Linear MIMO Model Identification using an Extended Kalman Filter

Matthew C Best

***Abstract*—** Linear Multi-Input Multi-Output (MIMO) dynamic models can be identified, with no a priori knowledge of model structure or order, using a new Generalised Identifying Filter (GIF). Based on an Extended Kalman Filter, the new filter identifies the model iteratively, in a continuous modal canonical form, using only input and output time histories. The filter's self-propagating state error covariance matrix allows easy determination of convergence and conditioning, and by progressively increasing model order, the best fitting reduced-order model can be identified. The method is shown to be resistant to noise and can easily be extended to identification of smoothly nonlinear systems.

***Keywords*—** System Identification, Kalman Filter, Linear Model, MIMO, Model Order Reduction.

I. Introduction

There are many papers exploring MIMO dynamic model system identification, with most broadly falling into categories of those employing neural networks (NN) and genetic algorithms, frequency domain methods, probabilistic approaches and in a few cases Kalman filtering and recursive least squares. Numerous examples of NN achieve excellent identified model performance, but in a black-box format which gives no insight into the plant dynamics. They typically employ high numbers of tuned parameters, with associated concerns over applicable domains of operation and parameter conditioning. Classical alternatives designed to address conditioning within a smaller parameter set include the well-known references on system identification (eg [1]) and papers using complicated statistical probability methods, eg [2,3]. While effective at solving theoretical and complex practical cases for identification, these are mathematically impenetrable to most engineers. Simpler processes, using frequency domain identification of SISO models in combinations are explored by many – eg [4,5] and these can be more effective at replicating higher frequency modes. However, translation into accurate time domain models is not always easily achieved and appropriate combination of the SISO cases can be problematic.

In this paper the well-known Kalman filter is applied to system identification. It too has found uses previously in this field, but in earlier research by the author [6,7] and in literature searches [8,9] Kalman filters have previously only been effective in identifying a subset of unknown parameters in an existing known model structure. Here we address many of the concerns above, identifying a whole model of unknown structure and minimum parameter set using an iterative time-domain approach. One previous paper operates in a similar way, using a recursive least-squares method [10] but this assumes a discrete model of known order. In this new method, the most appropriate model order can also be determined. Ultimately the best conditioned, lowest order model is identified in modal canonical form, which has the added advantage of revealing the most significant system eigenvalues.

M. C. Best is with the Department of Aeronautical and Automotive Engineering at Loughborough University, UK (phone +44-1509-227209; email: m.c.best@lboro.ac.uk).

This work was supported by Jaguar Land Rover and the UK-EPSRC grant EP/K014102/1 as part of the jointly funded Programme for Simulation Innovation

II. The Generalised Identifying Filter (GIF)

The standard Extended Kalman Filter (EKF) operates on nonlinear system and sensor models **f** and **h**, which relate the state vector **x**, measured sensor set **y**, known inputs **u** and model parameters **θ** at any instant k according to

$$\dot{\mathbf{x}}_k = \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta}_k\right) + \boldsymbol{\omega}_k \tag{1}$$

$$\mathbf{y}_k = \mathbf{h}\left(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta}_k\right) + \boldsymbol{\upsilon}_k \tag{2}$$

ω represents state propagation and modelling error, **υ** is the sensor error, and an optimal filter can be derived using estimates, or expectations of the error covariance matrices :

$$\mathbf{Q}_k = E\left(\boldsymbol{\omega}_k \boldsymbol{\omega}_k^T\right),\ \mathbf{S}_k = E\left(\boldsymbol{\omega}_k \boldsymbol{\upsilon}_k^T\right) = \mathbf{0},\ \mathbf{R}_k = E\left(\boldsymbol{\upsilon}_k \boldsymbol{\upsilon}_k^T\right) \tag{3}$$

The EKF also requires model Jacobians to be evaluated at each time step, defined

$$\mathbf{F}_k = \left.\frac{\partial \mathbf{f}\left(\mathbf{x}, \mathbf{u}_k, \boldsymbol{\theta}_k\right)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k}$$

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}\left(\mathbf{x}, \mathbf{u}_k, \boldsymbol{\theta}_k\right)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k} \tag{4}$$

Examples which employ and detail the full IEKF are given in [6,7]. In this paper we seek to identify a linear model

$$\dot{\mathbf{x}}_k = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k \tag{5}$$

Through recorded time-histories of the (single or multiple) input(s) **u** and single or multiple output(s) **y** only. We do not know, or seek to impose any physical structure in the model, so cannot attribute physical meaning to the states, though we will need to decide on the model order, and hence the number

of states.

In order to prescribe an appropriate, minimal number of parameters in the model, a suitable generic structure must be imposed on the model matrices. This could be any structure which assigns suitably independent states in a sparse system (**A**) matrix – ie a canonical form – and we will see advantages in prescribing a modal canonical form. This defines **A** in terms of the system eigenvalues, a choice which delivers dynamic information about the identified model and which also ensures identification of a well conditioned model. An appropriate prescription for a 5 state case is

$$\mathbf{A}=\begin{bmatrix} \sigma_1 & \omega_1 & 0 & 0 & 0 \\ -\omega_1 & \sigma_1 & 0 & 0 & 0 \\ 0 & 0 & \sigma_2 & \omega_2 & 0 \\ 0 & 0 & -\omega_2 & \sigma_2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_3 \end{bmatrix} \quad \mathbf{B}=\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \\ b_{51} & b_{52} \end{bmatrix} \qquad (6)$$

$$\mathbf{C}=\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} \\ c_{21} & c_{22} & c_{23} & c_{24} & c_{25} \end{bmatrix} \quad \mathbf{D}=\begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}$$

and this is expanded or contracted in an obvious way for a different number of inputs / outputs or a different choice of model order (number of states). Note that the eigenvalues are expected in complex conjugate pairs, $\lambda = \sigma + j\omega$ where possible; this is a sensible first assumption, but if initial results show some pairs with zero eigenfrequency it is easy to modify the structure to introduce more real eigenvalues.

One further constraint is then necessary, which can be applied in a number of ways. As written above, the excitation of each state by the inputs is scaled by the **B** matrix, but each output is then scaled again by the **C** matrix. Intuitively, if one considers a SISO case, only the **B** or the **C** parameters need to be uniquely identified to achieve a fully constrained system; one set of 5 parameters can be set to 1. It turns out that the required constraint for any *n* state model is to constrain exactly *n* of the *c* or *b* parameters regardless of the number of inputs or outputs. The best choice is to leave **B** to be identified and constrain only elements of the **C** matrix. (We need to allow very low, or zero parameters to develop in **B**, since any given single input will not generally excite all eigenvalues.) In a single output case, set **C** = [1 1 1 ….] and in models for multiple outputs, distribute the constraint to 'tie' given eigenvalue pairs (σ, ω) to given outputs. Eg in eqns 6 choose $c_{11} = c_{12} = c_{23} = c_{24} = c_{25} = 1$. The risk of poor conditioning in the result is then minimised, as the eigenvalue parameters will be identified in the order in which they are most influential within each output; we will see an example of this in Section IV.

The identification is achieved by concatenation of the states and all of the parameters to be identified into one large modified state vector, with m elements (34 in the example), eg

$$\mathbf{z}=[x_1 \quad x_2 \quad \cdots \quad x_5 \quad \sigma_1 \quad \omega_1 \quad \sigma_2 \quad \omega_2 \quad \sigma_3 \quad b_{11} \quad b_{12} \quad \cdots \quad b_{52} \quad c_{11} \quad c_{12} \quad \cdots \quad c_{25} \quad d_{11} \quad \cdots \quad d_{22}]^T \qquad (7)$$

With the model f(z) defined as above for the true states, and with expected propagation of the parameter states set to zero :

$$\begin{aligned} \dot{\mathbf{z}}_{1-5} &= \mathbf{f}\left(\mathbf{z}_{1-5}\right) = \mathbf{Ax} + \mathbf{Bu} \\ \dot{\mathbf{z}}_{6-m} &= \mathbf{f}\left(\mathbf{z}_{6-m}\right) = \mathbf{0} \end{aligned} \qquad (8)$$

and the output model $\mathbf{h} = \mathbf{Cx} + \mathbf{Du}$

Now the GIF also requires the Jacobian matrices (equations 4) and these follow a simple structure, due to the above linear definitions. The easiest way to construct them is to define **f** and **h** in a symbolic computing environment such as Maple, or Matlab's symbolic toolbox, and find the differential matrix with respect to the full state set *z* – eg using Matlab's **`jacobian`** command. The pattern of the Jacobian is also easily illustrated using our example; given the definitions in equations 6 & 7 – eg

$$\dot{\mathbf{z}}_1 = \mathbf{f}_1 = z_6 z_1 + z_7 z_2 + z_{11} u_1 + z_{16} u_2 \qquad (9)$$

Hence

$$\mathbf{F}=\frac{\partial \mathbf{f}}{\partial \mathbf{z}}=\begin{bmatrix} z_6 & z_7 & 0 & 0 & 0 & z_1 & z_2 & 0 & 0 & 0 \\ -z_7 & z_6 & 0 & 0 & 0 & z_2 & -z_1 & 0 & 0 & 0 \\ 0 & 0 & z_8 & z_9 & 0 & 0 & 0 & z_3 & z_4 & 0 \\ 0 & 0 & -z_9 & z_8 & 0 & 0 & 0 & z_4 & -z_3 & 0 \\ 0 & 0 & 0 & 0 & z_{10} & 0 & 0 & 0 & 0 & z_5 \\ \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 \end{bmatrix} \cdots$$

$$\cdots \begin{bmatrix} u_1 & 0 & 0 & 0 & 0 & u_2 & 0 & 0 & 0 & 0 & \mathbf{0}_{21-34} \\ 0 & u_1 & 0 & 0 & 0 & 0 & u_2 & 0 & 0 & 0 & \mathbf{0}_{21-34} \\ 0 & 0 & u_1 & 0 & 0 & 0 & 0 & u_2 & 0 & 0 & \mathbf{0}_{21-34} \\ 0 & 0 & 0 & u_1 & 0 & 0 & 0 & 0 & u_2 & 0 & \mathbf{0}_{21-34} \\ 0 & 0 & 0 & 0 & u_1 & 0 & 0 & 0 & 0 & u_2 & \mathbf{0}_{21-34} \\ \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34}^6 & \mathbf{0}_{34,21-34}^6 \end{bmatrix} \qquad (10)$$

where $\mathbf{0}_{21\text{-}34}$ indicates zero elements in columns 21-34 and $\mathbf{0}_{34}^6$ indicates zeros in rows 6 – 34. A similar simple and predictable structure arises for $H = \frac{\partial \mathbf{h}}{\partial \mathbf{z}}$.

Execution of the GIF requires a set of equations to be computed in sequence at each time step of the recorded time histories :

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k^{\mathrm{T}} \left[\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^{\mathrm{T}} + \mathbf{R} \right]^{-1} \qquad (11)$$

$$\mathbf{P}_k^* = \left[\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right] \mathbf{P}_k \qquad (12)$$

$$\mathbf{P}_{k+1} = \mathbf{P}_k^* + T \left[\mathbf{F}_k \mathbf{P}_k^* + \mathbf{P}_k^* \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q} \right] \qquad (13)$$

$$\hat{\mathbf{z}}_{k+1} = \hat{\mathbf{z}}_k + T \mathbf{f}_k + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{h}_k \right) \qquad (14)$$

where T is the time step length, which should be small relative to the system dynamics to ensure filter stability and accuracy. Error expectation matrices Q and R can be chosen nominally, as R = I and $\mathbf{Q} = \lambda \mathbf{I}'_m$. Where $\mathbf{I}'_m$ is the m x m identity matrix with zeros set for the true states (first 5 diagonal elements in this example) and also for the constrained c parameters. λ is the only tuning parameter needed to run the

GIF and it is typically set $\lambda < 1$, with higher values eliciting faster parameter migration, though with associated risk of instability in the filter if λ is set too high.

III. Identification Method and Example

A nominal linear model is used here as a test platform to illustrate appropriate data collection to apply to the GIF and explore its performance, convergence, robustness and flexibility. This test 'plant to be identified' is

$$\mathbf{A}=\begin{bmatrix} -4 & 0 & 0 & 0 & 0 \\ 0 & -15 & 10 & 0 & 0 \\ 0 & -10 & -15 & 0 & 0 \\ 0 & 0 & 0 & -8 & 40 \\ 0 & 0 & 0 & -40 & -8 \end{bmatrix} \quad \mathbf{B}=\begin{bmatrix} 0.1 & 0.1 \\ 0.3 & -3 \\ 1.5 & 0 \\ 10 & -0.5 \\ 0.7 & 1 \end{bmatrix} \quad (15)$$

$$\mathbf{C}=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 0.3 & 0.35 & -1.35 & -0.06 \\ -2.75 & -1.3 & 3.6 & 1.4 & 0.77 \end{bmatrix} \quad \mathbf{D}=\begin{bmatrix} 0 & 0 \\ 1.5 & 0.67 \\ -0.2 & 1.4 \end{bmatrix}$$

in which most matrix elements (parameters) were set arbitrarily, with some deliberately zero, but with **A** and the first row of **C** set to conform to a possible identified model structure. This is in order for performance to be illustrated in terms of the parameter match and not solely in terms of accuracy in the outputs.

The key to successful identification lies as much in the test (input) data used to excite the unknown plant as in the method which operates on the resulting **u** and **y** data. Here, test data is obtained by stimulating the test plant using a 100 second sequence of normally distributed white noise, sampled at a nominally high sampling rate of 500Hz (T=0.002) and then filtered to remove content above 25Hz. A small offset is also applied to each signal :

$$u_1 = N(0,1)+0.1 \quad u_2 = N(0,1)-0.1 \quad (0-25\text{Hz}) \qquad (16)$$

In practice the filtered noise signal can be approximated by defining a vector of 5000 normally distributed random numbers, treating these as points timed in the range 0-100 seconds at 50Hz, and then interpolating the data to increase the sampling rate to 500Hz.

White noise data is used to ensure excitation of the system dynamics across all frequencies, making no prior assumptions about system resonance frequencies, and a high sampling rate is required to reduce error or instability of the filter. Here the 25Hz low pass filter is not a requirement for the GIF to function, but improves the speed of convergence by allowing higher setting of the tuning parameter λ. The offset on each input is valuable to identify any system which is known to, or may contain non-zero elements in D. Its presence provides a significant zero frequency content, forcing separated determination of the C and D matrices.

Note that very little a priori knowledge of the system has been assumed here – the cutoff at 25Hz is simply 'well above' the highest dynamic frequencies of interest, and the appropriate range could easily be determined for any entirely unknown system by observation of output PSD in response to high bandwidth white noise inputs. Alternative input types could be considered, and these are likely to be effective, but the resulting models may be reliable only in the frequency range of the prescribed inputs; of course this might be done deliberately to identify a simple model for a specific limited purpose.

Output data was obtained by simulation of model eqns (15) from zero initial conditions and with outputs sampled at the same rate as the inputs. Only input and output data vectors (**u**, **y**) defined at each of the time samples k=1-50000 are provided to the GIF. Note that, in general it is wise to normalise both **u** and **y** prior to identification, eg by scaling according to the RMS of each signal, in order to maximise conditioning in the identified model, but in the examples shown here the outputs naturally emerge with similar magnitudes to the inputs, so this is unnecessary. For its first iteration the GIF is initialised with 'true' states $z_{1-n}=0$, with parameter states associated with the eigenvalues, $z_{(n+1)-2n}=-1$ and with most of the remaining parameter states $z_{(2n+1)-m}=0$; constrained c parameters within $z_{(2n+1)-m}$ are set to 1. This ensures the identified model initialises in a stable form but with unknown parameters. Various alternative starting conditions, using randomised $z_{(2n+1)-m}$ and a range of alternative (negative) $z_{(n+1)-2n}$ have been tested without significant variation in the results. A fast setting of $\lambda = 1$ is used for the majority of results discussed below and (although it is insensitive), **P** can sensibly be initialised $\mathbf{P_0} = \mathbf{Q}$.

The GIF is operated over the available 100 seconds of data repeatedly – essentially 'rinsing' the data through the filter multiple times. At the start of each iteration the true states are (appropriately) reset to zero, but all the parameter states, **K** and **P** matrices are carried over from the last step of the previous iteration.

IV. Identification Results

A. Convergence

First consider a simplified test, identifying model equations (15) from the two inputs to just the first output. In the first two iterations of the GIF, illustrated in Fig 1, all parameters rapidly diverge and some then start to converge, with the set at the end of iteration 2 providing output accuracy R = 99.4%

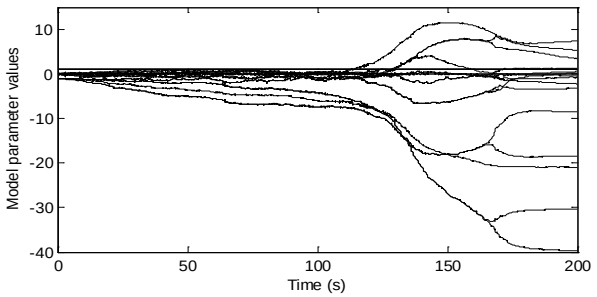

Fig. 1: Development of parameter values in the first two iterations

Here we use the 'percentage explanation' measure of performance

$$R = \left(1 - \frac{\sum_k \left(y_k - \hat{y}_k \right)^2}{\sum_k y_k^2} \right) \times 100 \tag{17}$$

In subsequent iterations accuracy rapidly reaches very nearly 100% (Fig. 2a) while the parameters take around 100 iterations to converge finally to a settled, final value (Fig. 2b). On the mid-range PC used to conduct these tests, 100 iterations took around 4.4 minutes. Trace(P) provides a simple single variable which can be used to detect convergence (Fig. 2c); the P matrix is the covariance of expected error in the states, so when all parameter states converge, so do their expected error. Alternatively a sum of the parameter states could be used for the same purpose. Note how different combinations of parameters achieve very close to 100% accuracy through iterations 2 – 100, yet the final parameter values conform very accurately to the original model (table 1). Interestingly the identified b_{51} parameter is not accurate – this demonstrates its relative insensitivity in influencing the output. One of the eigenvalues has also converged 'incorrectly' with real part -9.5 rather than -8 and in fact this will be true of most identification results shown here.

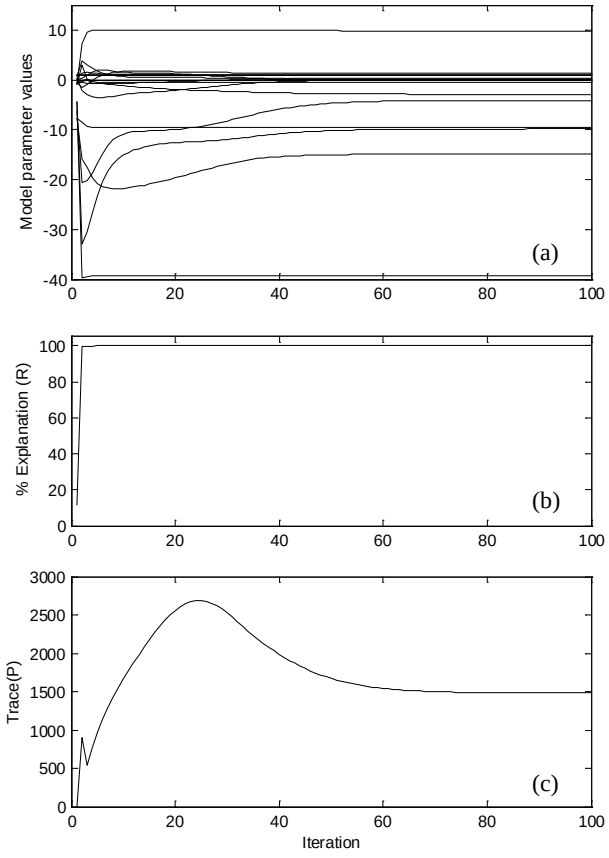

Fig. 2: Explanation and parameter convergence metrics (to 1 output)

Table 1: Final identified parameters (to 1 output)

Identified parameters (source model parameters)						
-4.09 (-4)	0	0	0	0	0.104 (0.1)	0.106 (0.1)
0	-14.89 (-15)	9.83 (10)	0	0	0.307 (0.3)	-2.87 (-3)
0	-9.83 (-10)	-14.89 (-15)	0	0	1.45 (1.5)	0.019 (0)
0	0	0	-9.51 (-8)	39.32 (40)	9.86 (10)	-0.413 (-0.5)
0	0	0	-39.32 (-40)	-9.51 (-8)	-0.096 (0.7)	1.02 (1)
1	1	1	1	1	0.0123 (0)	-0.002 (0)

Repeating the exercise with the GIF applied to all three outputs illustrates the point further. Fig. 3 shows parameter and trace(P) divergence, in spite of 100% accuracy, and table 2 highlights the unexpected results, including a coupled pair of ill-conditioned and insensitive parameters; in practice however, any model taken beyond around the 10th iteration is accurate and successful.

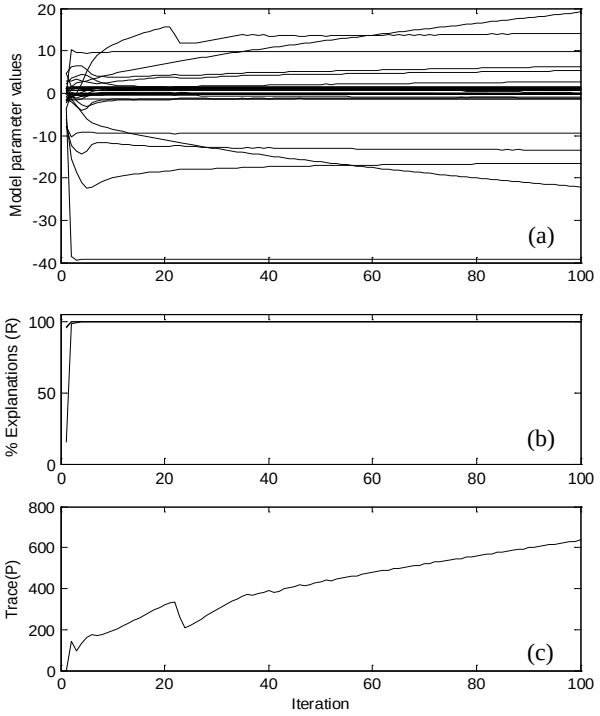

Fig. 3: Explanation and parameter convergence metrics (to 3 outputs)

Table 2: Final identified parameters (to 3 outputs)

Identified parameters (source model parameters)						
-15.95 (-4)	0	0	0	0	0.029 (0.1)	-37.6 (0.1)
0	-13.94 (-15)	0.57 (10)	0	0	4.39 (0.3)	41.55 (-3)
0	-0.57 (-10)	-13.94 (-15)	0	0	-2.19 (1.5)	-6.94 (0)
0	0	0	-9.46 (-8)	39.32 (40)	9.75 (10)	-0.36 (-0.5)
0	0	0	-39.32 (-40)	-9.46 (-8)	-0.17 (0.7)	1.00 (1)
1	1	1	1	1	0.0108 (0)	-0.002 (0)
-0.367 (2)	-0.606 (0.3)	-1.55 (0.35)	-1.36 (-1.35)	-0.054 (-0.06)	1.49 (1.5)	0.670 (0.67)
7.25 (-2.75)	9.45 (-1.3)	16.84 (3.6)	1.41 (1.4)	0.756 (0.77)	-0.182 (-0.2)	1.40 (1.4)

In the above example *c* parameters were constrained only in the first output, to allow easier comparison of identified parameters. However the suggested method prescribes sharing the *c* constraints across all three outputs, and we should have constrained as

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & c_{13} & c_{14} & c_{15} \\ c_{21} & c_{22} & 1 & 1 & c_{25} \\ c_{31} & c_{32} & c_{33} & c_{34} & 1 \end{bmatrix} \quad (18)$$

It was suggested this would allow a better conditioned model to evolve.

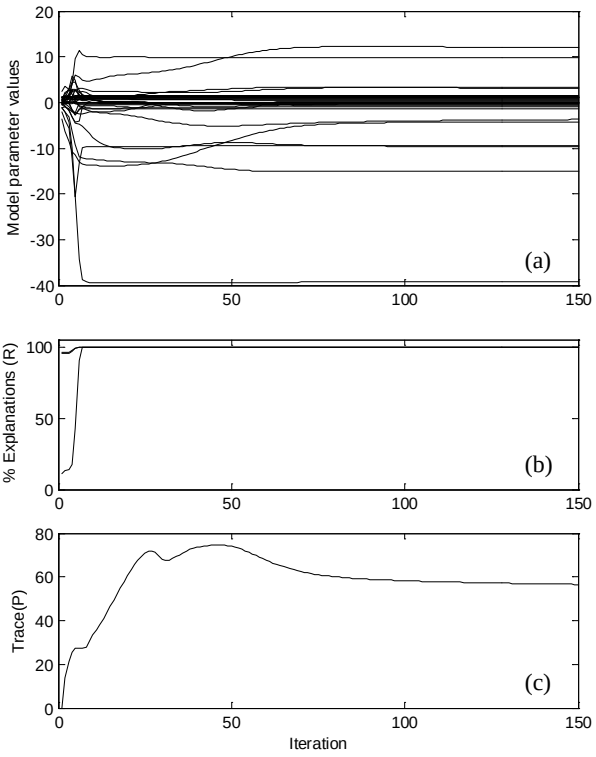

Fig. 4: Explanation and parameter convergence metrics (with better constraints to 3 outputs)

Table 3: Final identified parameters (with better constraints to 3 outputs)

Identified parameters (source model parameters *only given here where comparison is meaningful*)						
-4.10 (-4)	0	0	0	0	-0.285	-0.290
0	-14.93 (-15)	9.67 (10)	0	0	0.129	-0.923
0	-9.67 (-10)	-14.93 (-15)	0	0	0.452	0.07
0	0	0	-9.51 (-8)	39.32 (40)	9.86 (10)	-0.41 (-0.5)
0	0	0	-39.32 (-40)	-9.51 (-8)	-0.103 (0.7)	1.02 (1)
-0.360	3.33	2.98	1	1	0.0123 (0)	-0.002 (0)
0.73	1	1	-1.35 (-1.35)	-0.060 (-0.06)	1.49 (1.5)	0.670 (0.67)
1	-3.44	11.7	1.40 (1.4)	0.770 (0.77)	-0.181 (-0.2)	1.40 (1.4)

Indeed this is the case; Fig. 4 shows better convergence and table 3 gives the final model, this time with many variables not directly comparable to the original model due to the revised constraint structure. As in table 1, only parameter b_{51} is now 'incorrect'. Note that in this and the above experiments the identified model states are actually identified in a different order to that of equations (15), so the states (and hence model parameters) have been re-ordered to aid comparison in the tables.

B. Robustness to Noise

Returning to a single output case, zero mean white noise was added to the output in proportion to its RMS in ratios 10, 30, 50 and 70% prior to identification. The GIF converged in all cases, with predictably lower performance (table 4) but note that when the identified model outputs were then compared with noise-free original data, R_0 in the table, performance remains above 99% even in the worst case. Identified model parameters vary slightly as noise increases, but the technique and ultimate performance of the underlying models remains robust. Although not illustrated, similar results were seen from models identified for all three outputs. Of course any systematic bias in the noise would result in a model with modes identified to explain that bias.

Table 4: Converged model explanations under the influence of noise in the output

	Added noise ratio			
Explanation	10%	30%	50%	70%
R (of noisy data)	98.97	91.75	79.45	66.84
R_0 (of clean data)	99.96	99.97	99.60	99.90

C. Reduced and Higher order Models

If we assume inputs and outputs have been collected without knowledge of the source model order, it should be possible to establish the best order choice by running a series of identifications with incremental *n*. Consider identifications from all three outputs of model equations (15) with **C** constraints as prescribed in the method, ie

$$n=3,\ \mathbf{C} = \begin{bmatrix} 1 & 1 & c \\ c & c & 1 \\ c & c & c \end{bmatrix} \qquad n=4,\ \mathbf{C} = \begin{bmatrix} 1 & 1 & c & c \\ c & c & 1 & 1 \\ c & c & c & c \end{bmatrix} \quad (19)$$

$$n=6,\ \mathbf{C} = \begin{bmatrix} 1 & 1 & c & c & c & c \\ c & c & 1 & 1 & c & c \\ c & c & c & c & 1 & 1 \end{bmatrix} \qquad n=7,\ \mathbf{C} = \begin{bmatrix} 1 & 1 & c & c & c & c & c \\ c & c & 1 & 1 & c & c & c \\ c & c & c & c & 1 & 1 & 1 \end{bmatrix}$$

Accuracy, trace(P) and parameter convergence results for 3rd – 7th order fitted models are compared in Fig. 5; some results in Fig 5a and Fig 5b have been scaled here – the intention is to illustrate their divergent or convergent nature, and for this reason Fig 5a illustrates only the most divergent parameter in the set.

These results shed some light on the full output 5th order

results seen earlier. Here we see that all order choices apart from *n*=4 and *n*=5 result in divergent **P** and some divergent or slowly varying parameters; conversely we see very fast convergence in the 4th order result. As we would expect, final accuracy increases with model order; clearly any well-posed model with a greater number of parameters should fit the data with higher accuracy, and all results with *n* above 3 produce excellent explanations. Clearly the result at *n*=3 would be rejected, but results at *n*=6 or 7 are also undesirable, since **P** diverges markedly, even though most of the parameters do seem to settle (with some drift, as seen in the most divergent parameters illustrated).

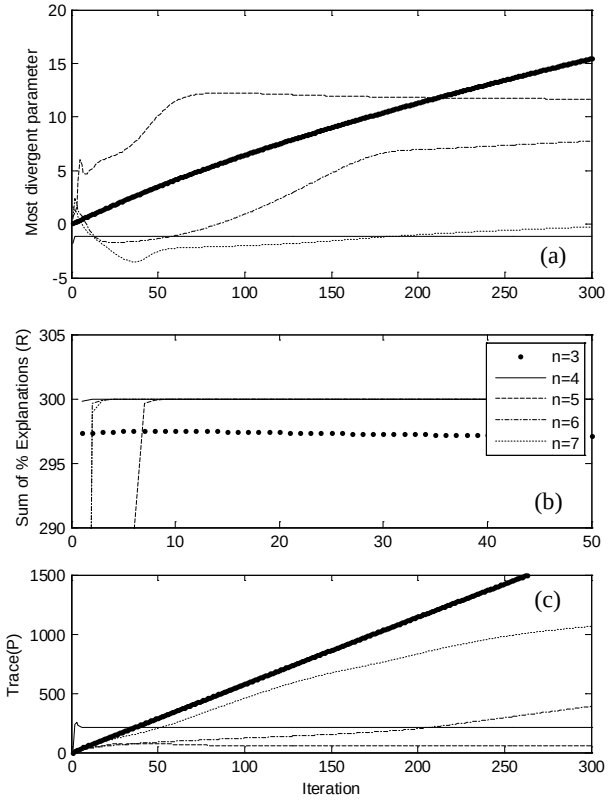

Fig. 5: Explanation and parameter convergence metrics (with varying *n*)

We know the original model has five eigenvalues, but the best trade-off between accuracy and a consistent (well conditioned) identified model is achieved with four. This is because the dynamic influence of the source model's eigenvalue at -4 is very weak in combination with the other modes; it turns out that the source model itself has been arbitrarily devised with poor conditioning. Identifications with five states and more than one output can yield unreliable estimates for the single eigenvalue, along with corresponding divergent parameters associated with it (in the **C** matrix). Provided the objective of any real identification is extraction of the best possible model to describe the outputs, the four state solution here is an excellent result, regardless of the fact it doesn't replicate the original model. For comparison, it's eigenvalues are identified at $\lambda_{1,2} = -9.57 \pm 39.38j$ and $\lambda_{1,2} = -12.68 \pm 10.27j$.

The GIF is an effective identification tool because a) results above demonstrate it can perfectly identify even a poorly conditioned model, and b) by incremental exploration of required model order, it is easy to establish the lowest order model which best explains the outputs.

V. Conclusion

A novel method for system identification using a simplified extended Kalman Filter has been presented. The filter identifies a linear model of any order from multiple input and multiple output data sources alone, without the need for any understanding of the underlying system dynamics. The model is identified in a modal canonical form, and the presented example shows that it can be identically matched where there is no noise, and matched with a high level of accuracy under high levels of noise, provided the noise does not correlate with the system. The method has just one tuning parameter, which influences speed, but not accuracy of convergence, and clear convergence metrics have been demonstrated. Further, by progressively incrementing model order it is easy to identify a reduced order model to approximate a complex system, and it is straightforward for the user to determine the best compromise between accuracy and good conditioning. Finally, it is in fact also very straightforward to extend the remit of this method beyond linear identification. Any smoothly nonlinear model structure can be identified using the same approach, and results exploring nonlinear model identification will be explored in future publications.

References

[1]. Soderstrom, T. and Stoica, P., System Identification, Prentice Hall International, 1989.

[2]. Aguero, J.C., Rojas, C.R., Hjalmarsson, H. and Goodwin, G.C., "Accuracy of Linear Multiple-Input Multiple-Output (MIMO) Models Obtained by Maximum Likelihood Estimation", Automatica, 2012, vol 48, no 4, pp 632-637.

[3]. Behzad, H., Shandiz, H., Toossian, N.A. and Abrishami, T., "Robot Identification using Fractional Subspace Method", Proceedings 2011 2nd International Conference on Control, Instrumentation and Automation, 2012, pp1193-1199.

[4]. Polifke, W., "Black-box System Identification for Reduced Order Model Construction", Annals of Nuclear Energy, 2014, vol 67, pp 109-128.

[5]. Ahn, H-J., Lee, S-W., Lee, S-H and Han, D-C., "Frequency Domain Control-relevant Identification of MIMO AMB Rigid Rotor", Automatica, 2003, vol 39, no 2, pp 299-307.

[6]. Best, M.C., "Parametric identification of vehicle handling using an extended Kalman filter", International Journal of Vehicle Autonomous Systems, Vol 5, No 3 / 4, 2007, pp 256 – 273.

[7]. Best M.C., Gordon T.J. and Dixon P.J., "An Extended Adaptive Kalman Filter for Real-time State Estimation of Vehicle Handling Dynamics", Vehicle System Dynamics, 2000, vol 34, no 1, pp 57-75.

[8]. Kallapur, A., Samal, M., Puttige, V., Anavatti, S. and Garratt, M., "A UKF-NN Framework for System Identification of Small Unmanned Aerial Vehicles", Proceedings IEEE Conference on Intelligent Transportation Systems, 2008, pp 1021-1026.

[9]. Hassani, V., Aguiar, A. P., Athans, M. and Pascoal, A.M., "Multiple Model Adaptive Estimation and Model Identification using a Minimum Energy Criterion", Proceedings American Control Conference, 2009, pp 518-523.

[10]. Ding, F., Liu, Y. and Bao, B., "Gradient-based and Least-Squares-based Iterative Estimation Algorithms for Multi-input Multi-output Systems", Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 2012, vol 226, no 1, pp 43-55.