal{V}(\theta,\hat\theta_k) \geq 0$$

with equality if and only if $\log p_\theta(X\,|\,Y) = \log p_{\hat\theta_k}(X\,|\,Y)$ almost everywhere. Therefore, according to (19), any value of θ for which $\mathcal{Q}(\theta,\hat\theta_k) > \mathcal{Q}(\hat\theta_k,\hat\theta_k)$ implies that $L(\theta) > L(\hat\theta_k)$.

This suggests a strategy of maximizing $\mathcal{Q}(\theta,\hat\theta_k)$, which must increase $L(\theta)$ via (19), and then setting θ_{k+1} equal to this maximizer and repeating the process. That is, the EM algorithm proceeds via repeated application of the following two steps which start from an estimate $\hat\theta_k$ of $\hat\theta_{\mathrm{ML}}$ and update to a better one $\hat\theta_{k+1}$ via the following.

1) **E Step**

$$\text{Calculate: } \mathcal{Q}(\theta,\hat\theta_k). \quad (20)$$

2) **M Step**

$$\text{Compute: } \quad \hat\theta_{k+1} = \arg\max_{\theta\in\Theta} \mathcal{Q}(\theta,\hat\theta_k). \quad (21)$$

Since a single iteration of the EM algorithm is generally not sufficient to provide a satisfactory estimate of $\hat\theta_{\mathrm{ML}}$, an EM algorithm normally consists of more than one iteration, generating the sequence of increasingly good parameter estimates $\{\hat\theta_0,\hat\theta_1,\hat\theta_2,\ldots\}$.

B. Application to Bilinear Systems

The most crucial choice in the deployment of the EM algorithm is the selection of the missing data. The few previous applications of the EM algorithm in control relevant estimation problems have interpreted the choice literally, in the sense that they have employed EM based methods to handle the case of censored measurements [22], [16] with respect to SISO ARMAX model structures.

This paper takes a different approach, whereby it is noticed that if, in addition to the measurements Y and U, the state sequence

$$X \triangleq \{x_1, x_2, \ldots, x_{N+1}\} \quad (22)$$

were available, then it would be possible to extract an estimate of $\hat\theta_{\mathrm{ML}}$ directly from (3) using simple linear regression techniques. Since knowledge of X would so radically simplify the estimation problem, it is designated here as the EM algorithm's missing data. This approach has also been used for the purposes of multiple linear time series modeling in [39] appearing in the statistics literature.

With this definition of the missing data, the first part of the EM algorithm requires that the function $\mathcal{Q}(\theta,\hat\theta_k)$ be computed. This may be achieved via the following lemma.

Lemma 4.1: With regard to the model structure (3), (5), if the missing data X is defined by (22), then the function $\mathcal{Q}(\theta,\hat\theta_k)$ defined in (16) is given by

$$\begin{aligned} -2\mathcal{Q}(\theta,\hat\theta_k) = \log\det P_1 + \\ \mathrm{Tr}\left\{P_1^{-1}\mathbf{E}_{\hat\theta_k}\{(x_1-\mu)(x_1-\mu)^T\,|\,Y\}\right\} \\ + N\log\det\Pi + \\ N\mathrm{Tr}\{\Pi^{-1}[\Phi - \Psi\Gamma^T - \Gamma\Psi^T + \Gamma\Sigma\Gamma^T]\} \end{aligned} \quad (23)$$

where

$$z_t^T \triangleq \left[x_t^T, u_t^T \otimes x_t^T, u_t^T\right] \quad \xi_t^T \triangleq \left[x_{t+1}^T, y_t^T\right] \quad (24)$$

$$\Phi \triangleq \frac{1}{N}\sum_{t=1}^{N} \mathbf{E}_{\hat\theta_k}\left\{\xi_t\,\xi_t^T\big|\,Y\right\} \quad (25)$$

$$\Psi \triangleq \frac{1}{N}\sum_{t=1}^{N} \mathbf{E}_{\hat\theta_k}\left\{\xi_t\,z_t^T\big|\,Y\right\}, \quad \Sigma \triangleq \frac{1}{N}\sum_{t=1}^{N} \mathbf{E}_{\hat\theta_k}\left\{z_t\,z_t^T\big|\,Y\right\}. \quad (26)$$

Proof: See Appendix I. ∎

In order to calculate the matrix-valued quantities Φ,Ψ, and Σ required for the computation of $\mathcal{Q}(\theta,\hat\theta_k)$ we note that manipulations using the basic properties of the Kronecker product $\otimes$

allow Φ, Ψ, and Σ to be expressed as shown in (27)–(29) at the bottom of the page, where

$$\hat{x}_{t\,|\,N} \triangleq \mathbf{E}_{\hat{\theta}_k}\{x_t \,|\, Y\}. \tag{30}$$

The computation of Φ, Ψ, and Σ therefore requires the availability of $\hat{x}_{t\,|\,N}$ together with $\mathbf{E}_{\hat{\theta}_k}\{x_t x_t^T \,|\, Y\}$ and $\mathbf{E}_{\hat{\theta}_k}\{x_{t+1} x_t^T \,|\, Y\}$ for $t \in [1, N]$. Crucially, at least the first two of these quantities may be computed using a Kalman smoother by exploiting the fact that the bilinear system (2) is also expressible in the following time-varying form:

$$\begin{aligned} x_{t+1} &= A_t x_t + B u_t + w_t \\ y_t &= C_t x_t + D u_t + v_t \end{aligned}$$

where

$$A_t \triangleq A + F(u_t \otimes I_n) \quad C_t \triangleq C + G(u_t \otimes I_n). \tag{31}$$

This provides $\hat{x}_{t\,|\,N}$ and $\mathbf{E}_{\hat{\theta}_k}\{x_t x_t^T \,|\, Y\}$ via $\mathbf{E}_{\hat{\theta}_k}\{x_t \,|\, Y\}$ and its associated covariance $P_{t\,|\,N}$.

However, the quantities $\mathbf{E}_{\hat{\theta}_k}\{x_{t+1} x_t^T \,|\, Y\}, t \in [1, N+1]$ are not obtainable by standard smoothing algorithms. Furthermore, the authors have found that any naïve implementation of the smoothing step (and subsequent maximization steps to come) will lead to failure on all but trivially sized problems due to errors associated with finite precision computation. On the other hand, (as will be illustrated empirically) with appropriate study and modifications embodied in the following lemma, the EM based methods derived here can be rendered highly reliable if appropriate steps are taken to use numerically robust procedures.

Lemma 4.2: The components

$$\mathbf{E}_{\hat{\theta}_k}\left\{y_t \; x_t^T \middle| \, Y\right\} = y_t \hat{x}_{t\,|\,N}^T \tag{32}$$

$$\mathbf{E}_{\hat{\theta}_k}\left\{x_t \; x_t^T \middle| \, Y\right\} = \hat{x}_{t\,|\,N} \hat{x}_{t\,|\,N}^T + P_{t\,|\,N}^{1/2} P_{t\,|\,N}^{T/2} \tag{33}$$

$$\mathbf{E}_{\hat{\theta}_k}\left\{x_t \; x_{t-1}^T \middle| \, Y\right\} = \hat{x}_{t\,|\,N} \hat{x}_{t-1\,|\,N}^T + M_{t\,|\,N} \tag{34}$$

required for the computation of (27)–(28) may be robustly computed as follows. The smoothed state estimate $\{\hat{x}_{t\,|\,N}\}$ is calculated via the reverse-time recursion

$$\hat{x}_{t\,|\,N} = \hat{x}_{t\,|\,t} + J_t\left[\hat{x}_{t+1\,|\,N} - \bar{A}_t \hat{x}_{t\,|\,t} - \bar{B} u_t - S R^{-1} y_t\right] \tag{35}$$

$$J_t \triangleq P_{t\,|\,t} \bar{A}_t^T P_{t+1\,|\,t}^{-1} \tag{36}$$

where all covariance matrices are computed from their square roots as, for example, $P_{t\,|\,t} = P_{t\,|\,t}^{1/2} P_{t\,|\,t}^{T/2}$. These are found by performing the following $\mathcal{QR}$-decompositions:

$$\begin{bmatrix} P_{t\,|\,t}^{T/2} \bar{A}_t^T & P_{t\,|\,t}^{T/2} \\ \bar{Q}^{T/2} & 0 \\ 0 & P_{t+1\,|\,N}^{T/2} J_t^T \end{bmatrix} = \mathcal{Q}^1 \begin{bmatrix} \mathcal{R}_{11}^1 & \mathcal{R}_{12}^1 \\ 0 & \mathcal{R}_{22}^1 \\ 0 & 0 \end{bmatrix} \tag{37}$$

$$\begin{bmatrix} P_{t-1\,|\,t-1}^{T/2} \bar{A}_{t-1}^T \\ \bar{Q}^{T/2} \end{bmatrix} = \mathcal{Q}^2 \begin{bmatrix} \mathcal{R}_1^2 \\ 0 \end{bmatrix} \tag{38}$$

$$\begin{bmatrix} R^{T/2} & 0 \\ P_{t\,|\,t-1}^{T/2} \bar{C}_t^T & P_{t\,|\,t-1}^{T/2} \end{bmatrix} = \mathcal{Q}^3 \begin{bmatrix} \mathcal{R}_{11}^3 & \mathcal{R}_{12}^3 \\ 0 & \mathcal{R}_{22}^3 \end{bmatrix} \tag{39}$$

and then setting

$$P_{t\,|\,N}^{T/2} = \mathcal{R}_{22}^1 \quad P_{t\,|\,t-1}^{T/2} = \mathcal{R}_1^2 \quad P_{t\,|\,t}^{T/2} = \mathcal{R}_{22}^3. \tag{40}$$

Here, the matrices $\bar{A}_t, \bar{B}, \bar{Q}, P_{t\,|\,t}$, and $P_{t+1\,|\,t}$ are defined as

$$\bar{A}_t \triangleq A - S R^{-1} C + (F - S R^{-1} G)(u_t \otimes I_n)$$

$$\bar{B} \triangleq B - S R^{-1} D \tag{41}$$

$$\bar{Q} \triangleq Q - S R^{-1} S^T. \tag{42}$$

The matrices $M_{N\,|\,N}$ and $M_{N+1\,|\,N}$ are calculated via the initialization

$$\begin{aligned} M_{N\,|\,N} &= (I - K_N \bar{C}_N) \bar{A}_{N-1} P_{N-1\,|\,N-1} \\ M_{N+1\,|\,N} &= \bar{A}_N P_{N\,|\,N} \end{aligned} \tag{43}$$

followed by the the backward recursion $\{M_{t\,|\,N}\}_{t=2}^{N-1}$ given by

$$M_{t\,|\,N} = P_{t\,|\,t} J_{t-1}^T + J_t (M_{t+1\,|\,N} - \bar{A}_t P_{t\,|\,t}) J_{t-1}^T. \tag{44}$$

$$\Phi = \sum_{t=1}^{N} \begin{bmatrix} \mathbf{E}_{\hat{\theta}_k}\left\{x_{t+1} \; x_{t+1}^T \middle| \, Y\right\} & \hat{x}_{t+1\,|\,N} y_t^T \\ y_t \hat{x}_{t+1\,|\,N}^T & y_t y_t^T \end{bmatrix} \tag{27}$$

$$\Psi = \sum_{t=1}^{N} \left[\begin{bmatrix} 1 & u_t^T \end{bmatrix} \otimes \begin{bmatrix} \mathbf{E}_{\hat{\theta}_k}\left\{x_{t+1} \; x_t^T \middle| \, Y\right\} \\ y_t \hat{x}_{t\,|\,N}^T \end{bmatrix}\right] \begin{bmatrix} \hat{x}_{t+1\,|\,N} \\ y_t \end{bmatrix} u_t^T \tag{28}$$

$$\Sigma = \sum_{t=1}^{N} \begin{bmatrix} \left(\begin{bmatrix} 1 \\ u_t \end{bmatrix}\begin{bmatrix} 1 \\ u_t \end{bmatrix}^T\right) \otimes \mathbf{E}\left\{x_t \; x_t^T \middle| \, Y\right\} & \begin{bmatrix} 1 \\ u_t \end{bmatrix} \otimes \left(\hat{x}_{t\,|\,N} u_t^T\right) \\ \begin{bmatrix} 1 & u_t^T \end{bmatrix} \otimes \left(u_t \hat{x}_{t\,|\,N}^T\right) & u_t u_t^T \end{bmatrix} \tag{29}$$

Finally, the reverse time recursion (35) is initialized by running to $t = N$ the (robust) Kalman filter recursions

$$K_t = P_{t\,|\,t-1}\bar{C}_t^T\left(\bar{C}_t P_{t\,|\,t-1}\bar{C}_t^T + R\right)^{-1} = \mathcal{R}_{12}^3 \quad (45)$$

$$\hat{x}_{t\,|\,t-1} = \bar{A}_{t-1}\hat{x}_{t-1\,|\,t-1} + \bar{B}u_{t-1} + SR^{-1}y_{t-1} \quad (46)$$

$$\hat{x}_{t\,|\,t} = \hat{x}_{t\,|\,t-1} + K_t\left(y_t - \bar{C}_t\hat{x}_{t\,|\,t-1} - Du_t\right) \quad (47)$$

for $t = 1, \ldots, N$.

Proof: See Appendix II. ∎

Equations (27) and (28) in concert with Lemmas 4.1 and 4.2 demonstrate that the computation of $\mathcal{Q}(\theta, \hat{\theta}_k)$ is somewhat complex, but straightforward. We now turn to the final part of the EM algorithm—the M-step, which requires the calculation of the value of θ that maximizes $\mathcal{Q}(\theta, \hat{\theta}_k)$. This is also straightforward, but must still be done with care in order to preserve numerical robustness. Its precise formulation depends on the following parameter space specification.

Standing Assumptions 4.1: Recalling the decomposition $\theta = [\beta^T, \eta^T]^T$ defined in (7), the set of candidate parameter vectors Θ is taken as

$$\Theta = \Theta_1 \times \Theta_2 \quad \beta \in \Theta_1 \quad \eta \in \Theta_2 \quad (48)$$

where Θ_1 is a closed hypercube in $\mathbf{R}^\ell, \ell = n^2(1+m)+n(m+1)(p+1)+mp$, and Θ_2 is a compact subset of $\mathbf{R}^v, v = n^2 + (n+p)^2$ for which all $\eta \in \Theta_2$ imply symmetric positive definite Π, P_1.

Lemma 4.3: Let Σ defined in (26) satisfy $\Sigma > 0$ and be used to define $\hat{\beta}$ according to [Ψ is also defined in (26)]

$$\hat{\beta} \triangleq [\mathrm{vec}\{\Gamma\}^T, \mu^T]^T, \quad \Gamma \triangleq \begin{bmatrix} A & F & B \\ C & G & D \end{bmatrix} = \Psi\Sigma^{-1}$$
$$\mu \triangleq \hat{x}_{1\,|\,N}. \quad (49)$$

If Θ defined by the the parameter space Assumptions 4.1 is such that $\hat{\beta}$ lies within Θ_1, then for any fixed $\eta \in \Theta_2$, the point (49) is the unique maximizer

$$\hat{\beta} = \arg\max_{\theta\in\Theta_1} \mathcal{Q}\left(\begin{bmatrix}\beta \\ \eta\end{bmatrix}, \theta'\right). \quad (50)$$

Furthermore, $\hat{\eta}$ given by

$$\hat{\eta} \triangleq \left[\mathrm{vec}\Pi^T, \mathrm{vec}{P_1}^T\right]^T, \quad \Pi \triangleq \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} = \mathcal{R}_{22}^{T/2}\mathcal{R}_{22}^{1/2} \quad (51)$$

$$P_1 \triangleq P_{1\,|\,N}^{1/2}P_{1\,|\,N}^{T/2} \quad (52)$$

forms a stationary point of $\mathcal{Q}(\cdot, \theta')$ with respect to η. Here, $P_{1\,|\,N}^{1/2}$ is defined by (37), (40), and $\mathcal{R}_{22}^{1/2}$ is defined by the Cholesky factorization (see [15, Alg. 4.2.4])

$$\begin{bmatrix} \Sigma & \Psi^T \\ \Psi & \Phi \end{bmatrix} = \begin{bmatrix} \mathcal{R}_{11} & \mathcal{R}_{12} \\ \oslash & \mathcal{R}_{22} \end{bmatrix}^T \begin{bmatrix} \mathcal{R}_{11} & \mathcal{R}_{12} \\ \oslash & \mathcal{R}_{22} \end{bmatrix}. \quad (53)$$

Note that the right-hand side of the expression for Π in (51) is

$$\Pi = \Phi - \Psi\Sigma^{-1}\Psi^T \quad (54)$$

realized in a numerically robust fashion that ensures essential properties of symmetry and non negative–definiteness of the result.

Proof: See Appendix III. ∎

C. A Summary of the Algorithm

The preceding derivations are now summarized in the interests of clearly defining the new algorithm proposed here.

EM Algorithm 4.1 (EM Algorithm for Bilinear Systems):

1) Initialize the algorithm by choosing a parameter vector $\hat{\theta}_0$.
2) **(E-Step)** Employ the square-root implementation of the modified Kalman smoother presented in Lemma 4.2 in conjunction with the parameter estimate $\hat{\theta}_k$ to calculate the matrices Φ, Ψ, and Σ as shown in (27) and (28).
3) **(M-Step)** In order to choose an updated parameter estimate $\hat{\theta}_{k+1}$, select Π, Γ, μ, and P_1 according to (49), (51), and (52).
4) If the algorithm has converged, terminate, otherwise return to step 2).

Regarding step 4), obvious strategies for gauging convergence involve copying those developed for gradient based search [10], [36]. In particular, this paper suggests a strategy of termination when relative likelihood increase on an iteration drops below a predetermined threshold.

D. Properties

This section describes some properties of the new methods proposed here. In relation to this, an essential point is that Algorithm 4.1 employs a nonminimal state–space parametrization and thus, for any candidate model, there exists a potentially infinite number of equivalent models mutually related via similarity transformations.

This raises obvious questions relating uniqueness and termination of iterates $\hat{\theta}_k$. Addressing these issues is a particular focus of the analysis to follow, for which the essential points are that under a persistence of excitation condition, the iterates are well defined, imply an evolution of likelihood that is attracted to a local maximizer, and which do not "wander" amongst systems that are input–output equivalent.

Lemma 4.4: Suppose that $\Pi, P_1 > 0$ for a system parametrized by $\hat{\theta}_k$ and that for the given data length N, the input sequence $\{u_t\}$ satisfies

$$\frac{1}{N}\sum_{t=1}^{N}\begin{bmatrix}1 \\ u_t\end{bmatrix}\begin{bmatrix}1 \\ u_t\end{bmatrix}^T > 0. \quad (55)$$

Then Σ, defined by (26), is positive definite and $\hat{\theta}_{k+1}$ is uniquely defined.

Proof: See Appendix IV. ∎

The property that perhaps most strongly recommends employing an EM algorithm is that further iterations cannot result in a lower likelihood, as was explained in Section IV-A. Algorithm 4.1 inherits this. Moreover, in the specific EM algorithm case considered here, the following lemma also establishes that

despite the overparametrization, the sequence of estimates $\hat{\theta}_k$ involved with the EM approach is uniquely defined.

Theorem 4.2: Let $\hat{\theta}_k$ parametrize a system (3), (5) with $\Pi, P_1 > 0$. Suppose that the input sequence $\{u_1, \ldots, u_N\}$ satisfies (55). Then

$$L(\hat{\theta}_{k+1}) \geq L(\hat{\theta}_k) \tag{56}$$

with equality if and only if $\hat{\theta}_{k+1} = \hat{\theta}_k$.

Proof: This follows from [14, Cor. 5.1] with the inclusion of input condition (55). ∎

Since the sequence of likelihoods associated with the EM algorithm is monotonically increasing it is clear that this sequence will converge under the very mild condition that $L(\theta)$ is bounded above for $\theta \in \Theta$. In the following theorem, we demonstrate that the associated limiting parameter values have an interpretation in terms of the likelihood function.

Theorem 4.2: Let $\{\hat{\theta}_k\} \subset \Theta$ be a sequence of estimates generated by EM Algorithm 4.1. Then a limit point $\theta^\star$ of $\{\hat{\theta}_k\}$, is a stationary point of $L(\theta)$ and the sequence $\{L(\hat{\theta}_k)\}$ converges monotonically to $L(\theta^\star)$.

Proof: This follows from the results provided in [14] extended in an obvious manner to the bilinear case considered here. ∎

Note that the conditions in this theorem that the functions $\mathcal{Q}$ and L are continuous on Θ and that L is differentiable in its interior are very mild and will be satisfied if, for example, $\Pi > 0$ and $P_1 > 0$ for all $\theta \in \Theta$. Again, see [14] for further discussion on this point.

V. Bilinear Estimation Via Gradient Based Search

The previous work most closely related to this paper is that of [46], [47], [45], and [49] wherein a different approach to the multivariable bilinear system estimation problem is taken, and which employs the model structure

$$\begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} = \begin{bmatrix} A & F & B \\ C & G & D \end{bmatrix} \begin{bmatrix} x_t \\ u_t \otimes x_t \\ u_t \end{bmatrix} + \begin{bmatrix} K \\ I \end{bmatrix} v_t. \tag{57}$$

This is an instance of the more general case (3) via the restriction $w_t = Kv_t$ for some zero mean i.i.d. process $\{v_t\}$. An attractive feature of this model structure is the relative simplicity of the associated mean square optimal one-step-ahead predictor, which can be written

$$\begin{aligned} \hat{x}_{t+1\,|\,t} &= \bar{A}_t \hat{x}_{t\,|\,t-1} + \bar{B} u_t + K y_t \\ \hat{y}_{t|t-1} &= C_t \hat{x}_{t\,|\,t-1} + D u_t \end{aligned} \tag{58}$$

where

$$\bar{A}_t \triangleq A - KC + (F - KG)(u_t \otimes I_n) \quad \bar{B} \triangleq B - KD \tag{59}$$

$$C_t \triangleq C + G(u_t \otimes I_n). \tag{60}$$

Therefore, with the assumption that $\{v_t\}$ is distributed as

$$v_t \sim \mathcal{N}(0, \sigma^2 I_p) \tag{61}$$

where $\sigma^2 > 0 \in \mathbf{R}$ (the more general case of spatial correlation can easily be handled [46], [29], but at the expense of extra notation which will detract from the essential arguments to follow) and with the redefinition of the parameter vector as

$$\theta^T \triangleq [\mathrm{vec}\{A\}^T, \ldots, \mathrm{vec}K^T] \tag{62}$$

then neglecting constant terms which are immaterial to the estimation process, the associated log likelihood function for the data is given as

$$L(\theta) = -\frac{Np}{2} \log \sigma^2 - \frac{1}{\sigma^2} V_N(\theta) \quad V_N(\theta) = \|E_N(\theta)\|^2. \tag{63}$$

Here, the dependence of the predictor in (58) on the parameter vector θ is denoted by $\hat{y}_{t|t-1}(\theta)$

$$E_N(\theta) \triangleq \left[y_1^T - \hat{y}_{1|0}^T(\theta), \ldots, y_N^T - \hat{y}_{N|N-1}^T(\theta) \right]^T \tag{64}$$

and the norm used in (63) is the Euclidean one. Notice that, according to (63) there is an essential decoupling between the estimation of σ^2 and the elements of the parameter vector θ defined in (62). Namely, under the model structure (57), the ML estimate $\hat{\theta}_{\mathrm{ML}}$ is given as an element satisfying

$$\hat{\theta}_{\mathrm{ML}} \in \{\theta \in \Theta : \|E_N(\theta)\| \leq \|E_N(\bar{\theta})\| \quad \forall \bar{\theta} \in \Theta\}. \tag{65}$$

In recognition of this, the previous work [45]–[47], [49] has focussed on the problem of minimizing $\|E_N(\theta)\|$, and has explored a gradient-based search approach.

Indeed, a gradient search strategy is employed in a wide variety of system identification applications [27], where it is common to note that, via the quadratic nature of $\|E_N(\theta)\|$, a first-order approximation can have good local accuracy. Therefore, with θ_k denoting the k'th iterate in a search for $\hat{\theta}_{\mathrm{ML}}$, the $k + 1'$st is found by

$$\theta_{k+1} = \arg\min_\theta \|E(\theta_k) + E'(\theta_k)(\theta - \theta_k)\| \approx \arg\min_\theta \|E(\theta)\| \tag{66}$$

where the prime notation $\cdot'$ denotes differentiation with respect to θ and hence, according to what is known as a "damped" Gauss–Newton update strategy

$$\theta_{k+1} = \theta_k + \mu p \tag{67}$$

where $\mu \in (0, 1]$ is a damping parameter, and p is a search direction which, according to (66) satisfies

$$E'(\theta_k)^T E'(\theta_k) p = -E'(\theta_k)^T E(\theta_k) \tag{68}$$

so that

$$p = -[E'(\theta_k)^T E'(\theta_k)]^\dagger E'(\theta_k)^T E(\theta_k). \tag{69}$$

Here, the indicated pseudoinverse is defined as

$$[E'(\theta_k)^T E'(\theta_k)]^\dagger \triangleq V_1 S_1^{-1} V_1^T \tag{70}$$

$$E'(\theta_k)^T E'(\theta_k) \triangleq [V_1, V_2] \begin{bmatrix} S_1 & \oslash \\ \oslash & \oslash \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = V_1 S_1 V_1^T \tag{71}$$

with the indicated rank based decomposition on the right of (71) being a singular value one [15]. When $E'_N(\theta_k)$ is full-column rank, then the above pseudoinverse will become a regular matrix inversion. However, with the choice (62) in which none of the elements in the system matrices of (57) are constrained, then the ensuing over-parametrization will ensure that $E'_N(\theta_k)$ is always rank deficient.

As recognized in [45] and [46], this rank deficiency can be characterized by identifying the set of systems $[\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}, \tilde{F}, \tilde{G}, \tilde{K}]$ that are input–output equivalent to $[A, B, C, D, F, G, K]$ according to

$$\tilde{A} = TAT^{-1} \quad \tilde{B} = TB \quad \tilde{C} = CT^{-1} \quad \tilde{D} = D \tag{72}$$

$$\tilde{F} = TFT^{-1} \quad \tilde{G} = GT^{-1} \quad \tilde{K} = TK \tag{73}$$

where $T \in \mathbf{R}^{n \times n}$ is an arbitrary invertible matrix. While this is a nonlinear mapping with respect to T, a locally linear approximating valid for small perturbations ΔT about $T = I_n$ may, as established in [45], [46], be expressed in the parameter space (62) as

$$\tilde{\theta} = \theta + Q \mathrm{vec}\{\Delta T\} \tag{74}$$

where

$$Q \triangleq \begin{bmatrix} A^T \otimes I_n - I_n \otimes A \\ B^T \otimes I_n \\ -I_n \otimes C \\ \oslash_{mp \times n^2} \\ \mathcal{F} \\ \mathcal{G} \\ K^T \otimes I_n \end{bmatrix} \quad \mathcal{F} \triangleq \begin{bmatrix} F_1^T \otimes I_n - I_n \otimes F_1 \\ \vdots \\ F_m^T \otimes I_n - I_n \otimes F_m \end{bmatrix} \tag{75}$$

$$\mathcal{G} \triangleq \begin{bmatrix} -I_n \otimes G_1 \\ \vdots \\ -I_n \otimes G_m \end{bmatrix} \tag{76}$$

and $F \triangleq [F_1, \ldots, F_m], G \triangleq [G_1, \ldots, G_m]$ with $F_i \in \mathbf{R}^{n \times n}, G_i \in \mathbf{R}^{n \times n}$. This implies that any search update in a direction $p = \tilde{\theta} - \theta = Q \mathrm{vec}\{\Delta T\}$ for any ΔT will only yield a system with equivalent input output properties and, hence, the columns of Q (locally) span the space of equivalent systems.

In recognition of this, the works [45], [46] (and in the linear case of $F = G = 0$ the papers [3], [28], [29]) suggest the use of local coordinates β that parametrize only the space of non input–output equivalent systems as

$$\tilde{\theta} = \theta + P\beta \tag{77}$$

where the columns of P are chosen (for example, by a QR factorization) to be orthogonal to the columns of Q. That is, the previous work [3], [28], [29], [45], [46] suggests that gradient based search should be performed in the reduced dimension space parametrized by β, whereby according to a damped Gauss–Newton search strategy

$$\theta_{k+1} = \theta_k + \mu P q \quad q = -[E'(\beta_k)^T E'(\beta_k)]^\dagger E'(\beta_k)^T E(\theta_k) \tag{78}$$

where $\cdot'$ now denotes differentiation with respect to β. A key motivation for this "local coordinate" approach is that β has dimension n^2 smaller than that of θ.

However, in assessing the utility of this strategy, a clear question arises as to how this search direction q formed via a local coordinate strategy relates to that obtained using a full strategy, i.e., p. In fact, under mild assumptions, they are identical.

Theorem 5.1: Consider the Gauss–Newton search directions p and q defined in (69) and (78), respectively. Suppose that the data is sufficiently informative that $\partial E(\beta)/\partial\beta$ has full-column rank. Then $p = q$.

Proof: We provide only a sketch of the proof, since much of it is identical to that provided in [50] where the LTI case is considered. The essential argument is that according to (77)

$$\frac{\partial}{\partial \beta} E(\beta) = \frac{\partial}{\partial \theta} E(\theta) P \tag{79}$$

and furthermore, according to the assumptions of the theorem, as derived in [50]

$$P = V_1 R \tag{80}$$

where R is some unitary matrix and V_1 is defined in (71). Substituting (80) into (79) and then into (78) establishes the result. ■

That is, local coordinate based Gauss–Newton search is identical to fully parametrized Gauss–Newton search, provided that the data is sufficiently informative that the row dimension of V_1 used via (71) in the fully parametrized case is chosen identical to the row dimension of P in (77) for local coordinate search.

This illustrates that local coordinate based search can be viewed as a special case of the fully parametrized search (67). The utility of this, is that it indicates that the choice of the row dimension of V_1 can be seen as a tuning parameter for which, at one value, local-coordinate search is obtained, but for others, a perhaps enhanced convergence rate is achievable.

Indeed, the experience of the authors is that, in many cases, it can be very valuable to truncate the row dimension of V_1 commensurate with the singular value spread in the corresponding S_1 being no greater than 10^6. The intuition here being to concentrate attention on directions of sufficiently changing cost, and ignore overly flat "valley" directions. We refer the reader to [50] for further details, which are implemented in the freely available toolbox [35].

VI. Computational Complexity

Having now developed a new EM based algorithm, and summarized existing gradient search based methods, the remainder

of the paper studies the relative benefits, computational costs, and performance of the two approaches. The first consideration in this area is that of computational load. In what follows, the order notation $y(x) = O(x)$ implies that

$$\limsup_{x\to\infty} \frac{y(x)}{x} < \infty.$$

A. EM Algorithm 4.1

A single iteration of EM Algorithm 4.1 is comprised of an E-step and an M-step. The E-step largely consists of a Kalman smoothing operation, for which, if the number of states n is (as is typical) larger than the number of inputs m or outputs p, the dominating computational cost is incurred by multiplying two dense $n \times n$ matrices at each sampling instant. That is, the cost of the E-step is $O(n^3N)$.

Unlike the operations found in the E-step, which are performed as many times as there are data points, those of the M-step are performed only once per iteration. A robust implementation involving pivoting operations to form Cholesky factors and solve linear systems will, after considering the dimensions of the quantities involved, incur a cost of $O(m^3n^3)$ computations [15]. This load is likely to be smaller than that associated with the E-step since, for a typical system, $N > m^3$.

Therefore, under the assumptions that $N > m^3$ and $n > m, p$, the computational cost *per iteration* of the EM based Algorithm 4.1 will be $O(n^3N)$.

B. Gauss–Newton Gradient Based Search

Interalia, the Gauss–Newton algorithm requires computation of the Jacobian matrix $E'(\beta)$, the Hessian approximation $E'(\beta)^TE'(\beta)$, and a QR factorization of (75) in order to obtain the search direction. Overwhelmingly, these computations dominate the FLOP count for each iteration of the algorithm. There are other operations necessary such as the computation of prediction errors $E(\theta)$, but these will not be further considered due to their significantly lower relative cost.

To examine the cost of the dominating operations, denote by n_θ, n_β, and n_f the quantities

$$n_\theta = (n+p)(n+nm+m)+np, \quad n_\beta = n_\theta - n^2 \tag{81}$$

$$n_f = 4n^2(1+m)+2p(2mn+4n+m+2) + 6n+3mn+4m \tag{82}$$

which are, respectively, the dimensions of θ defined by (62), the dimension of β defined in (77) (see [30]) and the number of multiplications required to compute $\mathrm{d}E(\beta)/\mathrm{d}[\beta]_k$ for a *single time update* and with respect to the k'th component of β. The latter is found by considering the associated state–space system that must be simulated, and simply counting the operations involved. Since this is lengthy, but straightforward, the details are omitted.

Since $E'(\beta)$ has N rows, then its computation involves $O(Nn_\beta n_f)$ FLOPS. Furthermore, since Q defined in (75) is of dimension $n_\theta \times n^2$, then a factorization of it in order to find P in (77) will require $O(n_\theta^3)$ FLOPS [15]. Finally, the formation of $E'(\beta)^TE'(\beta)$ necessary for the computation of the search direction q in (78) will require $O(Nn_\beta^2 p)$ FLOPS.

Therefore, under the assumption that $Np > n^2m$ so that the factorization of P is not the dominating term, there is a requirement of $O(Nn^4m^2p)$ FLOPS *per iteration* for a Gauss–Newton type search.

Compared to the EM algorithm developed here, this is $O(nm^2p)$ more operations per iteration, which can be significant. For example, in a tenth order three input/output situation profiled in the next section, the above analysis indicates that the EM based approach of this paper involves a FLOP requirement per iteration that is less that $1/200'$th of that required by the Gauss–Newton approach.

Of course, there are many other factors to consider such as computational load, memory requirements, suitability for caching, and of course the number of required iterations. Since a theoretical analysis of these issues is not tractable, they will be dealt with empirically in the following section.

VII. Empirical Study

This section is devoted to profiling the performance of the EM-based algorithm derived here relative to pre-existing gradient search methods based on Gauss–Newton iterations. For this purpose, we begin with a simple example whereby the true underlying system has order $n = 2, m = 2$ inputs and $p = 2$ outputs, and is given by the structure (57) with the choices made in [11] of

$$A = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.3 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \quad D = I_2 \tag{83}$$

$$F = \begin{bmatrix} 0.6 & 0 & 0.2 & 0 \\ 0 & 0.4 & 0 & 0.5 \end{bmatrix} \quad G = 0$$

$$K = \begin{bmatrix} 0.050 & 0.041 \\ 0.039 & 0.036 \end{bmatrix}. \tag{84}$$

This system was simulated with input $\{u_t\}$ and measurement corruption $\{v_t\}$ both being white random processes distributed as $v_t \sim \mathcal{N}(0, 0.01 \times I), u_t \sim \mathcal{N}(0, I)$ together with $N = 500$ ensuing data samples being used for identification.

In the first instance, the EM and GN methods were both initialized by using a subspace algorithm (N4SID derived in [44]) and a *linear* model structure to find preliminary values of A, B, C, D with $F, G = 0$ being used. For the EM method, $Q = I, R = 0.1I, S = 0, x_1 = 0, P_1 = I$ were chosen, and for the GN method, K was initialized as the steady-state Kalman gain implied by A, C, Q, R, S according to the solution of the associated Riccati equation.

With this initialization point fixed, one hundred further data sets were generated with different input and noise realizations, and both the EM and GN methods were used to find bilinear system estimates. The results, in terms of the convergence of the methods to the ML estimate, are shown in Fig. 1. There the evolution of the sample mean square prediction error (y axis) of the model obtained at the k'th iteration (x axis) is shown for the EM method (solid line), and the GN method (dashed line). The thicker lines are the average behavior over the one hundred

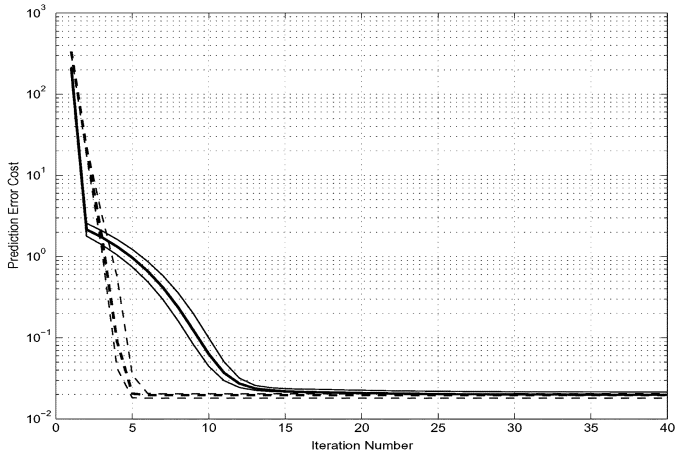

Fig. 1. Average, best, and worst case prediction error cost at each iteration of the EM (solid lines) and GN (dashed lines) algorithms; the thick lines indicate average performance while the thin lines indicate best and worst case performance for each algorithm. The GN algorithm performs well in this case with the worst case GN run being superior to the best case EM run.

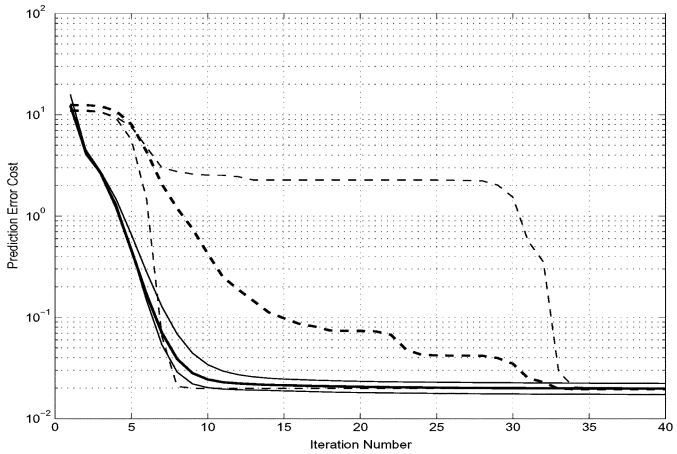

Fig. 2. Average, best, and worst case prediction error cost at each iteration of the EM (solid lines) and GN (dashed lines) algorithms; the thick lines indicate average performance while the thin lines indicate best and worst case performance for each algorithm. Note that the best and worst case lines for GN differ greatly from the average line, which suggests highly variable convergence rates in this case.

estimation experiments, and the thinner lines surrounding them are the best and worst cases chosen from the same ensemble.

Clearly, both methods are effective in that they compute ML estimates. This is evidenced by the achieved mean square errors being decreased to the global minimum value of $\mathrm{Tr}\{R\}$ in all cases. Furthermore, both methods appear reliable in that there is little variability in the convergence behavior on different data sets. Finally, in terms of computational load, the GN method is superior since on such a small sized problem its per-iteration FLOP count is comparable to EM but, as illustrated, it converges more rapidly and hence required fewer iterations.

It is important to note, that since a linear model is fitted to the nonlinear system (83), (84) as a starting point, it is quite a poor initialization, as illustrated by the high initial cost in Fig. 1. Consider now the case of the same one hundred data sets, but with a different initialization point, that was chosen randomly as

$$A = \begin{bmatrix} 0.0143 & 0.0671 \\ 0.0987 & 0.0084 \end{bmatrix} \quad B = \begin{bmatrix} 0.0067 & 0.0121 \\ 0.0346 & 0.0370 \end{bmatrix}$$
$$C = \begin{bmatrix} 0.0535 & 0.0610 \\ 0.0369 & 0.0949 \end{bmatrix} \quad D = \begin{bmatrix} 0.0137 & 0.0493 \\ 0.0076 & 0.0275 \end{bmatrix}$$
$$K = \begin{bmatrix} 0.0502 & 0.0410 \\ 0.0390 & 0.0356 \end{bmatrix} \quad F, G = 0_{2\times 4}.$$

The average, best and worst case convergence behavior for EM (solid line) and GN (dashed line) over the different data realizations for this case is shown in Fig. 2. First, note that in terms of initial cost, this initialization is an order of magnitude better than the one found previously by linear subspace identification.

More importantly though, note that the behavior of the EM method is essentially unchanged relative to the previous initialization, while the GN method performance is seriously degraded. Its average performance is now clearly inferior to the EM method, and the variability in performance over the data sets is now quite large.

This illustrates a fundamental aspect. In assessing the choice between EM and GN based methods for bilinear system estimation, there is a tradeoff between reliability and best case performance. The experience of the authors is that, as just illustrated, EM based methods are very reliable, but at the expense of slower convergence rate. This feature of EM-based methods is well recognized in the statistics literature [33], and will be further illustrated in this section.

This paper therefore suggests that a hybrid approach combining the two methods is a worthwhile strategy. This involves the initial iterations being performed by EM, where its ability to robustly "steer" the iterations toward a minimizer is exploited, with the final iterations performed via GN, where its strengths in providing fast local convergence are realized. Since both methods work with freely parametrized models, such a handover between schemes is straightforward. For the same data sets and initialization as generated in Fig. 2, this hybrid approach is illustrated in Fig. 3 as the dashed line. Handover to the GN methods was made after four EM iterations. Note how, in comparison to Fig. 2, the hybrid approach is able to capture both the robustness properties of the EM methods and the fast convergence rate properties of GN. Similar results have been reported for the simple case of estimating Gaussian regression models in the econometrics literature [38].

This profiles the relative performance of EM and GN based methods on a rather small sized, and specific problem. To provide further performance insight, this section now turns to the consideration of more realistic sized model structures, and more varied systems.

The first of this next class of simulation examples involves simulated data of length $N = 500$ from one hundred randomly chosen systems of state dimension $n = 5$ and with input and output dimensions $m, p = 2$. The "innovations" form model structure (57) was used to simulate the data, and the input $\{u_t\}$ and corruption $\{v_t\}$ were generated as in the previous examples. On each of these estimation experiments, an initial point for the

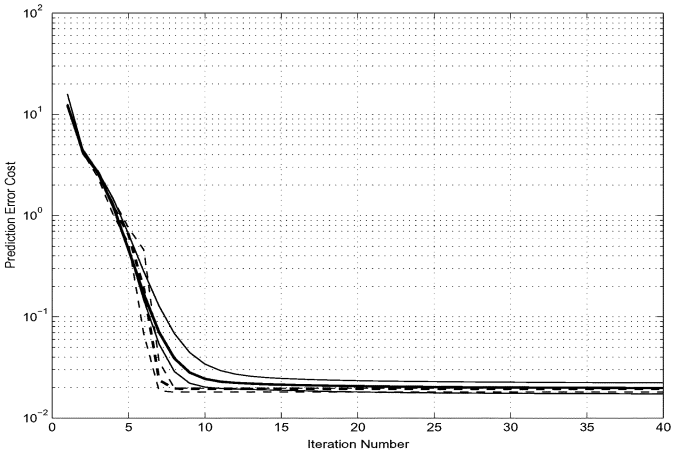

Fig. 3. Average, best, and worst case prediction error cost at each iteration of the EM (solid lines) and Hybrid (dashed lines) algorithms; the thick lines indicate average performance while the thin lines indicate best and worst case performance for each algorithm. Four iterations of the EM algorithm were used to initialize the GN algorithm. Note the average performance lines overlap for the first four iterations, while fast convergence is observed thereafter under the GN algorithm.

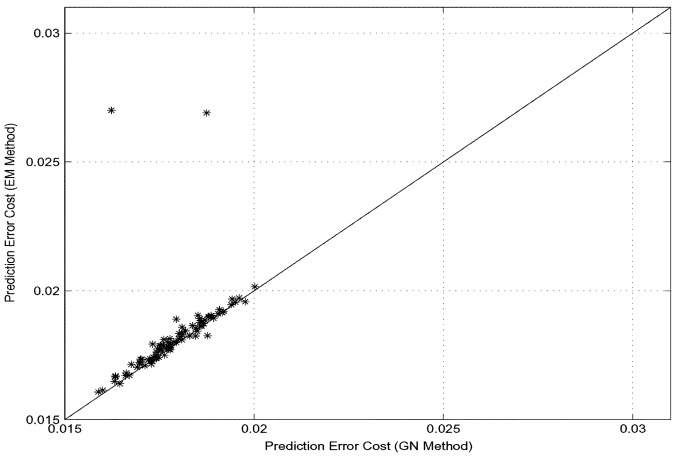

Fig. 4. Comparison of final prediction error cost values for the EM (vertical axis) and GN (horizontal axis) algorithms for randomly chosen systems where $N = 500, n = 5, m = 2$, and $p = 2$. There was one case where the EM algorithm failed to converge to an acceptable objective value and seven cases where the GN algorithms failed to do the same.

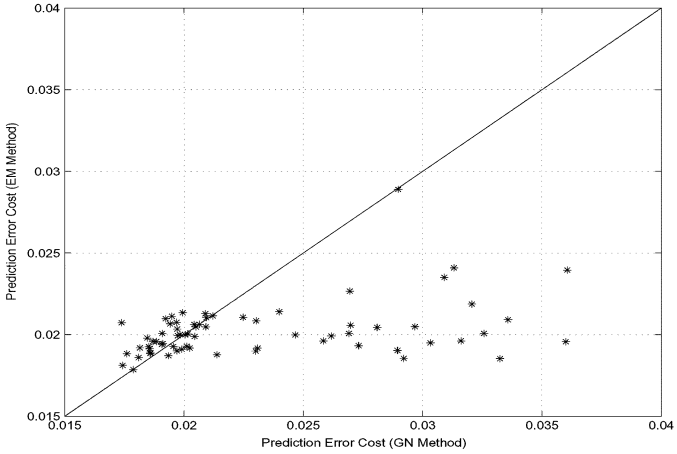

Fig. 5. Comparison of final prediction error cost values for the EM (vertical axis) and GN (horizontal axis) algorithms for randomly chosen systems where $N = 500, n = 10, m = 3$, and $p = 3$. There was one case where the EM algorithm failed to converge to an acceptable objective value and 31 cases where the GN algorithms failed to do the same.

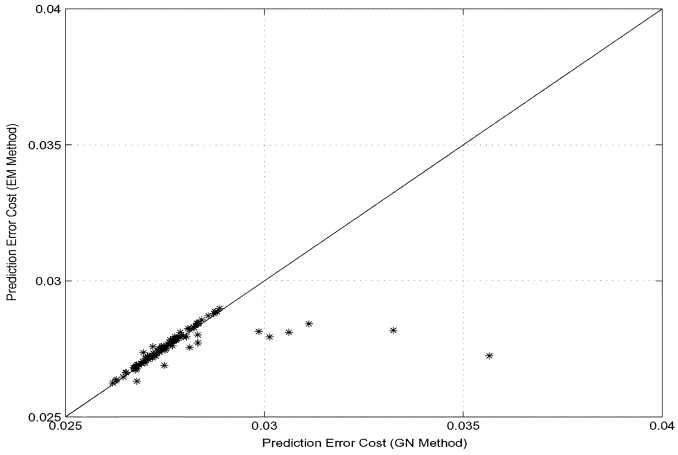

Fig. 6. Comparison of final prediction error cost values for the EM (vertical axis) and GN (horizontal axis) algorithms for randomly chosen systems where $N = 2000, n = 10, m = 3$, and $p = 3$. The EM algorithm converged to an acceptable objective value every time while on four occasions the GN algorithms failed.

EM and GN methods was found by fitting a linear system using N4SID, and setting $F, G = 0, Q = I, R = 0.1I$ and $S = 0$, and K as explained before.

Therefore, each of the one hundred simulation runs involves a different system, a different data realization and a different (data dependent) initialization point. On each of these runs, the EM algorithm was allowed to run for 100 iterations, while the GN algorithm were terminated after either the relative decrease of objective values fell below 10^{-5} or 100 iterations was reached.

The results are profiled in Fig. 4 where each star represents one simulation run, and the location of each star is determined by the final mean square error cost of the EM and GN based estimation methods. Specifically, the y-axis coordinate of the star is the terminal EM cost for a particular data/system realization, and the x-axis coordinate is its associated terminal GN cost. Since the stars are overwhelmingly clustered on the $y = x$ line representing equal performance, the two methods appear to be equally effective. However, what is not illustrated is seven cases where the GN method terminated in a local minima and one case where the EM method did likewise.

Progressing to a scenario of $n = 10'$th order systems with input and output dimensions $m, p = 3$, the results of the same randomly chosen system scenario, but with the maximal iteration count raised to 200, are shown in Fig. 5. Clearly, there is a significant rise in the relative number of cases where EM outperforms GN. If we deem a terminal mean square cost to be "acceptable" if it is less than $1.3 \times \mathrm{Tr}\{R\}$, then by this measure only 69% of the GN terminations were acceptable, compared to 99% of the EM terminations.

In relation to this, it is interesting to note that if the data length is increased to $N = 2000$ samples, then the results shown in Fig. 6 illustrate that the relative performance difference between EM and GN on this large sized system disappears, although in terms of reliability EM was still slightly ahead with 99% acceptable terminations versus 96% for GN.

TABLE I
ILLUSTRATION OF HOW MEASURED RELATIVE FLOP COUNTS PER ITERATION SCALE WITH INCREASING PROBLEM SIZE FOR GN AND ME METHODS. IN ALL CASES THE SYSTEM IS OF OUTPUT DIMENSION $p = 2$. IN THE LEFT TABLE, STATE DIMENSION $n = 2$ AND INPUT DIMENSION $m = 2$. IN THE MIDDLE TABLE, $m = 2$ AGAIN AND DATA LENGTH $N = 500$. IN THE RIGHT TABLE, $n = 2$ AND $N = 500$

N	GN	EM
100	1	1
200	1.6845	1.993
500	3.7614	4.9649
1000	7.2238	9.9181
2000	14.1752	19.8244
5000	35.0304	49.4829
10000	69.8209	98.9541
20000	139.3573	198.138
50000	347.983	495.3274

m	GN	EM
2	1	1
3	1.8567	1.1883
4	3.0184	1.4007
5	4.4878	1.6398
6	6.2563	1.9033
7	8.3874	2.1986
8	11.1029	2.5161
9	13.6935	2.8633
10	17.032	3.2403

n	GN	EM
2	1	1
3	2.7369	2.028
4	6.1577	3.7217
5	12.0547	6.2512
6	22.2475	9.8013
7	38.82	14.5566
8	62.0864	20.6898
9	97.2433	28.389
10	145.1126	37.833

However, it is important to understand that for these larger sized systems, although final estimate quality may be commensurate between the two methods, the amount of computation required to obtain the estimate is far from equivalent. In particular, under matlab 5.3 for which FLOP count can be quantified, the measured *total* FLOP count load to termination was (averaged over the one hundred experiments) thirty times more for a gradient based search than for the EM based method. A difference between minutes and hours per experiment then results.

This theme of considering measured FLOP count, but now per iteration, is continued in Table I. The quantity shown there is relative FLOP count normalized to whatever is required for the smallest case shown in each individual table. The left most table shows how FLOP count scales with data length N for an $n, m, p = 2$ system. The middle table shows relative FLOP scaling versus model order n for $N = 500$ data points, and an $m, p = 2$ system. Finally, the right most table shows relative FLOP scaling versus input dimension m for an $n = 2'$nd order system with $p = 2$ outputs and $N = 500$ data point.

This illustrates the point made in Section VI, that while both EM and GN based methods involve a computational load that scales linearly with data length N, this same load increases faster with system size for GN based methods than for the EM algorithm derived here.

To complete this empirical study, a more substantial size problem is considered. For this purpose, thirty randomly chosen twentieth order, four input, four output systems were estimated using the EM-based methods developed here. Initialization was via a linear N4SID method as before, and the convergence from these initializations is shown in Fig. 7. Note that in all cases, the global minimum of $\mathrm{Tr}\{R\}$ was reached, with acceptable variability in convergence to this minimum. A gradient based method was not profiled on this same problem, largely due to

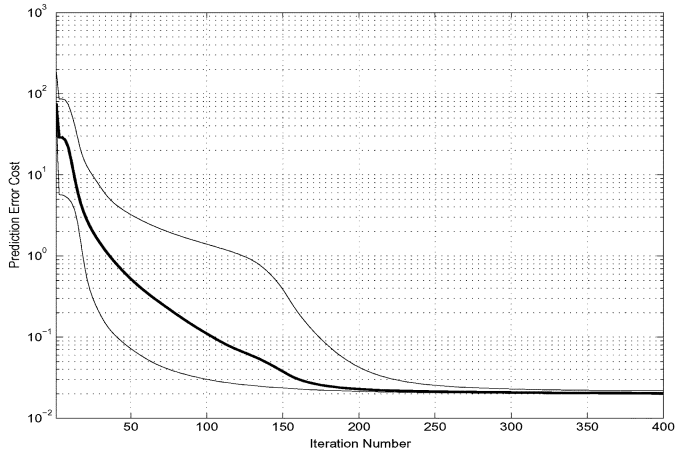

Fig. 7. Average, best, and worst case prediction error cost at each iteration of the EM algorithm deployed on thirty randomly chosen twentieth order four input four output systems. The thick line indicates average performance while the thin lines indicate best and worst case performance.

the fact that (as analyzed in Section VI) the FLOPs required per iteration are $O(mn^2p)$ times greater (with $nm^2p = 1280$ in this case) for a gradient based method relative to the EM-based one, which made the relevant GN simulation impossible.

VIII. CONCLUSION

This paper has derived, analyzed, and illustrated a new EM-based algorithm for maximum likelihood estimation of multivariable bilinear systems, and has profiled it against effective pre-existing methods employing gradient based search. This exposed that, while the EM and gradient search algorithms can both perform well, there are important relative strengths and weaknesses of the two approaches that should be considered.

As illustrated, the new EM based methods derived here have robustness advantages including the avoidance of local minima, and consistent convergence rate performance with respect to data realization, initialization point, and underlying true system. However, this is achieved at the cost of slower convergence rate relative to the best case scenario for gradient based methods. The latter is a strength for gradient based methods, but as was shown, this can come at the expense of lower reliability in terms of capture in local minima, and variability in convergence. Furthermore, for larger sized systems, the computational load (per iteration) for gradient based methods can be very much larger than for the EM based methods derived here.

Since the two approaches are both able to work with the same fully parametrized model structure, handover between them is straightforward. These facts, and further advantages to the EM-based method, such as the ability to accommodate nonsteady-state data records via the straightforward estimation of initial state, suggest the use of hybrid approaches whereby EM-based search is followed by gradient based search.

Further work could usefully examine how this handover should be managed. In relation to this, for readers interested in assessing the ideas in this paper, a suite of MATLAB based routines that implement all the estimation algorithms profiled here is available from the authors upon request [35].

APPENDIX I
PROOF OF LEMMA 4.1

Proof: Repeated application of Bayes' Rule, and use of Markov properties implied by (3) yields

$$\begin{aligned} &p_\theta(Y, X_{N+1}) \\ &\quad = p_\theta(x_{N+1}, y_N | Y_{N-1}, X_N) p_\theta(Y_{N-1}, X_N) \\ &\quad = p_\theta(x_{N+1}, y_N | x_N, u_N) \\ &\qquad \times p_\theta(x_N, y_{N-1} | Y_{N-2}, X_{N-1}) p_\theta(Y_{N-2}, X_{N-1}) \\ &\quad = p_\theta(x_{N+1}, y_N | x_N, u_N) \\ &\qquad \times p_\theta(x_N, y_{N-1} | x_{N-1}, u_{N-1}) p_\theta(Y_{N-2}, X_{N-1}) \\ &\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad (85) \\ &\quad = p_\theta(x_1) \prod_{t=1}^{N} p_\theta(x_{t+1}, y_t \,|\, x_t). \qquad (86) \end{aligned}$$

Furthermore, straightforwardly from (3), (5), and (9)

$$p_\theta(x_1) \sim \mathcal{N}(\mu, P_1) \quad \text{and} \quad p_\theta\left(\begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} \middle| x_t \right) \sim \mathcal{N}(\Gamma z_t, \Pi) \qquad (87)$$

where z_t is defined by (24). Using these densities and excluding terms that are independent of the quantities to be estimated, (86) may be expressed as

$$\begin{aligned} &-2 \log p_\theta(Y, X_{N+1}) \\ &\quad = \log\det P_1 + N \log\det \Pi \\ &\qquad + (x_1 - \mu)^T P_1^{-1} (x_1 - \mu) \\ &\qquad + \sum_{t=1}^{N} (\xi_{t+1} - \Gamma z_t)^T \Pi^{-1} (\xi_{t+1} - \Gamma z_t). \qquad (88) \end{aligned}$$

Applying the conditional expectation operator $\mathbf{E}_{\hat{\theta}_k}\{\cdot \,|\, Y\}$ to both sides of (88) yields (23). ∎

APPENDIX II
PROOF OF LEMMA 4.2

Proof: Equations (35) and (36) are the well-known Rauch–Tung–Striebel recursions for fixed interval Kalman Smoothing of the system (3), (5) (see, for example, [23]) once transformed according to the techniques in [18] to accommodate $S \neq 0$ as follows:

$$\begin{aligned} x_{t+1} &= Ax_t + F(u_t \otimes x_t) + Bu_t + w_t \\ &= Ax_t + F(u_t \otimes x_t) + Bu_t + w_t \\ &\quad + SR^{-1} \underbrace{(y_t - Cx_t - G(u_t \otimes x_t) - Du_t - v_t)}_{=0} \qquad (89) \\ &= [A + (F - SR^{-1}G)(u_t \otimes I_n) - SR^{-1}C] x_t \\ &\quad + (B - SR^{-1}D) u_t + SR^{-1} y_t + \bar{w}_t \qquad (90) \\ y_t &= Cx_t + G(u_t \otimes x_t) + Du_t + v_t \qquad (91) \end{aligned}$$

where now

$$\begin{bmatrix} \bar{w}_t \\ v_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} Q - SR^{-1}S^T & 0 \\ 0 & R \end{bmatrix} \right). \qquad (92)$$

Equation (37) is easily established by multiplying each matrix by its transpose and then comparing submatrices. Expressions (43) and (44) are established in [40, Prop P4.3], except for $M_{N+1\,|\,N} = \bar{A}_N P_{N\,|\,N}$, which is proved as follows. Define $\tilde{x}_{t\,|\,N} \triangleq x_t - \hat{x}_{t\,|\,N}$. Then, (90) implies

$$\begin{aligned} M_{N+1\,|\,N} &\triangleq \mathbf{E}\left\{ \tilde{x}_{N+1\,|\,N} \tilde{x}_{N\,|\,N}^T \right\} \\ &= \mathbf{E}\{ (\bar{A}_N \tilde{x}_{N\,|\,N} + \bar{w}_N) \tilde{x}_{N|N}^T \} \\ &= \bar{A}_N \mathbf{E}\left\{ \tilde{x}_{N\,|\,N} \tilde{x}_{N\,|\,N}^T \right\} = \bar{A}_N P_{N\,|\,N} \end{aligned}$$

where the assumed i.i.d. structure of $\{\bar{w}_t\}$ was used to proceed from the second to the third equality.

The quantities $P_{t\,|\,t}, P_{t\,|\,t-1}$ are well known as being computable via (38) and (39), [24], and the expression for computing $\hat{x}_{t\,|\,t}$ ((47)) is also very well-known [23]. ∎

APPENDIX III
PROOF OF LEMMA 4.3

Proof: Notice that (23) can be partitioned into two parts—one whose terms depend only upon μ and Γ and one whose terms depend solely on Π and P_1. Consider the $\mathrm{Tr}\{\cdot\}$ terms

$$\begin{aligned} &\mathrm{Tr}\{\Pi^{-1}[\Phi - \Psi\Gamma^T - \Gamma\Psi^T + \Gamma\Sigma\Gamma^T]\} \\ &\quad = \mathrm{Tr}\{\Pi^{-1}[(\Gamma - \Psi\Sigma^{-1})\Sigma(\Gamma - \Psi\Sigma^{-1})^T + \Phi - \Psi\Sigma^{-1}\Psi^T]\} \end{aligned}$$

and

$$\begin{aligned} &\mathrm{Tr}\left\{ P_1^{-1} \mathbf{E}_{\theta'}\{(x_1 - \mu)(x_1 - \mu)^T \,|\, Y\} \right\} \\ &\quad = \mathrm{Tr}\left\{ P_1^{-1}[(\hat{x}_{1\,|\,N} - \mu)(\hat{x}_{1\,|\,N} - \mu)^T + P_1] \right\}. \end{aligned}$$

They are clearly (globally) minimized over $\beta \in \Theta$ by (49). Furthermore, the chain rule and Lemma 5.1 provides

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\Pi} \log\det \Pi + \frac{\mathrm{d}}{\mathrm{d}\Pi} \mathrm{Tr}\{\Pi^{-1}(\Phi - \Psi\Sigma^{-1}\Psi^T)\} \\ &\quad = \Pi^{-1} - \Pi^{-1}(\Phi - \Psi\Sigma^{-1}\Psi^T)\Pi^{-1}. \\ &\frac{\mathrm{d}}{\mathrm{d}P_1} \log\det P_1 + \frac{\mathrm{d}}{\mathrm{d}P_1} \mathrm{Tr}\left\{ P_1^{-1} \mathbf{E}_{\hat{\theta}_k}\{(x_1 - \mu)(x_1 - \mu)^T \,|\, Y\} \right\} \\ &\quad = P_1^{-1} - P_1^{-1} P_{1\,|\,N} P_1^{-1}. \end{aligned}$$

These derivatives are clearly zero for the choices of $\Pi = \Phi - \Psi\Sigma^{-1}\Psi^T, P_1 = P_{1\,|\,N}$ and, hence, a stationary point of $\mathcal{Q}(\cdot, \theta)$ over $\eta \in \Theta$. In order to calculate Π so that positive–semidefiniteness and symmetry is ensured, compute the Cholesky factorization

$$\begin{bmatrix} \Sigma & \Psi^T \\ \Psi & \Phi \end{bmatrix} = \begin{bmatrix} \mathcal{R}_{11} & \mathcal{R}_{12} \\ \oslash & \mathcal{R}_{22} \end{bmatrix}^T \begin{bmatrix} \mathcal{R}_{11} & \mathcal{R}_{12} \\ \oslash & \mathcal{R}_{22} \end{bmatrix}.$$

Identifying submatrices on each side we obtain

$$\mathcal{R}_{11}^T\mathcal{R}_{11}=\Sigma \quad \mathcal{R}_{12}^T\mathcal{R}_{11}=\Psi \quad \text{and} \quad \mathcal{R}_{12}^T\mathcal{R}_{12}+\mathcal{R}_{22}^T\mathcal{R}_{22}=\Phi$$

and therefore

$$\begin{aligned}\mathcal{R}_{22}^T\mathcal{R}_{22}&=\Phi-\mathcal{R}_{12}^T\mathcal{R}_{12}\\&=\Phi-\mathcal{R}_{12}^T\mathcal{R}_{11}\left(\mathcal{R}_{11}^T\mathcal{R}_{11}\right)^{-1}\\\mathcal{R}_{11}^T\mathcal{R}_{12}&=\Phi-\Psi\Sigma^{-1}\Psi^T.\end{aligned}$$

■

Appendix IV
Proof of Lemma 4.4

Consider the following definitions:

$$\mathcal{A}\triangleq\frac{1}{N}\sum_{t=1}^{N}\begin{bmatrix}1\\u_t\end{bmatrix}\begin{bmatrix}1\\u_t\end{bmatrix}^T\otimes\hat{x}_{t\,|\,N}\hat{x}_{t\,|\,N}^T$$

$$\mathcal{B}\triangleq\frac{1}{N}\sum_{t=1}^{N}\begin{bmatrix}1\\u_t\end{bmatrix}\otimes\hat{x}_{t\,|\,N}u_t^T\quad\mathcal{C}\triangleq\frac{1}{N}\sum_{t=1}^{N}u_tu_t^T$$

$$\mathcal{D}\triangleq\frac{1}{N}\sum_{t=1}^{N}\begin{bmatrix}1\\u_t\end{bmatrix}\begin{bmatrix}1\\u_t\end{bmatrix}^T\otimes P_{t\,|\,N}$$

which allow Σ to be expressed as

$$\begin{aligned}\Sigma&=\frac{1}{N}\sum_{t=1}^{N}\mathbf{E}_{\hat{\theta}_k}\left\{\begin{bmatrix}x_t\\u_t\otimes x_t\\u_t\end{bmatrix}\begin{bmatrix}x_t\\u_t\otimes x_t\\u_t\end{bmatrix}^T\middle|\,Y\right\}\\&=\begin{bmatrix}\mathcal{A}+\mathcal{D}&\mathcal{B}\\\mathcal{B}^T&\mathcal{C}\end{bmatrix}.\end{aligned}$$

By construction

$$\frac{1}{N}\sum_{t=1}^{N}\begin{bmatrix}\hat{x}_{t\,|\,N}\\u_t\otimes\hat{x}_{t\,|\,N}\\u_t\end{bmatrix}\begin{bmatrix}\hat{x}_{t\,|\,N}\\u_t\otimes\hat{x}_{t\,|\,N}\\u_t\end{bmatrix}^T=\begin{bmatrix}\mathcal{A}&\mathcal{B}\\\mathcal{B}^T&\mathcal{C}\end{bmatrix}\geq 0$$

and, therefore, by virtue of the fact that $\mathcal{C}>0$

$$\begin{aligned}&\begin{bmatrix}\mathcal{A}-\mathcal{B}\mathcal{C}^{-1}\mathcal{B}^T&0\\0&\mathcal{C}\end{bmatrix}\\&\quad=\begin{bmatrix}I&-\mathcal{B}\mathcal{C}^{-1}\\0&I\end{bmatrix}\begin{bmatrix}\mathcal{A}&\mathcal{B}\\\mathcal{B}^T&\mathcal{C}\end{bmatrix}\begin{bmatrix}I&0\\-\mathcal{C}^{-1}\mathcal{B}^T&I\end{bmatrix}\geq 0.\end{aligned}$$

Since $\Pi>0$, Lemma 5.2 proves that $P_{t\,|\,N}>\alpha I$ for some positive constant α and all $t\geq 1$. Therefore

$$\begin{aligned}\mathcal{D}&=\frac{1}{N}\sum_{t=1}^{N}\begin{bmatrix}1\\u_t\end{bmatrix}\begin{bmatrix}1\\u_t\end{bmatrix}^T\otimes P_{t\,|\,N}\\&\geq\frac{1}{N}\sum_{t=1}^{N}\begin{bmatrix}1\\u_t\end{bmatrix}\begin{bmatrix}1\\u_t\end{bmatrix}^T\otimes\alpha I>0\end{aligned}$$

where we have used (55) and the well-known identity $\lambda(A\otimes B)=\lambda(A)\otimes\lambda(B)$ where $\lambda(A)$ denotes the eigenvalues of a matrix A. Finally, then

$$\begin{bmatrix}\mathcal{A}-\mathcal{B}\mathcal{C}^{-1}\mathcal{B}^T+\mathcal{D}&0\\0&\mathcal{C}\end{bmatrix}>0$$

and thus

$$\begin{aligned}&\begin{bmatrix}\mathcal{A}+\mathcal{D}&\mathcal{B}\\\mathcal{B}^T&\mathcal{C}\end{bmatrix}\\&=\begin{bmatrix}I&\mathcal{B}\mathcal{C}^{-1}\\0&I\end{bmatrix}\begin{bmatrix}\mathcal{A}-\mathcal{B}\mathcal{C}^{-1}\mathcal{C}^T+\mathcal{D}&0\\0&\mathcal{C}\end{bmatrix}\begin{bmatrix}I&0\\\mathcal{C}^{-1}\mathcal{B}^T&I\end{bmatrix}>0.\end{aligned}$$

As a consequence, according to (49), (51), and (52), θ_{k+1} is uniquely defined. ■

Appendix V
Technical Lemmata

Lemma 5.1: Suppose $M,N\in\mathbf{R}^{n\times n}$, and M is invertible. Then

$$\begin{aligned}\frac{\partial}{\partial M}\log\det M&=M^{-T}\\\frac{\partial}{\partial M}\mathrm{Tr}\{M^{-1}N\}&=-M^{-T}N^TM^{-T}\\\frac{\partial}{\partial M}\mathrm{Tr}\{MN\}&=N^T.\end{aligned}$$

Proof: See [17]. ■

Lemma 5.2: Consider the system (3), (5) with $\Pi,P_1>0$. Then, there exists a constant $\beta_1>0$ such that

$$P_{t\,|\,N}\geq\beta_1 I\qquad\forall t\geq 1. \tag{93}$$

Proof: Begin by transforming (3), (5) into (90)–(92). Notice that since $\Pi>0$ and $P_{t-1\,|\,t-1}\geq 0$ by construction, [$\bar{A}_t$ and $\bar{Q}$ are defined in (41) and (42)]

$$P_{t\,|\,t-1}=\bar{A}_tP_{t-1\,|\,t-1}\bar{A}_t^T+\bar{Q}\geq\bar{Q}\geq\alpha_1 I\qquad\forall t\geq 1 \tag{94}$$

for some constant $\alpha_1>0$. Therefore, via the matrix inversion lemma,

$$\begin{aligned}P_{t\,|\,t}&=P_{t\,|\,t-1}-P_{t\,|\,t-1}C^T(CP_{t\,|\,t-1}C^T+R)^{-1}CP_{t\,|\,t-1}\\&=\left(P_{t\,|\,t-1}^{-1}+C^TR^{-1}C\right)^{-1}\\&\geq\alpha_2 I\forall t\geq 1\end{aligned} \tag{95}$$

for some constant $\alpha_2>0$. Finally, (95) allows us to bound $P_{t\,|\,N}$ below as follows:

$$\begin{aligned}P_{t\,|\,N}&=P_{t\,|\,t}+J_t(P_{t+1\,|\,N}-P_{t+1\,|\,t})J_t^T\\&=P_{t\,|\,t}-P_{t\,|\,t}\bar{A}_t^T(P_{t+1\,|\,t})^{-1}\bar{A}_tP_{t\,|\,t}\\&\quad+P_{t\,|\,t}\bar{A}_t^TP_{t+1\,|\,t}^{-1}P_{t+1\,|\,N}P_{t+1\,|\,t}^{-1}\bar{A}_tP_{t\,|\,t}&(96)\\&=P_{t\,|\,t}-P_{t\,|\,t}\bar{A}_t^T\left(\bar{A}_tP_{t\,|\,t}\bar{A}_t^T+\bar{Q}\right)^{-1}\bar{A}_tP_{t\,|\,t}\\&\quad+P_{t\,|\,t}\bar{A}_t^TP_{t+1\,|\,t}^{-1}P_{t+1\,|\,N}P_{t+1\,|\,t}^{-1}\bar{A}_tP_{t\,|\,t}&(97)\\&=\left(P_{t\,|\,t}^{-1}+\bar{A}_t\bar{Q}^{-1}\bar{A}_t^T\right)^{-1}\\&\quad+P_{t\,|\,t}\bar{A}_t^TP_{t+1\,|\,t}^{-1}P_{t+1\,|\,N}P_{t+1\,|\,t}^{-1}\bar{A}_tP_{t\,|\,t}&(98)\\&\geq\left(P_{t\,|\,t}^{-1}+\bar{A}_t\bar{Q}^{-1}\bar{A}_t^T\right)^{-1}&(99)\\&\geq\alpha_3 I\qquad\forall t\geq 1&(100)\end{aligned}$$

for some constant $\alpha_3>0$. ■

REFERENCES

[1] R. Baheti, R. Mohler, and H. Spang, "Second order correlation method for bilinear system identification," *IEEE Trans. Autom. Control*, vol. AC-25, no. 6, pp. 1141–1146, Dec. 1980.

[2] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Statist.*, vol. 41, pp. 164–171, 1970.

[3] N. Bergboer, V. Verdult, and M. Verhaegen, "An efficient implementation of maximum likelihood identification of LTI state-space models by local gradient search," in *Proc. 41st IEEE Conf. Decision Control*, Las Vegas, NV, Dec. 2002.

[4] M. Borran and B. Aazhang, "EM-based multiuser detection in fast fading multipath environments," in *Proc. EURASIP*, vol. 8, 2002, pp. 787–796.

[5] J. W. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Trans. Circuits Syst.*, vol. CAS-25, no. 9, pp. 772–781, Sep. 1978.

[6] C. Bruni, G. Dipillo, and G. Koch, "Bilinear systems:An appealing class of "nearly linear" systems in theory and applications," *IEEE Trans. Autom. Control*, vol. AC-19, no. 4, pp. 334–348, Aug. 1974.

[7] P. Caines, *Linear Stochastic Systems*. New York: Wiley, 1988.

[8] H. Chen and J. Maciejowski, "An improved subspace identification method for Bilinear systems," in *Proc. IFAC Symp. System Identification*, 2000, pp. 164–170.

[9] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statist. Soc.*, ser. B, vol. 39, pp. 1–38, 1977.

[10] J. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Upper Saddle River, NJ: Prentice-Hall, 1983.

[11] W. Favoreel, B. De Moor, and P. Van Overschee, "Subspace identification of bilinear systems subject to white inputs," *IEEE Trans. Autom. Control*, vol. 44, no. 6, pp. 1157–1165, Jun. 1999.

[12] F. Fnaiech and L. Ljung, "Recursive identification of bilinear systems," *Int. J. Control*, vol. 45, pp. 453–470, 1987.

[13] M. Gabr and T. S. Rao, "On the identification of bilinear systems from operating records," *Int. J. Control*, vol. 40, pp. 121–128, 1984.

[14] S. Gibson and B. Ninness, "Robust maximum-likelihood estimation of multivariable dynamic systems," *Automatica*, vol. 41, no. 5, pp. 1667–1682, 2005.

[15] G. Golub and C. V. Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1989.

[16] G. Goodwin and A. Feuer, "Estimation with missing data," *Math. Comput. Model. Dyna. Syst.*, vol. 5, pp. 220–244, 1999.

[17] G. Goodwin and R. Payne, *Dynamic System Identification*. New York: Academic, 1977.

[18] G. Goodwin and K. Sin, *Adaptive Filtering Prediction and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[19] N. Gupta and R. Mehra, "Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 774–783, Dec. 1974.

[20] E. Hannan, *Multiple Time Series*. New York: Wiley, 1970.

[21] M. Inagaki and H. Mochizuki, "Bilinear system identification by volterra kernel estimation," *IEEE Trans. Autom. Control*, vol. AC-29, no. 8, pp. 746–749, Aug. 1984.

[22] A. Isaksson, "Identification of ARX models subject to missing data," *IEEE Trans. Autom. Control*, vol. 38, no. 5, pp. 813–819, May 1993.

[23] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York: Academic, 1970.

[24] T. Kailath, A. Sayed, and B. Hassabi, *Linear Estimation*. Upper Saddle River, NJ: Prentice-Hall, 2000.

[25] A. J. Krener, "Bilinear and nonlinear realizations of input-output maps," *SIAM J. Control*, vol. 13, pp. 827–834, 1975.

[26] E. Lehmann, *Theory of Point Estimation*. New York: Wiley, 1983.

[27] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[28] T. McKelvey and A. Helmersson, "System identification using an overparametrized model class—Improving the optimization algorithm," in *Proc. 36th IEEE Conf. Decision and Control*, San Diego, CA, Dec. 1997, pp. 2984–2989.

[29] ——, "A dynamical minimal parametrization of multivariable linear systems and its application to optimization and system identification," in *Proc. of the 14th World Congr. IFAC*, vol. H, Beijing, China, 1999, pp. 7–12.

[30] T. McKelvey, A. Helmersson, and T. Ribarits, "Data driven local coordinates for multivariable linear systems and their application to system identification," *Automatica*, vol. 40, pp. 1629–1635, 2004.

[31] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1996.

[32] S. Meddeb, J. Y. Tourneret, and F. Castanie, "Identification of bilinear systems using Bayesian inference," in *Proc. 1998 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Seattle, WA, May 1998, pp. 1609–1612.

[33] X. Meng and D. van Dyk, "The EM algorithm—An old folk-song sung to a fast new tune," *Journal of the Royal Statistical Society*, ser. B, vol. 59, pp. 511–567, 1997.

[34] I. Misztal and M. Perez-Enciso, "Sparse matrix inversion for restricted maximum likelihood estimation of variance components by Expectation-Maximization," *J. Dairy Sci.*, vol. 76, pp. 1479–1483, 1993.

[35] B. Ninness, A. Wills, and S. Gibson, "The university of newcastle identification toolbox (UNIT)," in *Proc. IFAC World Congr.*, Prague, Czech Republic, Jul. 2005.

[36] J. Nocedal and S. Wright, *Numerical Optimization*. New York: Springer-Verlag, 1999.

[37] W. J. Rugh, *Nonlinear System Theory: The Volterra/Wiener Approach*. Baltimore, MD: Johns Hopkins Univ. Press, 1981.

[38] P. A. Ruud, "Extensions of estimation methods using the EM algorithm," *J. Econometr.*, vol. 49, pp. 305–341, 1991.

[39] R. Shumway, "An approach to time series smoothing and forecasting using the EM algorithm," *J. Time Series Anal.*, vol. 3, pp. 253–264, 1982.

[40] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications*. New York: Springer-Verlag, 2000.

[41] D. Titterington, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions*. New York: Wiley, 1985.

[42] T. Söderström and P. Stoica, *System Identification*. Upper Saddle River, NJ: Prentice-Hall, 1989.

[43] V. Tsoulkas, P. Koukoulas, and N. Kalouptsidis, "Identification of input-output bilinear systems using cumulants," *IEEE Trans. Signal Process.*, vol. 49, pp. 2753–2761, 2001.

[44] P. van Overschee and B. de Moor, "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems," *Automatica*, vol. 30, pp. 75–93, 1994.

[45] V. Verdult, "Nonlinear system identification: A state-space approach," Ph.D. dissertation, Univ. Twente, Twente, The Netherlands, 2002.

[46] V. Verdult, N. Bergboer, and M. Verghaegen, "Maximum likelihood identification of multivariable bilinear state-space systems by projected gradient search," in *Proc. 41st IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 2002.

[47] V. Verdult and M. Verhaegen, "Subspace identification of mimo bilinear systems," in *Proc. European Control Conf.*, Karlsruhe, Germany, 1999.

[48] ——, "Subspace identification of multivariable linear parameter-varying systems," *Automatica*, vol. 38, pp. 805–814, 2002.

[49] V. Verdult and M. Verhaegen, "A kernel method for subspace identification of multivariable bilinear systems," in *Proc. IEEE Conf. Decision and Control*, vol. 4, 2003, pp. 3972–3977.

[50] A. Wills, B. Ninness, and S. Gibson, "On the estimation of fully parametrized state space models via gradient-based search," in *Proc. IFAC World Congr.*, 2005.

[51] C. F. Wu, "On the convergence properties of the EM algorithm," *Ann. Statist.*, vol. 11, pp. 95–103, 1983.

Stuart Gibson was born in Newcastle, Australia. He received Bachelor-level degrees in science and engineering from the University of Newcastle (specializing in mathematics/physics and computer engineering, respectively) in 1992 and 1996. He studied for the Ph.D. degree under the guidance of Professor B. Ninness at the University of Newcastle, and received the Ph.D. degree in 2004.

He is currently a Quantitative Analyst with Lehman Brothers International (Europe), London, U.K. His interests in the signals and systems area include the analysis of irregularly sampled time series, techniques for robustly estimating parameters of continuous-time models, and approaches to modeling with missing data.

Adrian Wills was born in Orange, NSW, Australia. He received the B.E. and Ph.D. degrees from The University of Newcastle, Australia, in 1999 and 2003, respectively.

He has held a postdoctoral research position since June 2003 working in the area of system identification at the University of Newcastle. Other research interests include optimization, model predictive control, and embedded processing for high performance control applications.

Brett Ninness was born in 1963 in Singleton, Australia. He received the B.E., M.E., and Ph.D. degrees in electrical engineering from the University of Newcastle, Australia, in 1986, 1991, and 1994, respectively.

In 1993, he joined the School of Electrical Engineering and Computer Science at the University of Newcastle, where he is currently an Associate Professor. Together with H. Hjalmarsson, he is jointly organizing the 14th IFAC Symposium on System Identification to be held in Newcastle in 2006.

Dr. Ninness is a Member of the Editorial Boards of *Automatica* and the *IEE Control Theory and Applications*, and has published more than 100 articles in the areas of system identification, stochastic signal processing and control. Details of his research and teaching activities are available at www.ee.newcastle.edu.au/brett.