

An Identification Toolbox for Profiling Novel Techniques

Brett Ninness^{a,1} Adrian Wills^b,

^a*School of Electrical Engineering and Computer Science, The University of Newcastle, Australia*

^b*School of Electrical Engineering and Computer Science, The University of Newcastle, Australia*

Abstract

This paper describes a Matlab (or Octave) based software package for the estimation of dynamic systems. It has been developed primarily as a vehicle for profiling novel approaches relative to existing methods within a common software framework in order to streamline comparisons. Key features of the toolbox include simplicity of use (particularly via automated entry of unspecified values), the support of a wide range of scalar and multivariable model structures which include certain nonlinear classes such as bilinear and Hammerstein–Wiener, the ability to handle both time and frequency domain data, the hand optimisation of certain key routines compiled against ATLAS libraries for optimum speed, the use of non-standard optimisation methods based on adaptive subspace gradient search and the Expectation-Maximisation method, and the fact that the toolbox is freely available for non-commercial use.

Key words: Parameter Estimation; System Identification

1 Introduction

This paper details system identification software developed to run under either a Matlab [16] or Octave [1] environment. It offers a suit of methods which have become standard tools within the system identification community. These principally include least-squares and subspace-based techniques in combination with shift operator transfer function and state space model structures. Both time and frequency domain data can be accommodated in these contexts.

More interestingly, the toolbox implements several new approaches which the authors have found to be effective. These include the Expectation Maximisation (EM) algorithm for computation of Maximum Likelihood estimates [10,11], the use of an adaptive Jacobian rank algorithm [26] in gradient based search for least squares estimates and, in some cases, the use of a delta operator model [19]. As well, a range of non-linear model structures including those of bilinear, and Hammerstein–Wiener type are supported. Orthonormal basis model parametrizations [21] are also implemented. Finally,

time domain data at irregularly spaced sampling instants is accommodated via estimation of continuous time models.

With regard to these non-standard aspects, the initial impetus for developing the software suite described here was as a means to efficiently profile the performance of these new methods relative to existing ones, and also as a means for effectively disseminating new methods to other researchers. A purpose of this paper is to announce its free availability (for non-commercial use) to other researchers who may wish to use it for similar purposes by incorporating their own algorithms within the framework it provides [2].

Another key aspect of the toolbox is that while it has been designed as a platform for algorithm development and evaluation by researchers, simplicity of use has been emphasised to provide utility for the non-specialist. For example, as is detailed later, all estimation methods and model structures are encapsulated in one command:

```
g=est(z,m)
```

with automatic entry (to default values) of unspecified entries. Here \mathbf{z} is data, \mathbf{m} determines the chosen model structure and \mathbf{g} is the returned estimate.

A final aspect of interest is that in the case of the Matlab port, several key routines have been hand-coded in C

* This work was supported by the Australian Research Council.

Email addresses: brett.ninness@newcastle.edu.au (Brett Ninness), adrian.wills@newcastle.edu.au (Adrian Wills).

¹ Corresponding author.

and compiled to mex files [16] linked against platform optimised ‘Automatically Tuned Linear Algebra Software’ (ATLAS) [3,25] libraries for maximum speed. These libraries provide interfaces to basic linear algebra routines that are optimised to provide performance comparable with machine-specific hand-tuned libraries. Their use in the toolbox achieves a roughly two-fold speed increase in key routines such as Kalman filtering and smoothing.

In order to profile these developments, the paper will present details of the model structures supported, the estimation methods implemented, and the algorithms employed. A demonstration of how the toolbox is used on some representative examples will conclude the paper.

2 Data Formats

The toolbox is able to perform identification from either time domain or frequency domain data. In the time domain case, the data samples may have been sampled at regularly spaced time intervals or irregular ones, although in the latter case only a restricted set of continuous time model structures can then be subsequently employed in the estimation phase.

In the frequency domain case, the regularity or otherwise of the frequency response measurements has no implications for subsequent toolbox capability.

In both situations, the data may be either scalar or multivariable on either or both of measured input and output, and in the time domain case the time series scenario involving no input measurement is catered for.

3 Model Structures

Depending on the data format, a range of model structures spanning time invariant and time varying, linear and non-linear, parametric and non-parametric are all accommodated as follows.

3.1 Linear Time Invariant Transfer Function

The toolbox supports standard transfer function models of the following form

$$y_t = \sum_{i=1}^m G^i(\rho, \theta) u_t^i + H(\rho, \theta) e_t \quad (1)$$

where u_t^1, \dots, u_t^m are each scalar measured inputs, y_t is a scalar measured output, e_t is a scalar zero mean i.i.d. process of variance $\mathbf{E}\{e_t^2\} = \sigma^2 < \infty$, and $\theta \in \mathbf{R}^n$ is a vector specifying the model parameters.

That is, in the transfer function case, multiple input, single output (MISO) transfer function model structures

are supported, where the elements $G^i(\rho, \theta)$ and $H(\rho, \theta)$ are rational according to

$$G^i(\rho, \theta) = q^{-k_i} \frac{B^i(\rho, \theta)}{A^i(\rho, \theta)}, \quad H(\rho, \theta) = \frac{C(\rho, \theta)}{D(\rho, \theta)} \quad (2)$$

$$A^i(\rho, \theta) = 1 + a_1^i \rho^{-1} + a_2^i \rho^{-2} + \dots + a_{m_a}^i \rho^{-m_a^i}, \quad (3)$$

$$B^i(\rho, \theta) = b_0 + b_1^i \rho^{-1} + b_2^i \rho^{-2} + \dots + b_{m_b}^i \rho^{-m_b^i}, \quad (4)$$

$$D(\rho, \theta) = 1 + d_1 \rho^{-1} + d_2 \rho^{-2} + \dots + d_{m_d} \rho^{-m_d}, \quad (5)$$

$$C(\rho, \theta) = 1 + c_1 \rho^{-1} + c_2 \rho^{-2} + \dots + c_{m_c} \rho^{-m_c}, \quad (6)$$

and k_i is a specified (i.e. not estimated) delay acting on the i 'th input.

Since any of the orders m_a^i, m_b^i, m_c, m_d may be set to zero, then the toolbox can implement any of the common FIR, ARX, ARMAX, ARMA, Output-Error and Box-Jenkins model structures [15].

The symbol ρ above is a time domain operator which may be set to either

$$\rho = q \quad (7)$$

the standard forward shift operator or

$$\rho = \delta \triangleq \frac{q-1}{\Delta} \quad (8)$$

the Euler difference ‘delta’ operator (derivative approximation) [20] with Δ being the underlying sampling period. For future reference in what follows, the inverse operator ρ^{-1} is then defined as either the backward shift operator q^{-1} or the inverse δ (integrator approximation) operator

$$\delta^{-1} = \Delta \sum_{k=0}^{\infty} q^{-k}. \quad (9)$$

In the case of frequency domain data, single-input single-output (SISO) transfer function structures of the following form are supported

$$Y(\omega_k) = G(\gamma_k, \theta) + \varepsilon_k \quad (10)$$

with G as described in (2)-(4). Here, ω_k is the radian per second value at which the frequency response $Y(\omega_k)$ has been measured, and γ_k is one of

$$\gamma_k = e^{j\omega_k \Delta}, \quad \gamma_k = \frac{e^{j\omega_k \Delta} - 1}{\Delta} \quad \text{or} \quad \gamma_k = j\omega_k \quad (11)$$

depending on whether the underlying operator ρ is taken as the shift q^{-1} , Euler difference δ or continuous time derivative d/dt . The term ε_k in (10) accounts for modelling errors.

3.2 Linear Time Invariant State Space

State space model structures are also supported, beginning in the linear time domain case with either of the forms

$$\begin{bmatrix} \rho x_t \\ y_t \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} w_t \\ e_t \end{bmatrix}, \quad (12)$$

or

$$\begin{bmatrix} \rho x_t \\ y_t \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} K \\ I \end{bmatrix} \epsilon_t \quad (13)$$

Here, both $u_t \in \mathbf{R}^m$ and $y_t \in \mathbf{R}^\ell$ may be vectors so that the multiple input, multiple output (MIMO) scenario can be accommodated, and u_t need not be specified in the case of time series modelling, in which case B and D are not included in the model structure. Also, the operator ρ may be specified as either the forward shift operator q or the continuous time derivative d/dt .

Therefore (12) and (13) differ only in the form of the assumed noise contributions to state update and measurement. In (12) the assumed zero mean i.i.d. state noise w_t and measurement noise e_t sequences are separate with joint covariance matrix

$$\Pi \triangleq \text{Cov} \left\{ \begin{bmatrix} w_t \\ e_t \end{bmatrix} \right\} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}. \quad (14)$$

The alternate form (13) is an innovations representation where ϵ_t is an i.i.d. zero mean process with

$$\text{Cov} \{\epsilon_t\} = \bar{R}. \quad (15)$$

In the Gaussian case, (12) and (13) are related in that K and \bar{R} in (13),(15) follow from the solution of the associated algebraic Riccati equation specified by A, C, Q, S, R in (12),(14).

These two cases (12) and (13) are parametrized (respectively) in the toolbox according to

$$\theta \triangleq \text{vec} \left\{ \begin{bmatrix} A & B & Q & S \\ C & D & S^T & R \end{bmatrix} \right\} \quad (16)$$

and

$$\theta \triangleq \text{vec} \left\{ \begin{bmatrix} A & B & K \\ C & D & \bar{R} \end{bmatrix} \right\} \quad (17)$$

where the $\text{vec}\{\cdot\}$ operator is one that forms a vector from a matrix by stacking the columns of the matrix on top of one another. That is, the parametrizations in both cases are full (non minimal) ones in which every element in every matrix is a parameter to be estimated.

A state space model structure can also be employed for estimation from frequency domain observations $\{Y(\omega_k)\}$. In this case the relationship (10) still holds but $G(\gamma_k, \theta)$ may be multivariable according to

$$G(\gamma_k, \theta) = C(\gamma_k I - A)^{-1}B + D, \quad \theta \triangleq \text{vec} \left\{ \begin{bmatrix} A & B \\ C & D \end{bmatrix} \right\}. \quad (18)$$

3.3 Nonlinear Models

In addition to the linear model structures just described, the toolbox also supports important classes of non-linear model structure.

3.3.1 Hammerstein–Wiener

This model structure involves linear time invariant dynamics “sandwiched” between time invariant memory-less non-linearities [6]. That is, it is an extension of (1) which can be described by

$$z_t = \sum_{i=1}^m G^i(\rho, \theta) X^i(u_t^i, \theta) \quad (19)$$

$$y_t = Z(z_t, \theta) + H(\rho, \theta)e_t \quad (20)$$

where $G^i(\rho, \theta)$ and $H(\rho, \theta)$ are as described in (2)-(6), the operator ρ may be either q or δ , and e_t satisfies the same assumptions as profiled following equation (1).

Furthermore, in (19), each of the $X^i(u_t^i, \theta)$ are memory-less non-linearities, as is $Z(z_t, \theta)$ in (20), and the parameters describing them are included in the vector θ that is estimated.

If the component of θ containing this description of the non-linearities is labelled α , then the memory-less functions $X(\cdot, \alpha)$ supported by the toolbox (which are identical to the supported $Z(\cdot, \alpha)$ functions) may be described as follows.

Polynomial

$$X(u_t, \alpha) = \alpha_1 u_t + \alpha_2 u_t^2 + \dots + \alpha_\ell u_t^\ell \quad ; \alpha = [\alpha_1, \dots, \alpha_\ell]^T \quad (21)$$

Saturation

$$X(u_t, \alpha) = \begin{cases} \alpha_3 u_t & ; u_t \in [\alpha_1, \alpha_2] \\ \alpha_2 \alpha_3 & ; u_t > \alpha_2 \\ \alpha_1 \alpha_3 & ; u_t < \alpha_1 \end{cases} \quad ; \alpha = [\alpha_1, \alpha_2, \alpha_3]^T \quad (22)$$

Deadzone

$$X(u_t, \alpha) = \begin{cases} u_t - \alpha_2 ; u_t > \alpha_2 \\ u_t - \alpha_1 ; u_t < \alpha_1 \\ 0 ; u_t \in [\alpha_1, \alpha_2] \end{cases} ; \alpha = [\alpha_1, \alpha_2]^T \quad (23)$$

Note that since the linear dynamics in a Hammerstein–Wiener model can accommodate overall gain, there is no loss of generality by assuming it to be one in the linear regions outside the deadzone. However, this is not the case in the saturation situation, which explains the need for the α_3 component in (22).

Piecewise Linear (Hinging Hyperplane) This non-linearity is described by a set of “hinge” functions of the form

$$X_k(u_t, \alpha_k) = \begin{cases} \alpha_{1,k} + \alpha_{2,k}u_t ; u_t > -\alpha_{1,k}/\alpha_{2,k} \\ 0 ; \text{Otherwise} \end{cases}$$

which are zero up to a “breakpoint”, and linear thereafter. The overall piecewise linear function is then the superposition of (say ℓ of) these hinges:

$$X(u_t, \alpha) = \alpha_0 + \alpha_1 u_t + \sum_{k=1}^{\ell} X_k(u_t, \alpha_k) \quad (24)$$

$$\alpha = [\alpha_0, \alpha_1, \alpha_{1,1}, \alpha_{2,1}, \dots, \alpha_{1,\ell}, \alpha_{2,\ell}]^T. \quad (25)$$

The linear case of $X(u_t, \alpha) = u_t$ is, of course, a final (trivial) possibility which is the default used in the toolbox unless specified otherwise.

3.3.2 Bilinear

A further class of nonlinear models accommodated by the toolbox are MIMO bilinear ones of the following form [9]

$$\begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} = \begin{bmatrix} A & F & B \\ C & G & D \end{bmatrix} \begin{bmatrix} x_t \\ u_t \otimes x_t \\ u_t \end{bmatrix} + \begin{bmatrix} w_t \\ v_t \end{bmatrix}, \begin{bmatrix} K \\ I \end{bmatrix} \epsilon_t \quad (26)$$

where \otimes represents the Kronecker tensor product [7], the noise modelling may be of either form discussed in relation to (12) or (13), and the parametrizations are again full ones according to either

$$\theta \triangleq \text{vec} \left\{ \begin{bmatrix} A & B & F & Q & S \\ C & D & G & S^T & R \end{bmatrix} \right\} \quad (27)$$

and

$$\theta \triangleq \text{vec} \left\{ \begin{bmatrix} A & B & F & K \\ C & D & G & \bar{R} \end{bmatrix} \right\} \quad (28)$$

depending on whether innovations form noise modelling is selected or not. Note that as indicated in (26), only the shift operator is supported for this structure.

3.4 Time Varying Models

Time varying parameter vectors θ_t can be accommodated in situations where the associated model structure can be placed in the linear regressor form

$$y_t = \phi_t^T \theta_t + e_t \quad (29)$$

where the regressor vector ϕ_t may be arbitrary. Two very common cases where this occurs are further supported by the toolbox via combined generation of ϕ_t and subsequent estimation. These are

ARX: This is the model structure (2)-(6) with the choices $D(\rho, \theta) = A(\rho, \theta)$, $m_c = 0$ so that

$$A(\rho, \theta)y_t = q^{-k}B(\rho, \theta)u_t + e_t \quad (30)$$

which leads to the regressor

$$\phi_t = [-y_{t-1}, \dots, -y_{t-m_a}, u_{t-k}, \dots, u_{t-k-m_b}] \quad (31)$$

when $\rho = q$ and to the regressor

$$\phi_t = \left[-\frac{1}{J(\delta)}y_t, -\frac{\delta}{J(\delta)}y_t, \dots, -\frac{\delta^{m_a}}{J(\delta)}y_t, \frac{1}{J(\delta)}u_{t-k}, \frac{\delta}{J(\delta)}u_{t-k}, \dots, \frac{\delta^{m_b}}{J(\delta)}u_{t-k} \right] \quad (32)$$

when $\rho = \delta$. In the above, $J(\delta)$ is a m_a order user chosen “observer polynomial” [19]. If not specified otherwise, the toolbox default is $J(\delta) = (\delta + 1/\Delta)^{m_a}$ which results in (32) being identical (in terms of vector space spanned) to (31).

Orthonormal Basis: This model structure is (2)-(6) with the choices $m_c = m_d = 0$ and $A(\rho, \theta)$ being a fixed user defined choice $A(\rho) = \prod_{i=1}^{m_a} (\rho - \xi_i)$ so that

$$y_t = q^{-k} \frac{B(\rho, \theta)}{A(\rho)} u_t + e_t. \quad (33)$$

Note that in the special case of $\rho = q$ and the choice $\xi_i = 0, i = 0, \dots, m_b$ this is the finite impulse response (FIR) structure. This “fixed denominator” case is termed orthonormal since the implementation used in the toolbox parametrizes (33) according to

$$\frac{B(\rho, \theta)}{A(\rho)} = \sum_{k=1}^{m_a} \theta_k \mathcal{B}_k(\rho) \quad (34)$$

where, in the case of $\rho = q^{-1}$

$$\mathcal{B}_k(q) = \frac{\sqrt{1 - |\xi_k|^2}}{q - \xi_k} \prod_{i=1}^{k-1} \left(\frac{1 - \bar{\xi}_i q}{q - \xi_i} \right) \quad (35)$$

which implies that $\{\mathcal{B}_k(e^{j\omega})\}$ form an orthonormal set on $L_2([-\pi, \pi])$ [21]. When $\rho = \delta$ the functions \mathcal{B}_k are simply those in (35) with the substitution $q = \delta\Delta + 1$.

4 Estimation Methods

With the model structures supported by the toolbox now defined, this section details the range of estimation methods employing these structures that are implemented in the toolbox. These include optimisation-based, projection/realisation-based, and non-parametric approaches.

The presentation in this section is limited to simply specifying available estimation techniques. Details on how they are actually implemented in terms of algorithmic choices are provided in a following section.

4.1 Quadratic Cost Optimisation

A primary estimation method involves solution of the following optimisation problem

$$\hat{\theta}_N \triangleq \arg \min_{\theta \in \mathbf{R}^n} V_N(\theta) \quad (36)$$

where the criterion $V_N(\theta)$ is of the least-squares sort

$$V_N(\theta) \triangleq \text{Trace}\{E(\theta)^\star E(\theta)\}. \quad (37)$$

Here, \star denotes conjugate transpose and

$$E^T(\theta) \triangleq [\varepsilon_1(\theta), \dots, \varepsilon_N(\theta)] \quad (38)$$

is a vector of differences between the observed data and the response of a model parametrized by θ .

In the case of time domain data, the elements of (38) are defined by

$$\varepsilon_t(\theta) \triangleq y_t - \hat{y}_{t|t-1}(\theta) \quad (39)$$

where $y_{t|t-1}(\theta)$ is the (mean square optimal) one step ahead prediction of y_t conditional on experimental observations up to and including time $t - 1$ and based on a model parametrized by θ .

In the case of state-space modelling (including the bilinear case) $\hat{y}_{t|t-1}(\theta)$ is computed by (a square root implementation of) a Kalman filter, while in the transfer

function case $\hat{y}_{t|t-1}(\theta)$ is computed via the steady state Kalman filter in transfer function form; viz.

$$\begin{aligned} \hat{y}_{t|t-1}(\theta) &= H^{-1}(\rho, \theta)Z(z_t, \theta) + [1 - H^{-1}(\rho, \theta)] y_t \\ z_t &= \sum_{i=1}^m G^i(\rho, \theta)X^i(u_t^i, \theta). \end{aligned} \quad (40)$$

In the situation of estimation from frequency domain data, the elements $E(\theta)$ are defined via (10) as

$$\varepsilon_k(\theta) \triangleq Y(\omega_k) - G(\gamma_k, \theta) \quad (41)$$

with $G(\gamma_k, \theta)$ given by (2)-(4) in the case of transfer function modelling, and by (18) in the case of state space descriptions.

In relation to this, it is widely recognised that estimation of continuous time operator $\gamma_k = j\omega_k$ models from frequency domain data is fraught with numerical difficulties. This is due to the often high dynamic ranges of quantities, particularly the measured frequency values ω_k which may vary over several decades.

In the state space modelling case, the toolbox addresses this problem using an effective method proposed in [17] wherein the desired continuous time estimate $G(s, \theta)$ is linked to a discrete time one $\mathcal{G}(q, \theta)$. This is achieved via the bilinear transform $T(q) \triangleq \lambda(q - 1)/(q + 1)$ according to $\mathcal{G}(q, \theta) = G(T(q), \theta)$ where λ is a user defined constant. Moreover an estimate $G(s, \hat{\theta})$ is obtained by in fact first finding a discrete time estimate $\mathcal{G}(q, \hat{\theta})$ (which avoids the afore-mentioned dynamic range problems) using a warped frequency scale in that the measurement $Y(\omega_k)$ is mapped according to

$$Y(\omega_k) \mapsto Y(\tilde{\omega}_k), \quad \tilde{\omega}_k = \frac{2}{\Delta} \tan^{-1} \frac{\omega_k}{\lambda}. \quad (42)$$

Since

$$T(e^{j\tilde{\omega}_k \Delta}) = j\lambda \tan \frac{\tilde{\omega}_k \Delta}{2} = j\omega_k \quad (43)$$

then the discrete time model estimate $\mathcal{G}(e^{j\tilde{\omega}_k \Delta}, \hat{\theta})$ will have identical frequency response as an associated continuous time estimate $G(j\omega_k, \hat{\theta})$ if the latter is formed as

$$G(s, \hat{\theta}) = \mathcal{G}(T^{-1}(q), \hat{\theta}). \quad (44)$$

In the case of state space modelling, this inverse mapping is very simply achieved by substituting $q = T^{-1}(s) = (\lambda + s)/(\lambda - s)$ to derive a mapping from discrete time to continuous time model estimates as

$$\hat{A} := \lambda(\hat{A} + I)^{-1}(\hat{A} - I), \quad \hat{B} := \sqrt{2\lambda}(\hat{A} + I)^{-1}\hat{B} \quad (45)$$

$$\hat{C} := \sqrt{2\lambda}\hat{C}(\hat{A} + I)^{-1}, \quad \hat{D} := \hat{D} - \hat{C}(\hat{A} + I)^{-1}\hat{B}. \quad (46)$$

4.2 Maximum Likelihood

In the situation of Gaussian distributed noise, the toolbox supports estimation according to the Maximum-Likelihood criterion

$$\hat{\theta}_N \triangleq \arg \min_{\theta \in \mathbf{R}^n} L(\theta) \quad (47)$$

where $L(\theta)$ is the log-probability of the observed data. Therefore, in the case of time domain measurements

$$L(\theta) \triangleq \log p(y_1, \dots, y_N | \theta) \quad (48)$$

where $p(y_1, \dots, y_N | \theta)$ is the joint probability density functions of the observations y_1, \dots, y_N . Via Bayes' rule, this depends on the model structure defined by the parameters θ according to

$$p(y_1, \dots, y_N | \theta) = p(y_0 | \theta) \prod_{t=1}^N p(y_t, | y_{t-1}, \dots, y_0, \theta) \quad (49)$$

In the case of transfer function model structures where the steady state Kalman filter (40) is used this implies

$$L(\theta) = \log p(y_0 | \theta) + \sum_{t=1}^N \log p_e(\varepsilon_t(\theta)) \quad (50)$$

where $\varepsilon_t(\theta)$ is given by (39) and $p_e(\cdot)$ is the probability density function (pdf) of e_t . When a state space model and hence a time varying Kalman filter is used to compute $\hat{y}_{t|t-1}(\theta)$, equation (49) leads to

$$L(\theta) = \log p(x_0 | \theta) + \sum_{t=1}^N \log p_e(\varepsilon_t(\theta)) \quad (51)$$

where now $p_e(\cdot)$ is the (non time stationary) pdf of the innovations $\varepsilon_t(\theta)$ and the effect of initial conditions has been encompassed by a pdf on the initial state vector x_0 .

Finally, suppose the available measurements $\{Y(\omega_k)\}$ are in the frequency domain. Then via discrete Fourier transform (DFT) of measurements from the state space model structure (12), and under mild assumptions, the joint log-likelihood of $Y(\omega_1), \dots, Y(\omega_N)$ is [28]

$$L(\theta) = \sum_{k=1}^N \log |P_k| + \left\| P_k^{-1/2} (Y_k - G(\gamma_k)) \right\|^2 \quad (52)$$

where

$$P_k \triangleq CA_k^{-1}QA_k^{-*}C^T + R, \quad A_k \triangleq \gamma_k I - A. \quad (53)$$

4.3 Subspace-Based Projection/Realisation

Subspace-based estimation methods, in which the final estimates do not minimise a (known) criterion are also implemented in the toolbox. In the case of time domain data, these techniques use the state space model structure (13) which implies that for any $r \geq 0$

$$\begin{aligned} \rho^r y_t &= CA^r x_t + D\rho^r u_t + \sum_{k=1}^r CA^{k-1}B\rho^{r-k}u_t \\ &+ R\rho^r \varepsilon_t + \sum_{k=1}^r CA^{k-1}K\rho^{r-k}\varepsilon_t. \end{aligned} \quad (54)$$

Here, and in what follows in this section, ρ may taken as either the forward shift operator q or the delta divided difference operator δ .

The relationship (54) may then be expressed for $r = 0, 1, \dots, f-1$ for some forward horizon length $f \geq 1$ in matrix-vector form as

$$Y_{t,f}^+ = \mathcal{O}_f x_t + \mathcal{U}_f U_{t,f}^+ + \mathcal{E}_f E_{t,f}^+. \quad (55)$$

where

$$\mathcal{O}_f \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{f-1} \end{bmatrix}, \quad \mathcal{U}_f \triangleq \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ \vdots & & \ddots & \vdots \\ CA^{f-2}B & \dots & \dots & D \end{bmatrix}, \quad (56)$$

$$\mathcal{E}_f \triangleq \begin{bmatrix} R & 0 & \dots & 0 \\ CK & R & \dots & 0 \\ \vdots & & \ddots & \vdots \\ CA^{f-2}K & \dots & \dots & R \end{bmatrix}$$

$$Y_{t,f}^+ \triangleq \begin{bmatrix} y_t \\ \rho y_t \\ \dots \\ \rho^{f-1} y_t \end{bmatrix}, \quad U_{t,f}^+ \triangleq \begin{bmatrix} u_t \\ \rho u_t \\ \dots \\ \rho^{f-1} u_t \end{bmatrix}, \quad E_{t,f}^+ \triangleq \begin{bmatrix} \varepsilon_t \\ \rho \varepsilon_t \\ \dots \\ \rho^{f-1} \varepsilon_t \end{bmatrix}. \quad (57)$$

Furthermore, again using the state space model (13), for any backward (past) horizon length $p \geq 0$

$$\rho^p x_t = \tilde{A}^p x_t + \sum_{k=1}^p \tilde{A}^{k-1} \left[\tilde{B} \rho^{p-k} u_t + K R^{-1} \rho^{p-k} y_t \right]$$

where

$$\tilde{A} \triangleq A - KC, \quad \tilde{B} \triangleq B - KD,$$

Therefore, by further defining

$$\mathcal{K}_p \triangleq \left[\tilde{B}, \tilde{A}\tilde{B}, \dots, \tilde{A}^{p-1}\tilde{B}, K, \tilde{A}K, \dots, \tilde{A}^{p-1}K \right],$$

and

$$Z_{t,p}^- \triangleq [\rho^{-1}u_t^T, \dots, \rho^{-p}u_t^T, \rho^{-1}y_t^T, \dots, \rho^{-p}y_t^T]^T$$

x_t may be expressed as

$$x_t = \tilde{A}^p \rho^{-p} x_t + \mathcal{K}_p Z_{t,p}^-. \quad (58)$$

Substituting (58) into (55) provides

$$Y_{t,f}^+ = \mathcal{O}_f \mathcal{K}_p Z_{t,p}^- + \mathcal{U}_f U_{t,f}^+ + \mathcal{E}_f E_{t,f}^+ \mathcal{O}_f \tilde{A}^p x_{t-p}. \quad (59)$$

Provided that $|\lambda_{\max}(\tilde{A})| < 1$ (the noise model is minimum phase) then $\|\tilde{A}\|^p \approx 0$ for large enough p , and hence the first term in (58) in combination with the last term in (59) can be taken as a ‘‘small’’ error term $V_{t,f}$ leading to the following linear regression in β, \mathcal{U}_f

$$Y_{t,f}^+ = \beta Z_{t,p}^- + \mathcal{U}_f U_{t,f}^+ + V_{t,f}$$

where

$$\beta \triangleq \mathcal{O}_f \mathcal{K}_p. \quad (60)$$

Expressing this relationship at all observed data points $t, t+1, \dots, t+N-1$ is then achieved by the matrix equation

$$Y_f = \beta Z_p + \mathcal{U}_f U_f + V_f \quad (61)$$

where

$$Y_f \triangleq [Y_{t,f}^+, Y_{t+1,f}^+, \dots, Y_{t+N-1,f}^+],$$

$$Z_p \triangleq [Z_{t,p}^-, Z_{t+1,p}^-, \dots, Z_{t+N-1,p}^-],$$

and U_f, V_f are defined (respectively) from $U_{t,f}^+, V_{t,f}$ in a manner identical to how Y_f is formed.

Note that in (61), all information (A, B, C, D) parametrizing the dynamics of the state space model (13) is contained in the matrices β, \mathcal{U}_f .

Recognising this, all subspace-based system identification methods supported in the toolbox (and indeed the very great majority of methods existing in the literature) function in a two stage manner in which an estimate $\hat{\beta}$ of β is obtained by linear projection, and the estimates of system matrices A, B, C, D are extracted via $\hat{\beta}$ in a second step, which will be examined in the following section since it is considered an algorithmic detail.

Additionally, a feature common to all methods in the toolbox, and almost all in the literature, is that in recognition of the linear regression formulation (61), a joint estimate of $[\beta, \mathcal{U}_f]$ is found via a least squares criterion

$$[\hat{\beta}, \hat{\mathcal{U}}_f] = \arg \min_{\beta, \mathcal{U}_f} \|Y_f - (\beta Z_p + \mathcal{U}_f U_f)\|_F \quad (62)$$

from which $\hat{\beta}$ is extracted. Here $\|A\|_F$ denotes Frobenius norm [12].

In case of frequency domain data, the toolbox implements the subspace based algorithm presented and analysed in [17] wherein taking the DFT at frequency ω_k of both sides of the state space model (13) implies

$$\Omega_{1,f} \mathcal{Y} = \mathcal{O}_f \mathcal{X} + \mathcal{U}_f \Omega_{m,f} + V. \quad (63)$$

with m being the number of inputs (columns of B) in the model (13), and with the definitions (53) and $X(\omega_k) \triangleq A_k^{-1} B$

$$\mathcal{Y} \triangleq \text{diag}_{1 \leq k \leq N} Y(\omega_k), \quad \mathcal{X} \triangleq [X(\omega_1), \dots, X(\omega_N)] \quad (64)$$

$$\Omega_{m,f} \triangleq \begin{bmatrix} I_m & I_m & I_m & I_m \\ \gamma_1 I_m & \gamma_2 I_m & \dots & \gamma_M I_m \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_1^{f-1} I_m & \gamma_2^{f-1} I_m & \dots & \gamma_M^{f-1} I_m \end{bmatrix} \quad (65)$$

where I_m and $m \times m$ identity matrix. Finally, V depends on the DFT of the innovations ε_t . As per the time domain case, (63) is a linear regression, with all information parametrizing dynamics contained in $\mathcal{O}_f \mathcal{X}$ and \mathcal{U}_f .

Using the over-bar notation $\overline{\mathcal{X}} = [\text{Re}\{\mathcal{X}\}, \text{Im}\{\mathcal{X}\}]$ to denote splitting into real and imaginary components, real valued estimates of $\beta \triangleq \mathcal{O}_f \overline{\mathcal{X}}$ and \mathcal{U}_f are obtained as

$$[\hat{\beta}, \hat{\mathcal{U}}_f] = \arg \min_{\beta, \mathcal{U}_f} \|\overline{\Omega_{1,f} \mathcal{Y}} - (\beta + \mathcal{U}_f \overline{\Omega_{m,f}})\|_F. \quad (66)$$

As in the time domain case $\hat{\beta}$ is extracted from the outcome of (66) and subsequently estimates of the system matrices are formed from it via methods explained in the following section.

4.4 Non-Parametric

This class of methods, as their title implies, do not involve estimation of a parameter vector θ or indeed an assumption of a particular type of model structure. Rather, a frequency response estimate $\hat{G}(\omega)$ at specified frequencies $\{\omega_k\}$ is the deliverable. The toolbox implements two well known methods for achieving this.

4.4.1 Empirical Transfer Function Estimate

The discrete Fourier transform (DFT) at frequency ω rad/s based on N data points u_1, \dots, u_N collected at sampling period Δ is defined here as

$$U_N(\omega) \triangleq \sum_{t=1}^N w_t u_t e^{-j\omega t \Delta} \quad (67)$$

where $\{w_t\}$ is an arbitrary windowing sequence. Using this, the non-parametric empirical transfer function estimate (ETF) [15] is implemented in the toolbox as

$$\widehat{G}(\omega) = Y_N(\omega)/U_N(\omega) \quad (68)$$

where the default window $\{w_t\}$ is of Hamming type, with Bartlett, Hanning and Boxcar together with arbitrary window length being supported.

4.4.2 Blackman-Tukey Estimate

Define (respectively) the sample auto-covariance of the observed input u_t and the sample cross-covariance between u_t and observed output y_t as

$$\widehat{R}_u(\tau) \triangleq \sum_{t=1}^N u_t u_{t-\tau}, \quad \widehat{R}_{yu}(\tau) \triangleq \sum_{t=1}^N y_t u_{t-\tau} \quad (69)$$

where quantities are periodically extended (e.g. $u_\tau = u_{\tau-N}$ for $\tau < 1$) to accommodate negative indices. These allow the formation of estimated auto and cross spectral densities as

$$\widehat{\Phi}_u(\omega) = \sum_{\tau=-N}^N w_\tau \widehat{R}_u(\tau) e^{-j\omega \tau \Delta}, \quad (70)$$

$$\widehat{\Phi}_{yu}(\omega) = \sum_{\tau=-N}^N w_\tau \widehat{R}_{yu}(\tau) e^{-j\omega \tau \Delta} \quad (71)$$

where again, $\{w_t\}$ is an arbitrary window sequence and extension of quantities, but this time symmetric (e.g. $\widehat{R}_u(-\tau) = \widehat{R}_u(\tau)$) is used in the case of negative indices. The so-called ‘‘Blackman–Tukey’’ estimate implemented in the toolbox is provided as

$$\widehat{G}(\omega) = \widehat{\Phi}_{yu}(\omega)/\widehat{\Phi}_u(\omega) \quad (72)$$

with again, a Hamming window as the default. In the time series case of no observed input, the toolbox returns the estimated power spectral density $\widehat{G}(\omega) = \widehat{\Phi}_y(\omega)$ with the last term defined commensurate with how $\widehat{\Phi}_u(\omega)$ is defined via (70),(69).

5 Algorithms

In order to implement the estimation methods profiled in the previous section, the toolbox employs a variety of algorithms which will now be described.

5.1 Gradient Based Search

For the purposes of computing an estimate $\widehat{\theta}_N$ satisfying the least squares criterion (36), the toolbox implements a variety of gradient based search methods. The default technique employing adaptive modulation of Jacobian rank has been developed by the authors [26], but several more standard techniques including Gauss–Newton search, Levenberg–Marquardt regularisation and data driven local co-ordinates (DDL) are also supported.

In order to profile these methods, let

$$J(\theta) = \frac{\partial}{\partial \theta} \text{vec} \{E(\theta)\} \quad (73)$$

be the Jacobian Matrix associated with the estimation error vector (38). The standard Gauss–Newton based search for $\widehat{\theta}_N$ satisfying (36) implemented in the toolbox is a set of iterations $\{\theta_k\}$ starting from an initial guess θ_0 which is then refined according to

$$\theta_{k+1} = \theta_k + \mu p \quad (74)$$

where the search direction p is a solution of

$$[J(\theta_k)^T J(\theta_k) + \lambda I] p = -J(\theta_k)^T E(\theta_k). \quad (75)$$

Here $\lambda > 0$ implies a Levenberg–Marquardt method, while $\lambda = 0$ (the default in the toolbox) leads to a Gauss–Newton update strategy. In both cases, μ is a step length, that while illustrated in (74) as fixed for simplicity of notation, in fact is not. At each iteration k , an initialisation of $\mu = 1$ is set. Then

$$V_N(\theta_{k+1}) < V_N(\theta_k) \quad (76)$$

is tested. If this test fails, $\mu := \mu/2$ is reset and the update (74) is recomputed followed by the test (76) until this test succeeds, or a maximum number of bisections is reached and the search terminates.

Note that in the case of state-space modelling with full parametrizations, the ensuing over-parametrization implies rank deficiency of the Jacobian $J(\theta)$ so that the solution to (75) is not unique in the Gauss–Newton case of $\lambda = 0$.

In this situation the toolbox first employs a Data Driven Local Co-Ordinate [18] re-parametrization in which at iteration k , via singular value decomposition, a matrix P_k with columns orthogonal to the null space of

$J(\theta)$ is found, and used to define a reduced dimension parametrization $\bar{\theta}$ according to

$$\bar{\theta} = P_k \theta. \quad (77)$$

This implies a reduced dimension search direction \bar{p} as a solution of

$$[\bar{J}(\theta_k)^T \bar{J}(\theta_k)] \bar{p} = -\bar{J}(\theta_k)^T E(\theta_k) \quad (78)$$

where

$$\bar{J}(\theta_k) \triangleq J(\theta_k) P_k \quad (79)$$

is the Jacobian associated with the new parametrization (77). This defines a modified Gauss–Newton update strategy of

$$\theta_{k+1} = \theta_k + \mu P_k \bar{p}. \quad (80)$$

However, in cases of poor input excitation, once again, the solution to (78) may not be unique. Indeed, due to this reason or (for example) due to over-modelling, this problem may also occur in the case of transfer function model structures, even though (for the given model order) the parametrizations employed there are minimal.

To deal with these situations, the toolbox employs the update strategy (80) with search direction \bar{p} given by

$$\bar{p} = -\bar{J}^\dagger(\theta_k) E(\theta_k) \quad (81)$$

where $\bar{J}^\dagger(\theta_k)$ is a pseudo-inverse defined as

$$\bar{J}^\dagger(\theta_k) = V_1 S_1^{-1} U_1^T \quad (82)$$

which depends on the singular value decomposition

$$\bar{J}(\theta) = U S V^T = [U_1, U_2] \begin{bmatrix} S_1 & \emptyset \\ \emptyset & \emptyset \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}. \quad (83)$$

The toolbox then chooses the dimension of S_1 adaptively, and in a manner which is coupled to the choice of step length μ . The details cannot be described here due to space restrictions, but are fully profiled in [27].

5.2 Expectation-Maximisation Search

To address the Maximum-Likelihood estimation criterion (47), the toolbox employs the Expectation-Maximisation (EM) algorithm [8] due to its robustness against trapping in local minima, and its modest computational requirements, especially for models of high input-output and/or state dimension [10,11].

An essential aspect of the EM algorithm is the postulate of a so-called complete data set $Z = (Y, X)$, which consists not only of the actual observations $Y = y_1, \dots, y_N$, but also another set of data X , termed ‘missing data’,

that was not observed, and is a key design variable chosen by the user.

In the toolbox, the EM algorithm is only available for the state-space model structure (12), and the missing data X is chosen as the unobserved state history x_1, \dots, x_N . The EM algorithm proceeds from this choice by first applying Bayes’ rule to the joint density $p(Z | \theta)$ to obtain

$$p(Z | \theta) = p(Z|Y, \theta)p(Y | \theta),$$

and therefore

$$L(\theta) \triangleq \log p(Y | \theta) = \log p(X, Y | \theta) - \log p(X|Y, \theta). \quad (84)$$

In this case, with $\mathbf{E}_{\theta'}\{\cdot | Y\}$ denoting expectation with respect to a probability density function determined by the parameters θ' , and conditional upon data Y , then operating on both sides of (84) with $\mathbf{E}_{\theta'}\{\cdot | Y\}$ leads to

$$\begin{aligned} L(\theta) &\triangleq \log p(Y | \theta) = \mathbf{E}_{\theta'}\{\log p(Y | \theta) | Y\} \\ &= \mathcal{Q}(\theta, \theta') - \mathcal{V}(\theta, \theta'), \end{aligned} \quad (85)$$

where

$$\mathcal{Q}(\theta, \theta') \triangleq \mathbf{E}_{\theta'}\{\log p(X, Y | \theta) | Y\} \quad (86)$$

and

$$\mathcal{V}(\theta, \theta') \triangleq \mathbf{E}_{\theta'}\{\log p(X | Y, \theta) | Y\}. \quad (87)$$

Consequently, the difference between the likelihoods associated with any two elements θ and $\hat{\theta}_k$, can be written

$$\begin{aligned} L(\theta) - L(\hat{\theta}_k) &= \left[\mathcal{Q}(\theta, \hat{\theta}_k) - \mathcal{Q}(\hat{\theta}_k, \hat{\theta}_k) \right] + \\ &\quad \left[\mathcal{V}(\hat{\theta}_k, \hat{\theta}_k) - \mathcal{V}(\theta, \hat{\theta}_k) \right]. \end{aligned} \quad (88)$$

where the last difference is the Kullback-Leibler divergence between $p(X|Y, \theta)$ and $p(X | Y, \hat{\theta}_k)$. This has the property [8]

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

with equality if and only if $\log p(X | Y, \theta) = \log p(X | Y, \hat{\theta}_k)$ almost everywhere [10]. Therefore, 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 maximising $\mathcal{Q}(\theta, \hat{\theta}_k)$, which must increase $L(\theta)$ via (88), and then setting $\hat{\theta}_{k+1}$ equal to this maximiser 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 a solution to (47) and updates to a better one $\hat{\theta}_{k+1}$ via

$$(1) \quad \underline{\mathbf{E} \text{ Step}} \quad \text{Calculate:} \quad \mathcal{Q}(\theta, \hat{\theta}_k); \quad (89)$$

(2) M Step

$$\text{Compute: } \hat{\theta}_{k+1} = \arg \max_{\theta} \mathcal{Q}(\theta, \hat{\theta}_k). \quad (90)$$

In the case of Gaussian noise (modulo initial condition effects) the estimation methods (36) and (47) coincide so that this EM algorithm offers an alternative to gradient based search for the computation of (36) which, as examined in detail in [10],[11] offers a particularly robust and reliable alternative, hence the inclusion of the method in the toolbox.

When implementing either of the steps (89) or (90) it is vital to ensure symmetry and non-negative definiteness of any covariance matrices, be they smoothed state covariances involved in computing (89) or estimating Π (recall (14)) in (90). For this purpose, the toolbox employs a square-root update algorithm for the Kalman Smoother associated with (89) and uses a QR decomposition to compute the square root of a Schur complement associated with implementing (90). Full details are available in [10,11].

5.3 Subspace-Based Algorithms

As is well known [12], the optimisation problem (62) has closed form solution

$$\begin{bmatrix} \hat{\beta} \\ \hat{\mathcal{U}}_p \end{bmatrix} = Y_f [Z_p^T, U_f^T] \begin{bmatrix} Z_p Z_p^T & Z_p U_f^T \\ U_f Z_p^T & U_f U_f^T \end{bmatrix}^{-1} \quad (91)$$

and hence via the formula for the inverse of a partitioned matrix [13]

$$\hat{\beta} = Y_f \Pi_{U_f} Z_p^T (Z_p \Pi_{U_f} Z_p^T)^\dagger \quad (92)$$

where \cdot^\dagger denotes Moore-Penrose pseudo-inverse (see (82)) and

$$\Pi_{U_f} \triangleq I - U_f^T (U_f U_f^T)^{-1} U_f. \quad (93)$$

In the interests of numerical efficiency and robustness, the toolbox performs the computation (92) with respect to an orthonormal basis by first computing the *QR* decomposition [12]

$$[U_f^T, Z_p^T, Y_f^T] = [Q_1, Q_2, Q_3] \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & R_{33} \end{bmatrix} \quad (94)$$

where $Q = [Q_1, Q_2, Q_3]$ is an $N \times N$ orthogonal matrix, R_{11} , R_{22} and R_{33} are all upper triangular matrices with

dimensions conformal to those of U_f , Z_p and Y_f , and Q_1 , Q_2 , Q_3 are also chosen conformally. This allows (92) to be efficiently computed as

$$\hat{\beta} = R_{23}^T R_{22} [R_{22}^T R_{22}]^\dagger. \quad (95)$$

It then remains to derive system parameter estimates. For this purpose all subspace-based methods implemented in the toolbox employ a factorisation using the singular value decomposition

$$W_f \hat{\beta} W_p = [U_1, U_2] \begin{bmatrix} S_1 & \emptyset \\ \emptyset & S_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \approx U_1 S_1 V_1^T \quad (96)$$

where in forming the approximation it has been assumed that the singular values in S_1 dominate those in S_2 . This implies via (60), estimates of \mathcal{O}_f and \mathcal{K}_p given as

$$\hat{\mathcal{O}}_f \triangleq W_f^{-1} U_1 (S_1)^{1/2}, \quad \hat{\mathcal{K}}_p \triangleq (S_1)^{1/2} V_1^T W_p^{-1}. \quad (97)$$

The matrices W_f and W_p are user chosen weightings with the following options corresponding to two well known subspace algorithms supported in the toolbox.

N4SID This refers to the ‘‘subspace based system identification’’ (4SID) methods proposed in [24,23] where the choice of

$$W_f = I, \quad W_p = Z_p$$

is made.

CCA Canonical Correlation Analysis is an idea having roots in Mathematical Statistics [5,4] and being principally introduced in a subspace-based estimation context by Larimore [14]. This involves the weightings

$$W_f = (Y_f \Pi_{U_f} Y_f^T)^{-1/2}, \quad W_p \triangleq (Z_p \Pi_{U_f} Z_p^T)^{1/2}. \quad (98)$$

In a final stage, estimates of system matrices A, B, C, D, K, \bar{R} are derived from $\hat{\mathcal{O}}_f$ and (possibly also) $\hat{\mathcal{K}}_p$. Again, there are two distinct methods used in the toolbox depending on whether the N4SID or CCA variant is chosen.

In the N4SID case, since via (58) $x_t = \mathcal{K}_p Z_{t,p}^-$ and via (60) $\beta = \mathcal{O}_f \mathcal{K}_p$ then $\beta Z_p = \mathcal{O}_f X_t$ where $X_t \triangleq [x_t, \dots, x_{t+N-1}]$, and hence the state sequence may be estimated as

$$\hat{X}_t = \arg \min_{X_t} \left\| \hat{\beta} Z_p - \mathcal{O}_f X_t \right\|_F. \quad (99)$$

This can be efficiently solved via QR factorisation as detailed via (94). Using the model structure (13), this

allows estimates of system matrices to be formed as

$$\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D} = \arg \min_{A, B, C, D} \left\| \begin{bmatrix} \rho \widehat{X}_t \\ Y_t \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \widehat{X}_t \\ U_t \end{bmatrix} \right\|_F \quad (100)$$

which again has obvious closed form solution due to the linear regression involved.

In the CCA case, given the definition (56) of the extended observability matrix \mathcal{O}_f , an estimate \widehat{C} of C is taken as the first ℓ (recall, $y_t \in \mathbf{R}^\ell$) rows of $\widehat{\mathcal{O}}_f$. That is,

$$\widehat{C} = [I_\ell, \emptyset] \widehat{\mathcal{O}}_f. \quad (101)$$

By further defining \mathcal{O}_f^+ and \mathcal{O}_f^- as (respectively) \mathcal{O}_f with the first and last ℓ rows removed:

$$\mathcal{O}_f^+ \triangleq [\emptyset, I_{(f-\ell)\ell}] \mathcal{O}_f, \quad \mathcal{O}_f^- \triangleq [I_{(f-1)\ell}, \emptyset] \mathcal{O}_f \quad (102)$$

then $\mathcal{O}_f^- A = \mathcal{O}_f^+$ which suggests the estimate

$$\widehat{A} \triangleq \arg \min_A \left\| \widehat{\mathcal{O}}_f^+ - \widehat{\mathcal{O}}_f^- A \right\|_F = \left[\widehat{\mathcal{O}}_f^- \right]^\dagger \widehat{\mathcal{O}}_f^+. \quad (103)$$

Finally, a simple consequence of the state space representation (13) is that for any $r \geq 0$ and assuming zero initial conditions ($x_{t-r} = 0$)

$$y_t = D u_t + \sum_{k=1}^r C A^{k-1} B \rho^{-k} u_t + \rho^{-r} v_t \quad (104)$$

where v_t depends on ε_t and is zero if $\varepsilon_t = 0$. Therefore, using the identities $\text{vec } AB = (I \otimes A) \text{vec } B = (B^T \otimes I) \text{vec } A$

$$y_t = M_t \theta + v_t \quad (105)$$

where $\theta = \text{vec}\{[B, D]\}$ and

$$M_t \triangleq \begin{bmatrix} u_t^T \otimes I_\ell, & \sum_{k=1}^r [\rho^{-k} u_t]^T \otimes C A^{k-1} \end{bmatrix}. \quad (106)$$

In the CCA case this leads to the estimates for B and D being computed in the toolbox via the linear regression problem

$$\widehat{B}, \widehat{D} = \arg \min_{B, D} \sum_{t=1}^N \|y_t - M_t \theta\|_2^2 \quad (107)$$

which again has well known closed form solution [12].

In both the N4SID and CCA cases, estimates \widehat{Q} and \widehat{R} and \widehat{S} of the covariance matrices in (14) are obtained as the sample covariances obtained when \widehat{X}_t given by

(99) together with estimates of A, B, C, D are substituted into the model (12). Estimates \widehat{K} of the Kalman gain and \widehat{R} of the innovations variance associated with the model structure (13) are then found by solution of an algebraic Riccati equation based on these estimates of A, C, Q, S, R .

Finally, in the case of frequency domain data, an almost identical procedure as for the time domain CCA case is employed. Namely, via the QR factorisation [12]

$$[\overline{\Omega_{m,f}}^T, \overline{\Omega_{1,f}} \mathcal{Y}^T] = [Q_1, Q_2] \begin{bmatrix} R_{11} & R_{12} \\ \emptyset & R_{22} \end{bmatrix} \quad (108)$$

the estimate $\widehat{\beta} = R_{22}^T$ has the same column range space as $\widehat{\beta}$ defined by (66), and hence is used in the subsequent steps (96), (97) with $W_f = W_p = I$ to produce an estimate $\widehat{\mathcal{O}}_f$ of the extended observability matrix defined in (56). From this, estimates \widehat{C} and \widehat{A} in the state space model (13) are found via (101) and (103).

Finally, estimates \widehat{B} and \widehat{D} are formed by solving the linear regression problem

$$\widehat{B}, \widehat{D} = \arg \min_{B, D} \sum_{k=1}^N \left\| Y(\omega_k) - [D + \widehat{C}(\gamma_k I - \widehat{A})^{-1} B] \right\|_F^2. \quad (109)$$

5.4 Recursive Techniques

When employing time varying models in the linear regression form (29), the toolbox provides an associated time varying parameter estimate $\widehat{\theta}_t$ according to

$$\widehat{\theta}_{t+1} = \widehat{\theta}_t + L_t (y_t - \phi_t^T \widehat{\theta}_t) \quad (110)$$

where L_t is a gain vector that may be computed in various ways. One possibility is

$$L_t = \mu \phi_t, \quad \mu \in (0, 1) \quad (111)$$

so that (110) becomes a ‘least mean square’ (LMS) algorithm [22]. Another possibility supported by the toolbox is

$$L_t = P_t \phi_t \quad (112)$$

where P_t satisfies the Riccati equation

$$P_t = \frac{1}{\mu} \left\{ P_{t-1} - \frac{P_{t-1} \phi_t \phi_t^T P_{t-1}}{\mu + \phi_t^T P_{t-1} \phi_t} \right\}; \quad \mu \in (0, 1) \quad (113)$$

initialised with some positive definite P_0 and with the ensuing algorithm being known as ‘Recursive Least Squares’ (RLS) with forgetting factor μ . Finally, for

cases in which the time variation of the parameters θ_t are believed to follow the random walk

$$\theta_{t+1} = \theta_t + \mu w_t \quad (114)$$

where w_t is a stationary zero mean white noise process with $\mathbf{E}\{w_t w_t^T\} = Q$, then the update law

$$L_t = \frac{\mu P_{t-1} \phi_t}{\sigma^2 + \mu \phi_t^T P_{t-1} \phi_t} \quad (115)$$

where P_t satisfies the Riccati equation

$$P_t = P_{t-1} - \mu \frac{P_{t-1} \phi_t \phi_t^T P_{t-1}}{\sigma^2 + \mu \phi_t^T P_{t-1} \phi_t} + \mu Q \quad (116)$$

is supported by the toolbox. It is an instance of the Kalman filter and hence mean-square optimal under Gaussian assumptions.

6 Software Description and Illustration

For the purposes of finding a system estimate from available data, a user of the toolbox need know only one command:

```
g=est(z,m,opt);
```

All arguments, and the returned output \mathbf{g} are structure variables, which (typically) contain individual elements of mixed type.

To be more specific, the structure \mathbf{z} defines the observed data. It must have element $\mathbf{z.y}$ and may have element $\mathbf{z.u}$ which (respectively) are matrices of output and input data, with each column representing a different input or output, and each row a different time sample.

The structure \mathbf{m} defines the model structure to be used. At a minimum, it must contain the entry $\mathbf{m.A}$ which, if an integer, defines a model order, and if a matrix/vector defines an estimate. The structure \mathbf{opt} , defines optional specifications pertaining to the algorithm being used. If there are none the user wishes to dictate, then $\mathbf{g=est(z,m)}$ is a legal command.

To make this concrete, consider the data set shown in figure 1 with measured input stored in vector \mathbf{u} and output in vector \mathbf{y} . Then the following sequence of commands will estimate a fifth order model from the data.

```
>> z.y=y; z.u=u; m.A=5; g=est(z,m);
```

As mentioned, \mathbf{g} is a structure, and in the above example it will contain 30 elements. In order to extract a summary of the properties of the estimate, the `details` command provides

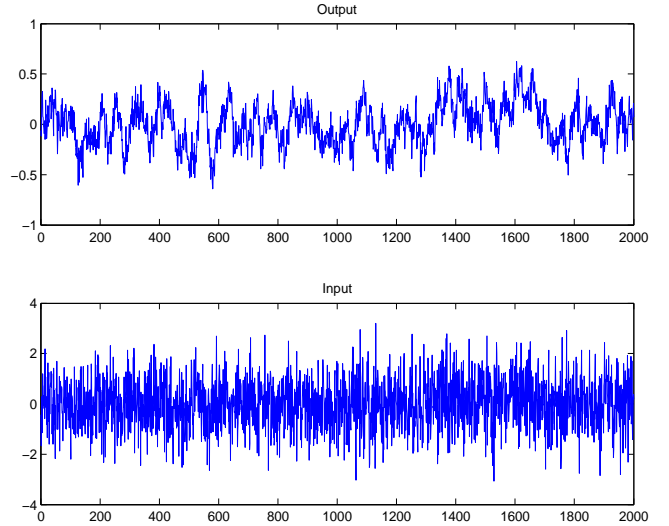


Fig. 1. Observed Input/Output Data.

```
>> details(g)
-----
Details for Estimated Model Structure
-----
Operator used in model      = q
Sampling Period            = 1.000000 seconds
Estimated Innovations Variance = 5.873686e-03
Model Structure Used       = Output Error
Estimation algorithm       = Gauss-Newton search
Input #1 block type        = linear
Output block type          = linear
-----

Input #1 to Output #1 Estimated T/F model + standard devs:
-----
      1      q^-1      q^-2      q^-3      q^-4      q^-5
B = 0.0037 -0.0097  0.0065  0.0067 -0.0107  0.0077
SD= 0.0035  0.0114  0.0217  0.0229  0.0232  0.0233

      1      q^-1      q^-2      q^-3      q^-4      q^-5
A = 1.0000 -2.2342  1.2688  0.3107 -0.3970  0.0558
SD= 0      0.0046  0.0277  0.1196  0.2630  0.2638

delay = 0 samples

Poles at 0.8247*exp(+j0.1085), -0.5015, 0.1782, 0.9178.
```

This illustrates a main point. A philosophy underlying the toolbox is that, in order to maximise utility for the inexperienced, defaults are used as opposed to issuing error messages. In particular, the above indicates that since only an order was specified, the operator type (q), sampling period ($\Delta = 1$), and model order type (Output Error) have all been set as defaults.

In order to assess the quality of this model, a standard model validation test may be performed

```
validate(z,g);
```

This provides the sample-correlation of the error residuals as shown in figure 2(a) and (not shown) sample cross correlation $\hat{R}_{u\varepsilon}(\tau)$ between input and residuals. Clearly, from figure 2(a) there appears to be some undermodelling. In order to illustrate the specification of a Box-Jenkins structure, the following commands specify that a first order noise model should be added to the

model structure ($m.D=1$), that the progress of the Gauss–Newton iterations should be displayed (`opt.dsp=1`) and that this new model should also be validated

```
>> m.D=1; opt.dsp=1; g1=est(z,m,opt);
```

```
=====
START ESTIMATION PROCESS:
Estimating parameters for "BJ" model structure using "q" operator.
INITIALISATION:
Finding initial dynamics model via Steiglitz-McBride...
Best results achieved using 5 iteration(s)
Finding initial noise Model by Hannan-Rissanen...
MAIN SEARCH:
Algorithm: Gradient Based Search
=====
```

Iter#	Cost	G-N Norm	Bisec#	SV#(/13)	SV Tol	Dir
0	5.040e-03	-	-	-	2.500e-05	-
1	5.036e-03	1.804e-03	0	11	2.500e-05	rGN
2	5.036e-03	2.108e-04	0	12	6.250e-06	rGN
3	5.035e-03	2.819e-04	1	13	6.250e-06	rGN
4	5.035e-03	1.068e-04	1	13	6.250e-06	rGN
5	5.035e-03	2.554e-05	1	13	6.250e-06	rGN
6	5.035e-03	8.922e-06	1	13	6.250e-06	rGN
7	5.035e-03	2.848e-06	1	13	6.250e-06	rGN
8	5.035e-03	1.056e-06	0	13	1.563e-06	rGN
9	5.035e-03	2.466e-06	1	13	1.563e-06	rGN
10	5.035e-03	7.715e-07	1	13	1.563e-06	rGN
11	5.035e-03	2.328e-07	1	13	1.563e-06	rGN
12	5.035e-03	7.417e-08	1	13	1.563e-06	rGN
13	5.035e-03	2.415e-08	1	13	1.563e-06	rGN
14	5.035e-03	8.349e-09	1	13	1.563e-06	rGN
15	5.035e-03	3.067e-09	0	13	3.906e-07	rGN
16	5.035e-03	7.458e-09	1	13	3.906e-07	rGN
17	5.035e-03	2.266e-09	1	13	3.906e-07	rGN
18	5.035e-03	7.019e-10	1	13	3.906e-07	rGN
19	5.035e-03	2.228e-10	1	13	3.906e-07	rGN
20	5.035e-03	7.361e-11	1	13	3.906e-07	rGN
21	5.035e-03	2.565e-11	1	13	3.906e-07	rGN
22	5.035e-03	9.554e-12	0	13	9.766e-08	rGN
23	5.035e-03	2.237e-11	1	13	9.766e-08	rGN
24	5.035e-03	6.826e-12	1	13	9.766e-08	rGN
25	5.035e-03	2.120e-12	1	13	9.766e-08	rGN
26	5.035e-03	6.774e-13	1	13	9.766e-08	rGN
27	5.035e-03	2.255e-13	1	13	9.766e-08	rGN
28	5.035e-03	7.943e-14	1	13	9.766e-08	rGN
29	5.035e-03	2.995e-14	0	13	2.441e-08	rGN
30	5.035e-03	6.725e-14	1	13	2.441e-08	rGN
31	5.035e-03	2.057e-14	1	13	2.441e-08	rGN

but we ignore this in order to streamline the presentation. Note in passing, that if $M.C = 1$ had been specified the toolbox would have assumed that $M.A = M.D$ and hence an ARMAX structure was implied. This, in turn could have been overridden by the specification `M.type='bj'` to ensure a Box–Jenkins structure with first order noise model. Other legal options for `M.type` are `fir`, `arx`, `armax`, `oe` and `ss` (state space).

To illustrate a more complex example, consider the MISO data shown in figure 3. This comes from a system

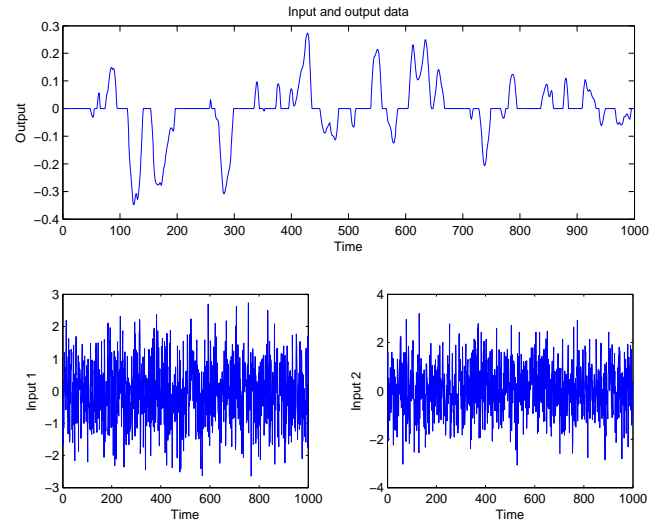


Fig. 3. Observed MISO Input/Output Data from Hammerstein System.

with saturation and dead-zone non-linearities on inputs 1 and 2 respectively, that then pass through linear systems of order 4 and 3 respectively with the output then passing through a further dead-zone non-linearity.

In order to estimate a model structure of this form we proceed as follows

```
>> z.y=y; z.u=u;
>> m.A=[4;3]; m.B=[3;2]; m.delay=[1;1];
>> m.in(1).type='hinge';
>> m.in(2).type='deadzone';
>> m.out.type='deadzone';
>> m.out.upper=0.1; m.out.lower=-0.1;
>> g=est(z,m);
```

Note that now two model orders $m_a^1 = 4$, $m_a^2 = 3$ are specified, which are different, for the two linear components associated with the two inputs. It is important that the different orders are specified as a column vector. If they had been a row vector, the toolbox would interpret them not as orders, but an initial estimate of a polynomial denominator and/or numerator in a SISO model.

In the above, a time delay of one sample is specified for both of the inputs. Furthermore, it is specified that a

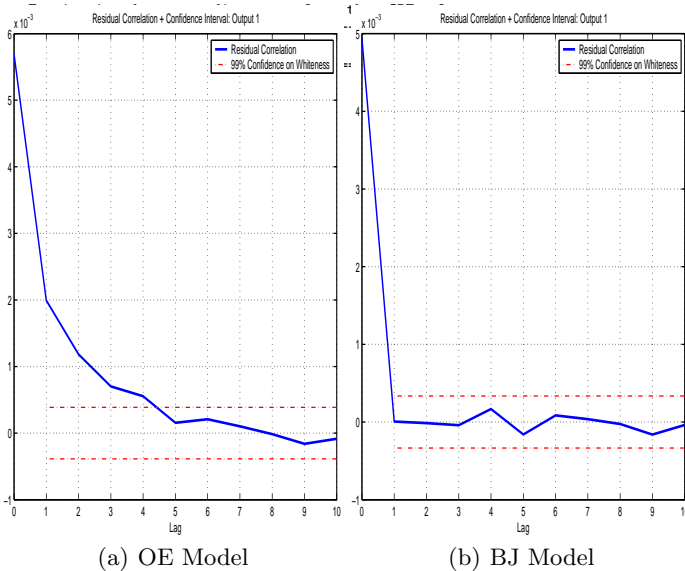


Fig. 2. Model Validation

The ensuing validation results are shown in figure 2(b). Of course, in practise, data other than that used for estimation should be employed for this validation purpose,

piecewise linear (hinging plane) structure should be used to model the non-linearity on the first input, while for the second, prior knowledge that it is of dead-zone type is employed. Finally, a dead-zone non-linearity is specified for the output, with initial estimates for the dead-zone region as $[-0.1, 0.1]$. If only one of `m.out.upper` and `m.out.lower` were specified, the toolbox would have assumed that one was the negative of the other so that a symmetric non-linearity was implied.

As before, the results of this estimation experiment can be summarised via use of the `details` command.

```
>> details(g)
-----
Details for Estimated Model Structure
-----
Operator used in model      = q
Sampling Period            = 1.000000 seconds
Estimated Innovations Variance = 6.260299e-05
Model Structure Used       = Not Specified
Estimation algorithm       = Gauss-Newton search
Input #1 block type        = saturation
Input #2 block type        = deadzone
Output block type          = deadzone
-----

Input #1 to Output #1 Estimated T/F model + standard devs:
-----
      1      q^-1      q^-2      q^-3      q^-4
B = -0.0001  0.0049 -0.0045  0.0023 -0.0025
SD=  0.0001  0.0005  0.0015  0.0024  0.0026

      1      q^-1      q^-2      q^-3      q^-4
A = 1.0000 -3.3660  4.2352 -2.3630  0.4939
SD=  0      0.0002  0.0016  0.0059  0.0172

delay = 1 samples

Poles at 0.7430*exp(+j0.1324), 0.9124, 0.9806.
-----

Input #2 to Output #1 Estimated T/F model + standard devs:
-----
      1      q^-1      q^-2      q^-3
B =  0.0022  0.0038  0.0024  0.0023
SD=  0.0006  0.0016  0.0024  0.0025

      1      q^-1      q^-2      q^-3
A = 1.0000 -2.1180  1.4118 -0.2839
SD=  0      0.0008  0.0049  0.0279

delay = 1 samples

Poles at 0.3732, 0.8527, 0.8921.
-----

Input Non-linearity Parameters and standard deviations:
-----
Input block #1 of type saturation has estimates and standard dev:

upper limit = 0.8412, sd = 0.0035
lower limit = -0.7523, sd = 0.0065

Input block #2 of type deadzone has estimates and standard dev:

upper limit = 0.5730, sd = 0.0055
lower limit = -0.4649, sd = 0.0063
-----

Output Non-linearity Parameters and standard deviations:
-----
Output block of type deadzone has estimates and standard dev:

upper limit = 0.1291, sd = 0.0033
lower limit = -0.1218, sd = 0.0037
```

By way of information, the true noise variance was $\sigma^2 = 10^{-4}$, the true dead-zone region on input 2 was (lower, then upper limit) $[-0.5, 0.6]$. Furthermore, the above hinge parametrization corresponds to a dead-zone with

limits $[-0.49, 0.59]$ while the underlying true one was $[-0.8, 0.9]$. The true dead-zone on the output had limits of $[-0.12, 0.13]$.

As a final example, we consider a MIMO bilinear state-space model, where the true underlying system has order $n = 2$, $m = 2$ inputs and $p = 2$ outputs, and is given by the structure (26)–(27) with the choices made in [9] of

$$A = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \quad (117)$$

$$D = I_2, \quad F = \begin{bmatrix} 0.6 & 0 & 0.2 & 0 \\ 0 & 0.4 & 0 & 0.5 \end{bmatrix}, \quad G = 0. \quad (118)$$

This system was simulated with input $\{u_t\}$, state noise ω_t and measurement corruption $\{v_t\}$ being white random processes distributed as $\omega_t \sim \mathcal{N}(0, 0)$, $v_t \sim \mathcal{N}(0, 0.01 \times I)$, $u_t \sim \mathcal{N}(0, I)$ together with $N = 500$ ensuing data samples being used for identification. For this structure and data combination we employed an Expectation-Maximisation (EM) algorithm set via (`opt.alg='em'`) to estimate the parameter values as follows:

```
>> m.A=2; m.type='bilinear'; opt.dsp=1; opt.alg='em';
>> g=est(z,m,opt);

=====
START ESTIMATION PROCESS:
Estimating parameters for "BILINEAR" model structure using "q" operator.
INITIALISATION:
Finding initialisation for estimate of Dynamics via Subspace ID...
MAIN SEARCH:
Algorithm: Expectation-Maximisation
-----
Iter#      PE-Cost      Log Likelihood
-----
1          3.750e+00      -1.484e+03
2          3.623e+00      -1.438e+03
3          3.526e+00      -1.403e+03
4          3.426e+00      -1.364e+03
5          3.325e+00      -1.324e+03
6          3.221e+00      -1.279e+03
7          3.115e+00      -1.231e+03
8          3.006e+00      -1.177e+03
9          2.894e+00      -1.117e+03
10         2.778e+00      -1.048e+03
.
.
.
190        1.998e-02      3.608e+03
191        1.998e-02      3.608e+03
192        1.998e-02      3.608e+03
193        1.998e-02      3.608e+03
194        1.998e-02      3.608e+03
195        1.997e-02      3.608e+03
196        1.997e-02      3.608e+03
197        1.997e-02      3.608e+03
198        1.997e-02      3.608e+03
199        1.997e-02      3.608e+03
200        1.997e-02      3.608e+03
-----
Termination due to iteration count > OPT.maxit
-----
END ESTIMATION PROCESS
=====

>> validate(z,g);
```

As indicated in the output listed above, the algorithm begins with a subspace estimate and then proceeds with

the EM algorithm, which updates the display with prediction error and log-likelihood values for each iteration. The validation results are shown in Figure 4 and indicate that the residuals appear to be relatively uncorrelated. Note that a specification (`opt.alg='gn'`) would

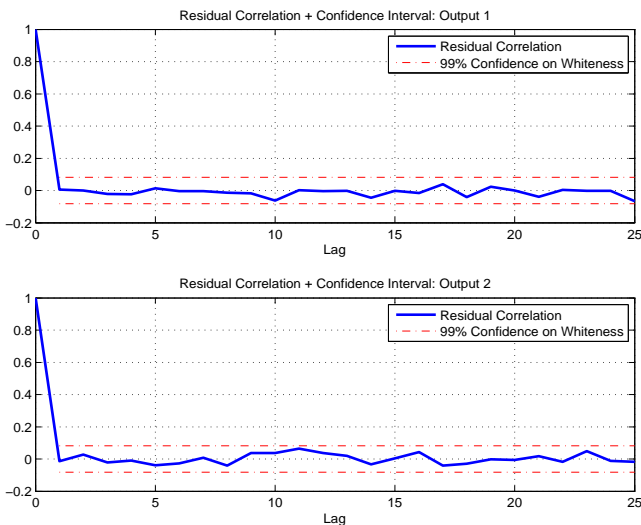


Fig. 4. Validation test for 2-state bilinear example.

have accommodated this same bilinear model structure but forced the employment of a Gauss-Newton derived gradient-based search.

Further features of the toolbox that are not profiled here in the interests of brevity, but are nevertheless worth mentioning are:

- An estimated model structure \mathbf{g} may be used to specify the model structure \mathbf{m} in cascaded estimation experiments;
- Frequency domain plots, complete with error bounds may be simply generated via the `showbode(g)` and `shownyq(g)` commands;
- A delta operator parametrization may be specified by setting `m.op='d'`;
- Finally, online help for all functions is available. In particular `help est` gives a catalogue of all options available.

7 Conclusion

This paper has detailed the model structures, identification methods, and numerical algorithms employed in a system identification toolbox developed as a platform for evaluating new approaches relative to existing methods.

It is freely available (for non-commercial purposes) via [2] for use by other researchers in streamlining algorithm development and profiling.

Time Domain Data							
Model Type		Algorithm			Operator		
		GN	EM	SID	q	δ	s
Linear	TF	✓	×	×	✓	✓	×
	SS	✓	✓	✓	✓	✓	✓
Non-Linear	H/W TF	✓	×	×	✓	✓	×
	Bilin. SS	✓	✓	×	✓	×	×
Frequency Domain Data							
Linear	TF	✓	×	×	✓	✓	✓
	SS	✓	✓	✓	✓	✓	✓

Table 1

Summary of supported model structures and employed algorithms. NB. TF=Transfer Function, SS=State Space, H/W=Hammerstein-Wiener, GN=Gradient Based Search, SID= Subspace Identification.

While, as described here, the toolbox currently provides a relatively comprehensive list of features, further work is planned to address the model structure/algorithm combinations shown as currently unsupported in Table 1. In most cases (eg. EM for transfer function models) this will involve developing new underlying theory and algorithm design.

References

- [1] www.octave.org.
- [2] www.sigpromu.org/staff/idtoolbox.
- [3] *Automatically tuned linear algebra software (atlas)*, <http://math-atlas.sourceforge.net>, (2006).
- [4] H. AKAIKE, *Markovian representation of stochastic processes by canonical variables*, SIAM Journal of Control, 13 (1975), pp. 162–173.
- [5] H. AKAIKE, *Stochastic theory of minimal realisation*, IEEE Transactions on Automatic Control, AC-19 (1974), pp. 667–674.
- [6] E.-W. BAI, *An optimal two stage identification algorithm for hammerstein-wiener nonlinear systems*, Automatica, 34 (1998), pp. 333–338.
- [7] J. W. BREWER, *Kronecker products and matrix calculus in system theory*, IEEE Transactions on Circuits and Systems, 25 (1978), pp. 772–781.
- [8] A. DEMPSTER, N. LAIRD, AND D. RUBIN, *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society, Series B, 39 (1977), pp. 1–38.
- [9] W. FAVOREEL, B. DE MOOR, AND P. VAN OVERSCHEE, *Subspace identification of bilinear systems subject to white inputs*, IEEE Trans. Automat. Control, 44 (1999), pp. 1157–1165.
- [10] S. GIBSON AND B. NINNESS, *Robust maximum-likelihood estimation of multivariable dynamic systems*, Automatica, 41 (2005), pp. 1667–1682.
- [11] S. GIBSON, A. WILLS, AND B. NINNESS, *Maximum-likelihood parameter estimation of bilinear systems*, IEEE Transactions on Automatic Control, 50 (2005), pp. 1581–1596.

- [12] G. GOLUB AND C. V. LOAN, *Matrix Computations*, Johns Hopkins University Press, 1989.
- [13] G. GOODWIN AND R. PAYNE, *Dynamic System Identification*, Academic Press, 1977.
- [14] W. LARIMORE, *Canonical variate analysis in identification, filtering and adaptive control*, in Proceedings of the 29th IEEE Conference on Decision and Control, Hawaii, 1990, pp. 596–604.
- [15] L. LJUNG, *System Identification: Theory for the User, (2nd edition)*, Prentice-Hall, Inc., New Jersey, 1999.
- [16] MATHWORKS, *MATLAB Users Guide, Version 7*, The Mathworks, 2004.
- [17] T. MCKELVEY, H. AKÇAY, AND L. LJUNG, *Subspace-based multivariable system identification from frequency response data*, IEEE Transactions on Automatic Control, 41 (1996), pp. 960–979.
- [18] T. MCKELVEY, A. HELMERSSON, AND T. RIBARITS, *Data driven local co-ordinates for multivariable linear systems and their application to system identification*, Automatica, 40 (2004), pp. 1629–1635.
- [19] R. MIDDLETON AND G. GOODWIN, *Digital Estimation and Control: A Unified Approach*, Prentice-Hall, Inc., New Jersey, 1990.
- [20] ———, *Digital Estimation and Control: A Unified Approach*, Prentice-Hall, Inc., New Jersey, 1990.
- [21] B. NINNESS AND F. GUSTAFSSON, *A unifying construction of orthonormal bases for system identification*, IEEE Transactions on Automatic Control, 42 (1997), pp. 515–521.
- [22] V. SOLO AND X. KONG, *Adaptive Signal Processing Algorithms*, Prentice Hall, 1995.
- [23] P. VAN OVERSCHEE AND B. DE MOOR, *N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems*, Automatica, 30 (1994), pp. 75–93.
- [24] P. VAN OVERSCHEE AND B. D. MOOR, *Subspace Identification for Linear Systems*, Kluwer Academic Publishers, 1996.
- [25] R. C. WHALEY, A. PETITET, AND J. J. DONGARRA, *Automated empirical optimization of software and the ATLAS project*, Parallel Computing, 27 (2001), pp. 3–35. Also available as University of Tennessee LAPACK Working Note #147, UT-CS-00-448, 2000 (www.netlib.org/lapack/lawns/lawn147.ps).
- [26] A. WILLS AND B. NINNESS, *On gradient-based search for multivariable system estimates*, in Accepted for publication in the Proceedings of the IFAC World Congress, Prague, July 2005.
- [27] ———, *On gradient based search for multivariable system estimates*, IEEE Transactions on Automatic Control, (To appear, 2007).
- [28] A. WILLS, B. NINNESS, AND S. GIBSON, *Robust maximum likelihood estimation from frequency domain data*, in Submitted to European Control Conference, ECC07, 2007.