NARMAX Model and Its Application to Forecasting Geomagnetic Indices

Dr Hua-Liang (Leon) Wei

Senior Lecturer in System Identification and Data Analytics

Head of Dynamic Modelling, Data Mining & Decision Making (3DM) Lab

Complex Systems & Signal Processing Research Group

Department of Automatic Control & System Engineering

University of Sheffield

Sheffield, UK

Key Topics

The University Of Sheffield.

- **NARMAX Methodology**
  - ◊ NARMAX method
  - ◊ OFR-ERR algorithm

    (orthogonal forward regression and error reduction ratio algorithms)

- **Application**

  Forecast of geomagnetic indices

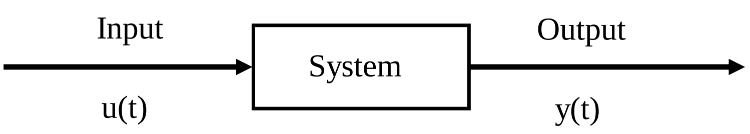
Part 1

Linear and Nonlinear Models of Dynamic Systems

Dynamic System Identification (1)
– Learning From Data

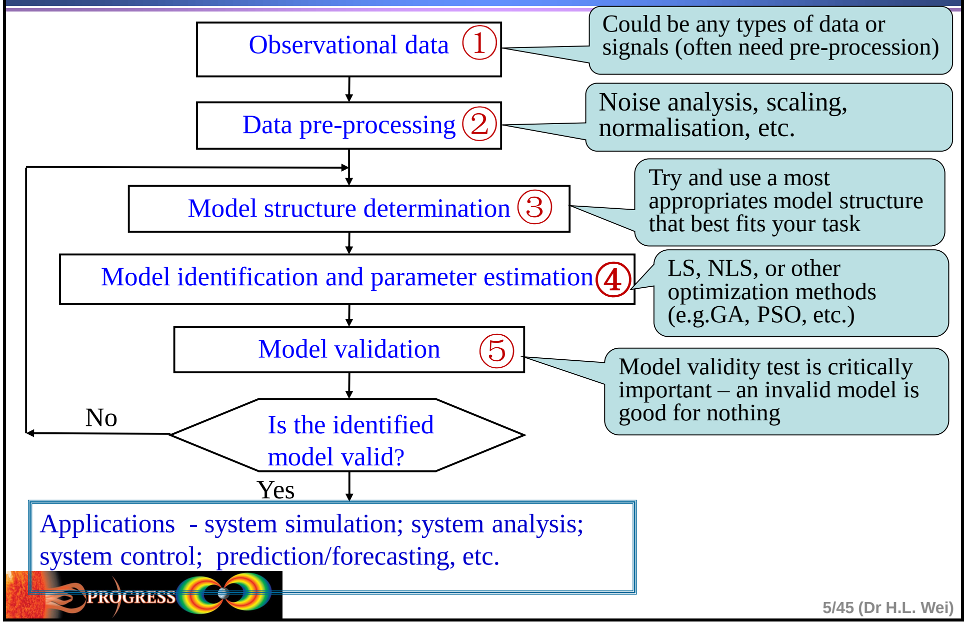
For a system where the model (both the model structure and the associated parameters) are known, one can directly analyse the system using the given model.

If, however, the model structure of the system is *unknown*, but only some observational data are available, how can we do to uncover the inherent dynamics of the system?



Dynamic System Identification (2)

– A Comprehensive Procedure



ARX and ARMAX models

- **ARX model**

ARX — Auto-Regressive (AR) with eXogenous inputs

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + \cdots + a_p y(k-p) + b_1 u(k-1) + b_2 u(k-2) + \cdots + b_q u(k-q) + e(k)$$

- **ARMAX model**

ARMAX — Auto-Regressive (AR), Moving Average (MA) with eXogenous inputs

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + \cdots + a_p y(k-p) + b_1 u(k-1) + b_2 u(k-2) + \cdots + b_q u(k-q) + e(k) + c_1 e(k-1) + c_2 e(k-2) + \cdots + c_r e(k-r)$$

NARX and NARMAX models

- **NARX model**

  NARX - Nonlinear Auto-Regressive (NAR) with eXogenous inputs

$$y(k) = f(y(k-1), y(k-2), \cdots, y(k-p), u(k-1), u(k-2), \cdots, u(k-q)) + e(k)$$

- **NARMAX model**

  NARMAX - Nonlinear Auto-Regressive (NAR), Moving Average (MA) with eXogenous inputs

$$y(k) = f(y(k-1), y(k-2), \cdots, y(k-p), u(k-1), u(k-2), \cdots, u(k-q), e(k-1), e(k-2), \cdots, e(k-r)) + e(k)$$

- AR, ARMA, ARX, and ARMAX are special cases of NARMAX.

Polynomial NARX Model (1)

For the NARX model

$$y(k) = f(y(k-1), y(k-2), \cdots, y(k-p), u(k-1), u(k-2), \cdots, u(k-q)) + e(k)$$

Let $x_j(k) = \begin{cases} y(k-j), & 1 \le j \le p \\ u(k-j+p), & p+1 \le j \le n, \quad (n = p+q) \end{cases}$

$$\mathbf{x}(k) = [x_1(k), x_2(k), \cdots, x_n(k)]^T$$

Then, $y(k) = f(\mathbf{x}(k)) + e(k) = f(x_1(k), x_2(k), \cdots, x_n(k)) + e(k)$

e.g. $y(k) = f(y(k-1), y(k-2), u(k-1)) = f(x_1(k), x_2(k), x_3(k))$

$$= \theta_0 + \theta_1 x_1(k) + \theta_2 x_2(k) + \theta_3 x_3(k) + \theta_4 x_1(k)x_1(k) + \theta_5 x_1(k)x_2(k) + \theta_6 x_1(k)x_3(k) + \theta_7 x_2(k)x_2(k) + \theta_8 x_2(k)x_3(k) + \theta_9 x_3(k)x_3(k) + e(k)$$

$$\begin{cases} x_1(k) = y(k-1) \\ x_2(k) = y(k-2) \\ x_3(k) = u(k-1) \end{cases}$$

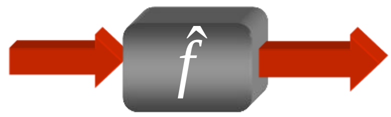
or $y(k) = \theta_0 + \sum_{i_1=1}^{3} \theta_{i_1} x_{i_1}(k) + \sum_{i_1=1}^{3} \sum_{i_2=i_1}^{3} \theta_{i_1 i_2} x_{i_1}(k) x_{i_2}(k) + e(k)$

Polynomial NARX Model (2)

- One approach to approximate the unknown function *f* is

$$y(k) = \hat{f}(x_1(k), x_2(k), \cdots, x_n(k)) + e(k)$$
$$= f_0 + \sum_{1\le i\le n} f_i(x_i(k)) + \sum_{1\le i\le j\le n} f_{ij}(x_i(k), x_j(k)) + \cdots$$
$$+ \sum_{1\le i_1\le\cdots\le i_\ell\le n} f_{i_1 i_2\cdots i_\ell}(x_{i_1}(k), x_{i_2}(k), \cdots, x_{i_\ell}(k)) + e(k)$$

- Here the aim is to approximate a high-dimensional function *f* using a set of lower dimensional functions.

$\mathbf{x}(k) = [x_1(k), x_2(k), \cdots, x_n(k)]^T$  $y(k)$

PROGRESS

Polynomial NARX Model (3)

- Polynomial approximation

$$y(k) = \hat{f}(x_1(k), x_2(k), \cdots, x_n(k)) + e(k)$$
$$= \theta_0 + \sum_{i_1=1}^{n} \theta_{i_1} x_{i_1}(k) + \sum_{i_1=1}^{n} \sum_{i_2=i_1}^{n} \theta_{i_1 i_2} x_{i_1}(k) x_{i_2}(k) + \cdots$$
$$+ \sum_{i_1=1}^{n} \cdots \sum_{i_\ell=i_{\ell-1}}^{n} \theta_{i_1 i_2\cdots i_\ell} x_{i_1}(k) x_{i_2}(k) \cdots x_{i_\ell}(k) + e(k)$$

- An example (a model for Dst prediction)

$$Dst(k) = 0.02486 + 0.98368 Dst(k-1)$$
$$-0.92130\ [Dst(k-1)]^3 \times VBs(k-1)$$
$$+0.51936\ [Dst(k-1)] \times [Dst(k-1)]^2 \times VBs(k-2)$$
$$-1.25977 Dst(k-1) \times [VBs(k-1)]^2 \times VBs(k-2)$$

HL Wei, SA Billings & MA Balikhin, *J. Geophysical Research-Space Physics*, 109, A07212, 2004.

PROGRESS

Polynomial NARX Model (4)

- **Some KEY issues in NARX modelling**
- How to determine the model order?

$$y(k) = f(y(k-1), y(k-2), \cdots, y(k-p), u(k-1), u(k-2), \cdots, u(k-q)) + e(k)$$

- How to chose model variables?
- How to determine nonlinear degree of the model?

$$y(k) = \theta_0 + \sum_{i_1=1}^{n} \theta_{i_1} x_{i_1}(k) + \sum_{i_1=1}^{n} \sum_{i_2=i_1}^{n} \theta_{i_1 i_2} x_{i_1}(k) x_{i_2}(k) + \cdots$$

- How to determine model terms/regressors?
- How to determine model size/length/complexity?

Polynomial NARX Model (5)

- **Advantages of the polynomial NARX model**
- Widely applicable and applied
- Tractable: linear-in-the-parameters form; easy to operate
- Computational efficient: easy to compute
- Transparent: significant model terms and variables are clearly known
- Less sensitive to noise and thus usually generalises well
- Physically interpretable: can be related back to the underlying system
- Frequency domain analysis of nonlinear systems is allowable by mapping a time-domain model into the frequency domain

Challenges of Black-Box Modelling for Dynamic Systems

The University Of Sheffield.

- Model variable selection and determination
- Model structure determination
- Model term selection
- Model parameter estimation
- Model validity test
- Model interpretability

PROGRESS

The University Of Sheffield.

Part 2

NARMAX Model Identification and Construction

PROGRESS

The University Of Sheffield.

Part 2A Orthogonal Basis

Signal Approximation with Orthogonal Regression

PROGRESS

The University Of Sheffield.

Projection onto Orthogonal Vectors(1)

Let $x_1, x_2, \ldots, x_m$ be m orthogonal vectors defined in n-dimensional space R^n; and y a signal in R^n.

Assuming that we want to approximate y using $x_1, x_2, \ldots, x_m$, a conventional approach is:

$$y = c_1x_1 + c_2x_2 + \ldots + c_mx_m + e$$

where $c_1, c_2, \ldots, c_m$ are parameters and e is approximation error. Note that e is assumed to be independent of $x_1, x_2, \ldots, x_m$.

We can show that

$$<x_1, y> = c_1 <x_1, x_1> \ \Rightarrow\ c_1 = \frac{<x_1, y>}{<x_1, x_1>} = \frac{x_1^T y}{x_1^T x_1},$$

$$<x_2, y> = c_2 <x_2, x_2> \ \Rightarrow\ c_2 = \frac{<x_2, y>}{<x_2, x_2>} = \frac{x_2^T y}{x_2^T x_2}, \quad \ldots$$

$$<x_m, y> = c_m <x_m, x_m> \ \Rightarrow\ c_m = \frac{<x_m, y>}{<x_m, x_m>} = \frac{x_m^T y}{x_m^T x_m},$$

PROGRESS

Projection onto Orthogonal Vectors(2)

We can also show that

$$<y,y> = c_1^2 <x_1,x_1> + c_2^2 <x_2,x_2> + \ldots + c_m^2 <x_m,x_m> + <e,e>$$

That is,

$$y^T y = c_1^2 x_1^T x_1 + c_2^2 x_2^T x_2 + \ldots + c_m^2 x_m^T x_m + e^T e$$

or

$$\| y \|^2 = c_1^2 \| x_1 \|^2 + c_2^2 \| x_2 \|^2 + \ldots + c_m^2 \| x_m \|^2 + \| e \|^2$$

So,

$$\frac{\| e \|^2}{\| y \|^2} = 1 - c_1^2 \frac{\| x_1 \|^2}{\| y \|^2} - c_2^2 \frac{\| x_2 \|^2}{\| y \|^2} - \ldots - c_m^2 \frac{\| x_m \|^2}{\| y \|^2}$$

Projection onto Orthogonal Vectors(3)

$$\frac{\| e \|^2}{\| y \|^2} = 1 - c_1^2 \frac{\| x_1 \|^2}{\| y \|^2} - c_2^2 \frac{\| x_2 \|^2}{\| y \|^2} - \ldots - c_m^2 \frac{\| x_m \|^2}{\| y \|^2}$$

Recalling that $c_k = \frac{x_k^T y}{x_k^T x_k} = \frac{x_k^T y}{\| x_k \|^2}, k = 1, 2, .., m,$ we have

$$\frac{||e||^2}{||y||^2} = 1 - \left(\frac{x_1^T y}{||x_1||^2}\right)^2 \frac{||x_1||^2}{||y||^2} - \left(\frac{x_2^T y}{||x_2||^2}\right)^2 \frac{||x_2||^2}{||y||^2} - \ldots - \left(\frac{x_m^T y}{||x_m||^2}\right)^2 \frac{||x_m||^2}{||y||^2}$$

$$= 1 - \frac{(x_1^T y)^2}{||x_1||^2 ||y||^2} - \frac{(x_2^T y)^2}{||x_2||^2 ||y||^2} - \ldots - \frac{(x_m^T y)^2}{||x_m||^2 ||y||^2}$$

$$= 1 - ERR_1 - ERR_2 - \cdots - ERR_m$$

where ERR_k $(k = 1,2 \ldots ,m)$ is called the kth ***Error Reduction Ratio***, indicating how much (in percentage) of the approximation error can be reduced by the kth vector.

Note that $0 \leq \mathrm{ERR}_k \leq 1$, and $\sum \mathrm{ERR}_k \leq 1$

Projection onto Orthogonal Vectors(4)

The University Of Sheffield.

A simple example

$$y=\begin{bmatrix}1\\2\\5\end{bmatrix},\quad x_1=\begin{bmatrix}1\\0\\0\end{bmatrix},\quad x_2=\begin{bmatrix}0\\1\\0\end{bmatrix},\quad x_3=\begin{bmatrix}0\\0\\1\end{bmatrix}$$

$$c_1=\frac{x_1^T y}{x_1^T x_1}=1,\quad c_2=\frac{x_2^T y}{x_2^T x_2}=2,\quad c_3=\frac{x_3^T y}{x_3^T x_3}=5,$$

So, $y = x_1 + 2x_2 + 5x_3$

$$ERR_1=\frac{(x_1^T y)^2}{||x_1||^2||y||^2}=\frac{1}{30}=0.0333$$

$$ERR_2=\frac{(x_2^T y)^2}{||x_2||^2||y||^2}=\frac{4}{30}=0.1333$$

$$ERR_3=\frac{(x_3^T y)^2}{||x_3||^2||y||^2}=\frac{25}{30}=0.8333$$

x_1 accounts for 3.33% of the variation in y

x_2 accounts for 13.33% of the variation in y

x_3 accounts for 83.33% of the variation in y

PROGRESS

Projection onto Orthogonal Vectors(5)

The University Of Sheffield.

Question: Knowing x_1, x_2, x_3 and y, and assuming that we want to choose only one from x_1, x_2, x_3 that best approximates y, which one we would use?

What if we use only two?

$$y=\begin{bmatrix}1\\2\\5\end{bmatrix},\quad x_1=\begin{bmatrix}1\\0\\0\end{bmatrix},\quad x_2=\begin{bmatrix}0\\1\\0\end{bmatrix},\quad x_3=\begin{bmatrix}0\\0\\1\end{bmatrix}$$

An alternative question: Assuming that we want to choose a minimal subset of $\{x_1, x_2, x_3\}$ that accounts for no less than 80% of variation in y (i.e. 'overall ERR > 80%'), which and how many vector(s) should be used?

What if we want to achieve approximation that accounts for no less than 90% of the variation in y ?

PROGRESS

Part 2B Non-orthogonal Basis

Forward Orthogonal Regression

Forward Orthogonal Regression (1)

We use a simple example to illustrate the forward orthogonal process. We now have 3 linearly independent vectors, together with a 4th observed signal:

$$X = \left\{ \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}, \quad y = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}.$$

- **Step 1.**

Recalling the definition of the Error Reduction Ratio (ERR), we check the ERR index for each of the 3 vectors in S:

$$\text{err}_1 = \frac{(x_1^T y)^2}{||x_1||^2 ||y||^2} = \frac{5}{6} = 0.8333$$

$$\text{err}_2 = \frac{(x_2^T y)^2}{||x_2||^2 ||y||^2} = \frac{27}{50} = 0.54$$

$$\text{err}_3 = \frac{(x_3^T y)^2}{||x_3||^2 ||y||^2} = \frac{5}{6} = 0.8333$$

So, we choose either the 1st or 3rd vector.

Forward Orthogonal Regression (2)

- **Step 2.** We choose x_3 as the first orthogonal vector:

$$q_1 = x_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \text{(we know that } ERR_1 = 83.33\%\text{)}$$

Step 2 searches for a new vector to join q_1.

If x_1 joins q_1, we have

$$q_1 = x_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$$v = x_1 - \frac{q_1^T x_1}{q_1^T q_1} q_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} - \frac{2}{1}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix},$$

$$\text{err} = \frac{(v^T y)^2}{(v^T v)(y^T y)} = \frac{25}{150} = 16.67\%$$

If x_2 joins q_1, we have

$$q_1 = x_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$$v = x_2 - \frac{q_1^T x_2}{q_1^T q_1} q_1 = \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix} - \frac{2}{1}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix},$$

$$\text{err} = \frac{(v^T y)^2}{(v^T v)(y^T y)} = \frac{1}{30} = 3.33\%$$

Forward Orthogonal Regression (3)

Now we have 2 orthogonal vectors:

$$q_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \ (\text{ERR}_1 = 83.33\%), \quad q_2 = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} \ (\text{ERR}_2 = 16.67\%)$$

Since $\text{ERR}_1 + \text{ERR}_2 = 100\%$, meaning that the two vectors q_1 and q_2 totally explain the variation of y. So, there is no need to search further.

$$X = \left\{ \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}, \quad y = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}.$$

We can work out that,

$$y = 5q_1 + q_2 \text{ and } y = x_1 + 3x_3$$

Forward Orthogonal Regression (4)

The University Of Sheffield.

- **A general idea**

Let $x_1, x_2, \ldots, x_m$ be m vectors defined in n-dimensional space R^n; and y a signal in R^n.

Note that $x_1, x_2, \ldots, x_m$ can be linearly dependent or there is some multicollinearity among them.

We want to find an optimal or sub-optimal subset S of $\{x_1, x_2, \ldots, x_m\}$, such that y can be satisfactorily represented by elements of S.

Note that for the above scenario, the ordinary least squares method may not work well.

PROGRESS

Forward Orthogonal Regression (5)

The University Of Sheffield.

♦ **A general procedure**

- **Step 1.** Calculate ERR index for each of $x_1, x_2, \ldots, x_m$:

$$\text{err}_k = \frac{(x_k^T y)^2}{||x_k||^2 ||y||^2}, \quad k = 1,2,\ldots,m$$

  Choose the vector that has the maximum 'err' as the 1st orthogonal vector (q_1) .
- **Step 2.** Orthogonalize each of $x_1, x_2, \ldots, x_m$ (except that selected in Step 1) with q_1; work out ERR value for each of the orthogonalized vectors. Choose the one that with the maximum 'err' as the 2nd orthogonal vector (q_2) .
- **Step 3,4, ...** Repeat the same process as in Step 2, until a satisfactory approximation is achieved.

The above procedure is called orthogonal forward regression (OFR) or orthogonal least squares (OLS) algorithm

Forward Orthogonal Regression (6)

The University Of Sheffield.

♦ Why Using OFR rather than ordinary least squares?

Suppose we have a data tabular at the bottom, and we want to find a general regression model to characterize the dependent relation of y on the three independent variables x_1, x_2, x_3:

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_3+\beta_4x_1x_1+\beta_5x_1x_2+\beta_6x_1x_3+\beta_7x_2x_2+\beta_8x_2x_3+\beta_9x_3x_3$$

X_1	X_2	X_3	Y
2	2	8	8
0	0	0	0
1	2	5	6
1	1	2	3.5
2	2	8	8
1	1	2	3.5
3	2	13	10
0	1	1	2

Ordinary least squares failed to detect the correct model: $\beta_0 = 0$, $\beta_1 = -0.2121$, $\beta_2 = 0$, $\beta_3 = 2.5682$, $\beta_4 = 0$, $\beta_5 = 0$, $\beta_6 = -0.1212$, $\beta_7 = 0$, $\beta_8 = -0.5455$, $\beta_9 = -0.0227$.

The OFR algorithm, however, perfectly detect the correct model (with only 3 terms), step by step:

Step 1: x_1 was selected (ERR=96.154%, β_1=1)

Step 2: x_2 was selected (ERR= 3.693%, β_2=2)

Step 3: x_1x_2 was selected (ERR= 0.153%, β_5=1/2)

The University Of Sheffield.

Part 2C Dictionary Learning

For NARXMAX Model Identification

PROGRESS

Dictionary Learning

In NARMAX model identification, we need to design a dictionary in advance. **We use a simple example to illustrate the basic idea:**

$$y(k) = f(y(k\text{-}1), y(k\text{-}2), u(k\text{-}1)) + e(k)$$

Define: $D_0 = \{1\}$,

$$D_1 = \{y(k-1), y(k-2), u(k-1)\},$$

$$D_2 = \begin{Bmatrix} y(k-1)y(k-1) \\ y(k-1)y(k-2) \\ y(k-1)u(k-1) \\ y(k-2)y(k-2) \\ y(k-2)u(k-1) \\ u(k-1)u(k-1) \end{Bmatrix},$$

$$D_3 = \begin{Bmatrix} y(k-1)y(k-1)y(k-1) \\ y(k-1)y(k-1)y(k-2) \\ y(k-1)y(k-1)u(k-1) \\ y(k-1)y(k-2)y(k-2) \\ y(k-1)y(k-2)u(k-1) \\ y(k-1)u(k-1)u(k-1) \\ y(k-2)y(k-2)y(k-2) \\ y(k-2)y(k-2)u(k-1) \\ y(k-2)u(k-1)y(k-1) \\ u(k-1)u(k-1)u(k-1) \end{Bmatrix}$$

We can use D_0, D_1, D_2 and/or D_3 to create vector sets, and then apply the OFR algorithm to select important vectors (ie model terms, one by one), and build a compact or sparse model.

Part 3

NARMAX Model Application for Forecasting Geomagnetic Indices

Part 3A

Kp Index Prediction

Kp Index Prediction (1)

Variable	Description	Input or output
V	Solar wind speed [km/s]	Input
Bs	Southward interplanetary magnetic field [nT]	
VBs	solar wind rectified electric field [mv/m] [VBs=V·Bs/1000]	
p	Solar wind pressure [nPa]	
$p^{1/2}$	Square root of solar wind pressure	
Kp	Kp index (variable of interest)	Output

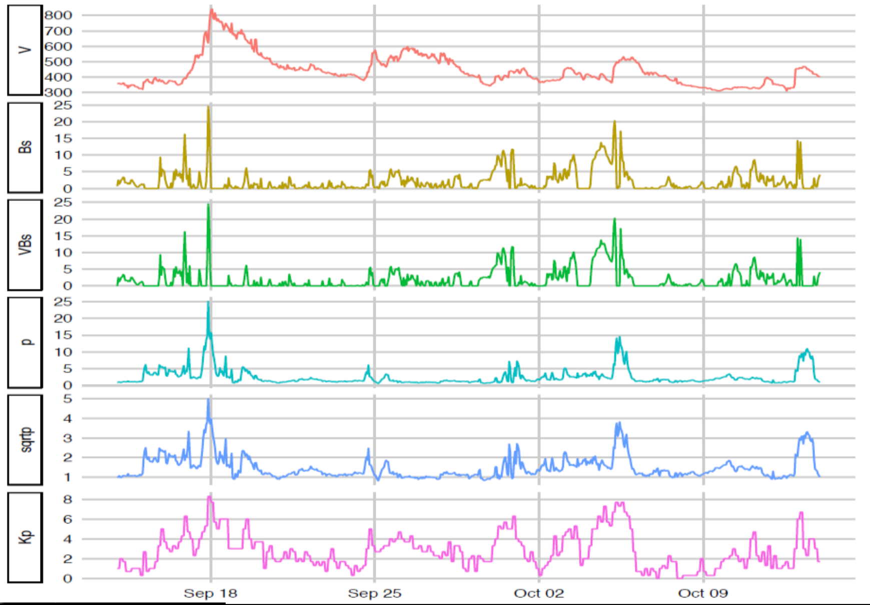
- Training data: Hourly data, January – June, 2000
- Test data: Hourly data, July – December, 2000

The identified model:

$$
\begin{aligned}
Kp(k) = {} & 0.325543Kp(k-3) - 0.000043V(k-1)\cdot p^{1/2}(k-1) + 0.673034Bs(k-1) \\
& - 0.164093Bs(k-1)\cdot p^{1/2}(k-1) - 0.000003V^2(k-1) \\
& + 0.000217V(k-1)\cdot Bs(k-2) - 0.006701Bs(k-1)\cdot Bs(k-2) \\
& - 0.005810Bs(k-1)\cdot p(k-2) - 2.179360 + 0.753122\, p^{1/2}(k-1) \\
& + 0.006105V(k-1) - 0.387292VBs(k-1) + 0.136271VBs(k-1)\cdot p^{1/2}(k-1)
\end{aligned}
$$

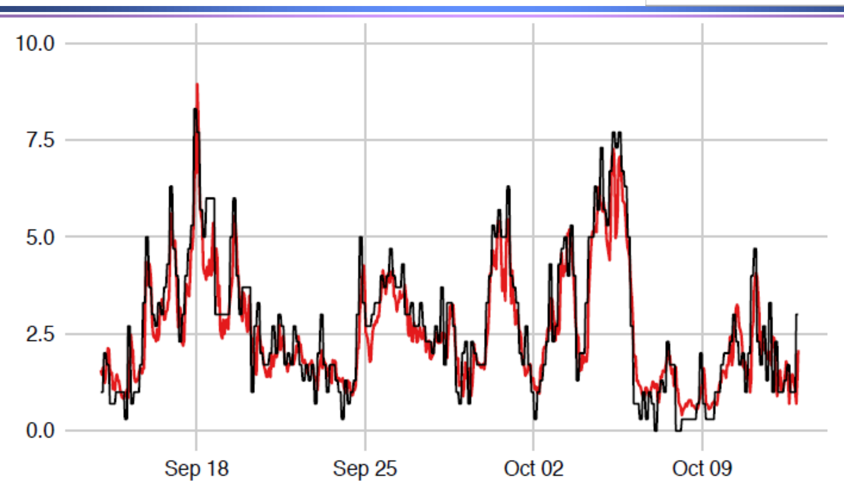
Kp Index Prediction (2)

The University Of Sheffield.



Kp Index Prediction (3)

The University Of Sheffield.



Comparison between the 3-hour ahead prediction of the Kp index during a 30-day interval between September and October of year 2000. **Red line** indicates the model predicted Kp values.

34/45 (Dr H.L. Wei)

The University Of Sheffield.

Part 3B

Forecasting the daily averaged flux electrons with energy > 2MeV at Geostationary orbit

PROGRESS

Forecast of Electron Flux (1)
at the Radiation Belt

The University Of Sheffield.

As a case study, we use the following data to train models:

Output variable:
Daily data of 120 days (22nd May 1995 - 17th Sept 1995) for electron flux at the radiation belt (>2MeV).
(data were from GOES 7 & 8 satellites)

Input variables:
Hourly data of 120 days (22nd May 1995-17th Sept 1995)

- V_{sw} (solar wind velocity)
- *VBs* (solar wind rectified electric field)
- *Pdyn* (flow pressure)
- *Sym-H index* (symmetric part of disturbance [nT])
- *Asy-H index* (asymmetric part of disturbance [nT])

(data were from ACE & WIND spacecraft and geomagnetic indices)

Forecast of Electron Flux (2)
at the Radiation Belt

The University Of Sheffield.

Our objective is to build models from these hourly and daily data, and use the models to forecast the future behaviour of electron flux.

Data Observed Today and Some Previous Days

Hourly recorded
V_{sw} (solar wind velocity)
VBs (rectified electric field)
Pdyn (flow pressure)
Sym-H index
Asy-H index

Daily recorded *Electrons*

→

Predict Tomorrow's Behaviour

Flux of electrons (> 2MeV)

PROGRESS

Forecast of Electron Flux (3)
– MISO NARX Model

The University Of Sheffield.

- We have 5 input variables (V, VBs, P, Sym-H, Asy-H), and 1 output variable (electron flux).
- We use previous values of these input and output variables to build models. Specifically, we use the values below to predict the future value of electron flux:

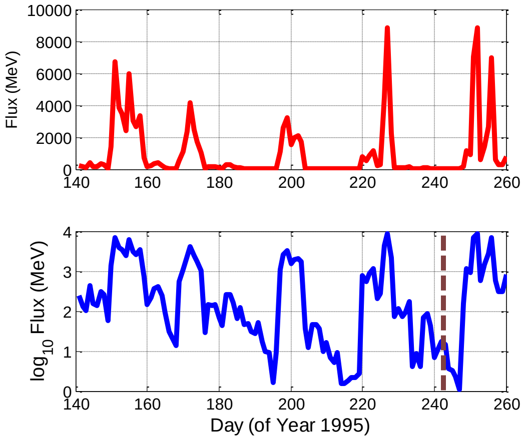
2 days before,	day before,	yesterday,	today	→ tomorrow
$Flux(d-3)$,	$Flux(d-2)$,	$Flux(d-1)$,	$Flux(d)$,	
$V(d-3)$,	$V(d-2)$,	$V(d-1)$,	$V(d)$,	
$VBs(d-3)$,	$VBs(d-2)$,	$VBs(d-1)$,	$VBs(d)$,	→ $Flux(d+1)$ = ??
$P(d-3)$,	$P(d-2)$,	$P(d-1)$,	$P(d)$,	
$SysH(d-3)$,	$SysH(d-2)$,	$SysH(d-1)$,	$SysH(d)$,	
$AsyH(d-3)$,	$AsyH(d-2)$,	$AsyH(d-1)$,	$AsyH(d)$,	

PROGRESS

Forecast of Electron Flux (4)
at the Radiation Belt

The University Of Sheffield.

We use V_{sw}, *VBs*, *Pdyn*, *Sym-H*, and *Asy-H* as inputs, and electron flux (maxima) as output (shown below).



The daily electron flux data: Day 141 - 260 of year 1995 (22 May-17 Sept).

- *141- 243 (22 May -31 Aug) for model identification*
- *244-260 (01 -17 Sept) for model test*

Forecast of Electron Flux (5)
at the Radiation Belt

The University Of Sheffield.

We consider the following multiple input NARX model:

$$
\begin{aligned}
y(k) = f[&y(k-1), \quad y(t-2), \quad y(k-3), \quad y(k-4),\\
&u_1(k-1), \quad u_1(k-2), \quad u_1(k-3), \quad u_1(k-4),\\
&u_2(k-1), \quad u_2(k-2), \quad u_2(k-3), \quad u_2(k-4),\\
&\quad \dots \qquad \dots \qquad \dots \qquad \dots\\
&u_5(k-1), \quad u_5(k-2), \quad u_5(k-3), \quad u_5(k-4)] + e(k)
\end{aligned}
$$

where

$$
\begin{aligned}
y(k) &= flux(k),\\
u_1(k) &= V(k),\\
u_2(k) &= VBs(k),\\
u_3(k) &= Pdyn(k),\\
u_4(k) &= SysH(k),\\
u_5(k) &= AsyH(k),
\end{aligned}
$$

Forecast of Electron Flux (6)
at the Radiation Belt

The University Of Sheffield.

We have applied the OFR-ERR method to the 103 training data (day141-243, 1995), and obtained a simple model containing 6 model terms:

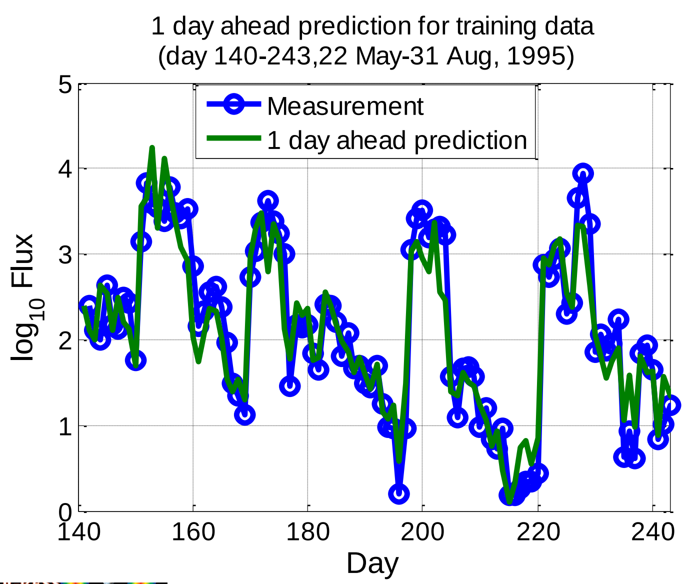
Index	Model term	Parameter	Contribution ERR (100%)
1	Flux(d-1)	0.71090335	92.8682
2	V(d-3)*AsyH(d-1)	0.00008062	0.9910
3	SysH(d-4) *AsyH(d-1)	0.00011492	0.4564
4	VBs(d-3)*VBs(d-4)	0.00000116	0.2947
5	SysH(d-4)	0.03559492	0.1115
6	SysH(d-4)* Pdyn(d-4)	-0.00384037	0.1433

PROGRESS

41/45 (Dr H.L. Wei)

Forecast of Electron Flux (7)
in the Radiation Belt

The University Of Sheffield.



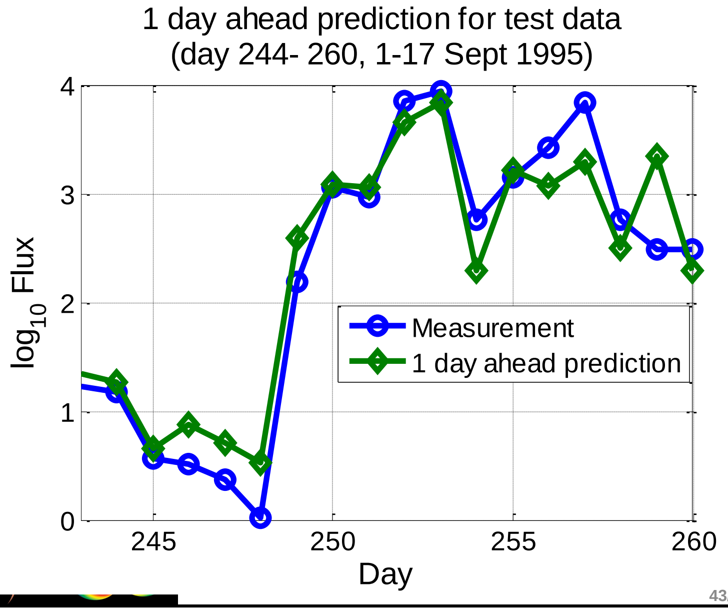
PROGRESS

42/45 (Dr H.L. Wei)

Forecast of Electron Flux (8)

in the Radiation Belt

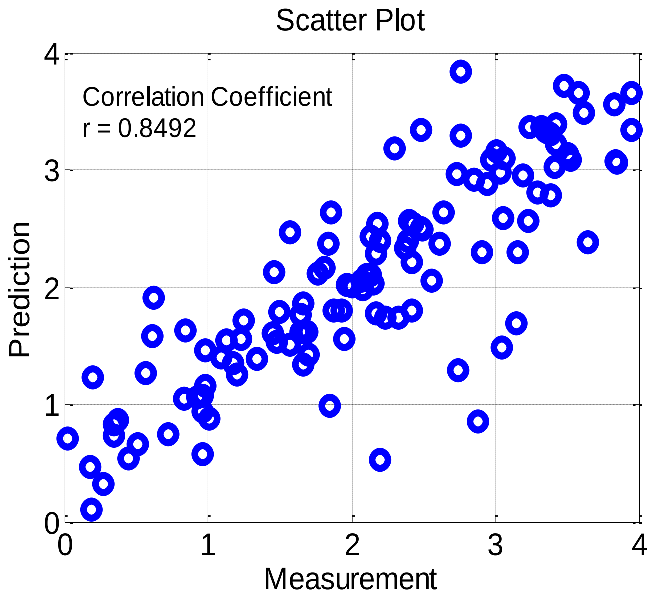
The University Of Sheffield.



Forecast of Electron Flux (9)

at the Radiation Belt

The University Of Sheffield.



Concluding Remarks

◊ The NARMAX and OFR-ERR Methods

- The orthogonal forward regression (OFR) and error reduction ratio (ERR) algorithms provide a powerful tool for compact nonlinear model building from data.
- NARMAX models are transparent and can be written down. This is highly desirable in many scenarios.
- NARMAX method can be used not only for prediction but also more importantly for system analysis. For example, it can detect how the system output relates to the inputs, and how the inputs interact with other.

Acknowledgement

We gratefully acknowledge that part of this work was supported by:

- EC Horizon 2020 Research and Innovation Action Framework Programme (Grant No 637302 and grant title “PROGRESS”).
- Engineering and Physical Sciences Research Council (EPSRC) (Grant No EP/I011056/1)
- EPSRC Platform Grant (Grant No EP/H00453X/1)

THE END