Review of rational (total) nonlinear dynamic system modelling, identification, and control

Quanmin Zhu,∗ Yongji Wang, Dongya Zhao, Shaoyuan Li, and Stephen A. Billings

∗Department of Engineering Design and Mathematics, University of the West of England, Bristol, UK; bDepartment of Control Science and Engineering, School of Automation, Huazhong University of Science and Technology, Wuhan, China; cCollege of Chemical Engineering, China University of Petroleum, Qingdao, China; dDepartment of Automation, Shanghai Jiao Tong University, Shanghai, China; eDepartment of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK

(Received 23 May 2013; accepted 6 September 2013)

This paper is a summary of the research development in the rational (total) nonlinear dynamic modelling over the last two decades. Total nonlinear dynamic systems are defined as those where the model parameters and input (controller outputs) are subject to nonlinear to the output. Previously, this class of models has been known as rational models, which is a model that can be considered to belong to the nonlinear autoregressive moving average with exogenous input (NARMAX) model subset and is an extension of the well-known polynomial NARMAX model. The justification for using the rational model is that it provides a very concise and parsimonious representation for highly complex nonlinear dynamic systems and has excellent interpolatory and extrapolatory properties. However, model identification and controller design are much more challenging compared to the polynomial models. This has been a new and fascinating research trend in the area of mathematical modelling, control, and applications, but still within a limited research community. This paper brings several representative algorithms together, developed by the authors and their colleagues, to form an easily referenced archive for promotion of the awareness, tutorial, applications, and even further research expansion.

Keywords: rational (total) nonlinear dynamic systems; nonlinear rational model set; identification; validation; U-model and U-control

1. Introduction

The nonlinear autoregressive moving average with exogenous input (NARMAX) model set has been extensively studied in theory (Chen & Billings, 1989; Haber & Unbehauen, 1990; Sontag, 1978) and gradually adopted in applications (Proll & Karim, 1994; Wang, 1994). There are two main streams of sub-model sets: polynomial and rational models. A polynomial NARMAX model is defined as linear in the parameters and nonlinear in the regression terms, and can be used to represent a wide range of linear and nonlinear systems. The advantage for identification of polynomial models comes from the fact that the model is linear in the parameters. The rational model (Billings & Chen, 1989; Billings & Zhu, 1991; Sontag, 1979) represents an extension of the polynomial model set and is defined as a ratio of two polynomial expressions. Therefore, rational model is nonlinear in both the parameters and the regression terms. This is induced by the denominator polynomial. Accordingly, the identification and control of rational models are more challenging based on the total nonlinear structures.

Rational models have been gradually adopted in various applications of nonlinear system modelling and control (Ford, Titterington, & Kitsos, 1989; Kambhampati, Mason, & Warwick, 2000; Ponton, 1993), particularly the importance of modelling of chemical kinetics has increased sharply as a consequence of the applicability of modelling of catalytic reactions (Dimitrov and Kamenski, 1991; Kamenski & Dimitrovl, 1993). Rational models are not only alternative expressions in approximating a wide range of data-sets in chemical engineering, but also are a class of mechanistic models, which most of the previous experience or theoretical considerations had not put forward (Dimitrov & Kamenski, 1991).

A number of methodologies and algorithms have been developed for rational model structure detection, parameter estimation, and correlation-based validity tests. These include a prediction error estimator (Billings & Chen, 1989), generalised least squares estimators (Billings & Zhu, 1991; Zhu & Billings, 1991, 1993), recursive least squares parameter estimator (Zhu & Billings, 1993), orthogonal model structure detection and parameter estimation algorithm (Billings & Zhu, 1994a), genetic model structure detection algorithm (Billings & Mao, 1998), back propagation (BP) parameter estimator (2003), implicit least squares parameter estimator (Zhu, 2005), and correlation-based model validity tests (Billings & Zhu, 1994b, 1995; Zhang, Zhu, & Longden, 2007; Zhu, Zhang, & Longden, 2007).

∗Corresponding author. Email: quan.zhu@uwe.ac.uk

© 2013 Taylor & Francis
It should be noted that so far it is still an open research question on how to design rational model-based control systems analytically. It is hoped that the U-model framework explained in this paper will provide a concise analytical solution in future studies.

The rest of the paper is organised into the following sections. In Section 2, rational model is defined to provide a framework for the following algorithm development. Subsequently, an extended least squares (ELS) parameter estimator is presented and a full, step-by-step implementation in computation is listed as a guiding procedure for user’s applications. In Section 3, a BP algorithm is tailored for rational model parameter estimation together with a step-by-step implementation of the algorithm. In Section 4, orthogonal computation is introduced to provide a solution for rational model structure detection (term selection) and parameter estimation. In Section 5, the latest development of correlation-based model validation procedure is presented for validating identified linear and nonlinear models including neural networks and fuzzy logic models. In Section 6, both U-model and U-control are explained conceptually, in terms of the U-transform, for control system design. In Section 7, a brief conclusion is drawn to summarise the results.

2. ELS algorithm for parameter estimation
(Billings & Zhu, 1991)

2.1. Rational (total) model

Rational or total nonlinear dynamic systems can be mathematically described with a ratio of two polynomials and are sometimes commonly known as rational models:

\[
y(t) = \frac{\hat{y}(t) + e(t)}{b(t)} = \frac{a(t)}{b(t)} + e(t) = \frac{a[u(t - 1), \ldots, u(t - n_u), y(t - 1), \ldots, y(t - n_y), e(t - 1), \ldots, e(t - n_e)]}{b[u(t - 1), \ldots, u(t - n_d), y(t - 1), \ldots, y(t - n_d), e(t - 1), \ldots, e(t - n_d) + e(t)],}
\]

where \(y(t)\) and \(\hat{y}(t)\) are the measured and one-step-ahead predicted model outputs, respectively, \(u(t)\) is the input, \(e(t)\) is the model error, and \(t = 1, 2, \ldots\) is the time index. Generally, the numerator \(a(t)\) and denominator \(b(t)\) are functions of past inputs, outputs, and errors, and can be expressed in terms of polynomials:

\[
a(t) = \sum_{j=1}^{num} p_{nj}(t)\theta_{nj}, \quad b(t) = \sum_{j=1}^{den} p_{dj}(t)\theta_{dj}.
\]

The regression terms \(p_{nj}(t)\) and \(p_{dj}(t)\) are the products of past inputs, outputs, and errors, such as \(u(t - 1)y(t - 3)\), \(u(t - 1)e(t - 2)\), \(y^2(t - 1)\), and \(\theta_{nj}, \theta_{dj}\) are the associated parameters. The task of model structure detection is to select candidate regression terms from a pre-assigned large candidate base. The task of parameter estimation, for a given or detected model structure, is to estimate the associated parameters from the measured inputs and outputs. Model validation is the final step to diagnose the feasibility of an identified model if it is a real representative of the underlying system. The correlation-based model validation uses data sequences of input, output, and residual (difference between measured system output and model predicted output) to form a series of correlation functions to determine if the residual has been reduced to an uncorrelated sequence with zero mean and finite variance.

Several remarks relating to the characteristics of the rational model of Equation (2.1) are noted below.

1. The rational model includes almost all the other smooth linear and nonlinear models as its subsets. For example, the polynomial NARMAX model is a special case of rational model (2.1) by setting denominator polynomial \(b(t) = 1\). Many neuro-fuzzy systems have been expressed as nonlinear rational models (Wang, 1994), for example, fuzzy systems with a centre defuzzifier, product inference rule, singleton fuzzifier, and Gaussian membership function. The normalised radial basis function network is also a type of rational model. When the centres and widths are estimated, this can become a rational model parameter estimation problem.

2. The rational model can be much more concise than a polynomial expansion, for example,

\[
y(t) = \frac{u(t - 1)}{1 + y^2(t - 1)} = u(t - 1)[1 - y^2(t - 1) + y^4(t - 1) \ldots].
\]

3. The rational model can be frequently used to represent complicated system structures with a fairly low degree in both the numerator and denominator polynomials. For example, large and quick deviations in a system output can be expressed as

\[
y(t) = \frac{1}{1 + u(t - 1)}. \quad (2.4)
\]

When \(u(t-1)\) approaches \(-1\), the model output response will be quickly increased. The power to capture quick and large changes is gained by introducing the denominator polynomial.
2.2. Bias analysis using the ordinary least squares algorithm

Directly estimating parameters of model (2.1) is difficult in formulation and time consuming in computation even though a prediction error algorithm has been developed (Billings & Chen, 1989). This is because the model is nonlinear in the parameters. An alternative approach is to multiply the denominator polynomial out (2.1) to yield a model expressed as linear in the parameters, then least square algorithms can be applied with proper reformulation. To this end, multiplying the denominator polynomial out (2.1) to yield a model error variance; which is listed below.

\[ \psi = \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & \frac{1}{N} \sum_{k=1}^{N} p_d^T(t) p_d(t) \end{bmatrix} \]

(2.10)

\[ \psi \] is the error vector associated with \( \Phi^T \Psi \) and formulated as

\[ Y(t) = a(t) - y(t) \sum_{j=2}^{\text{num}} p_{dj}(t)\theta_{dj} + b(t)e(t) \]

\[ = \sum_{j=1}^{\text{num}} p_{nj}(t)\theta_{nj} - y(t) \sum_{j=2}^{\text{den}} p_{dj}(t)\theta_{dj} + \xi(t) \]

\[ = \Phi(t)\Theta + \xi(t), \quad (2.5) \]

where

\[ Y(t) = y(t)p_{d1}(t)\theta_{d1} \quad \text{is the dependent variable vector and composed of} \]

\[ \Psi = \begin{bmatrix} Y(1) \cdots Y(N) \end{bmatrix}^T; \quad (2.9) \]

\( \Psi \) is the error matrix associated with \( \Phi^T \Phi \) and formulated as

\[ \psi = \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & \frac{1}{N} \sum_{k=1}^{N} p_d^T(t) p_d(t) \end{bmatrix} \]

An iterative procedure has been developed to obtain the unbiased parameter estimates, which is listed below.

Step 1. Use an ordinary least squares algorithm (set to a null matrix \( \Psi \) and vector \( \psi \) in Equation (2.7)) to compute parameter vector \( \Theta \). This estimate provides initial values for subsequent iterative computations.

Step 2. Compute the model error sequence by \( e(t) = y(t) - \frac{\sum_{j=1}^{\text{den}} p_{nj}(t)\theta_{nj}}{\Phi(t)} \). Then, estimate the model error variance by \( \sigma^2_e = \frac{1}{N} \sum e^2(t) \).

Step 3. Update matrices \( \Phi \) and \( \Psi \), vectors \( \Phi^T \Psi \) and \( \psi \) through formulations (2.8)–(2.11).

Step 4. Use the ELS algorithm (2.7) to estimate parameter vector \( \hat{\Theta} \).

Step 5. Go back to Step 2 and repeat until the parameter estimate vector \( \hat{\Theta} \) and model error variance \( \sigma^2_e \) converge to constant values.

2.4. Statistics of the estimator

The bias and variance of the estimates have been analyzed (Zhu & Billings, 1991) in case of model error \( e(k) \) is reduced to an uncorrelated noise sequence, which are given as follows:

\[ \text{Bias}(\hat{\Theta}) = 0, \]

\[ \text{Cov}(\hat{\Theta}) \approx \sigma^2_e \sigma^2_\epsilon \left[ \Phi^T \Phi - \sigma^2_e \Psi \right]^{-1}, \quad \text{for} \quad \sigma^2_e < 1, \]

\[ \sigma^2_e = E[b^2(t)], \quad E[e] \text{ is the expectation function.} \quad (2.12) \]

2.5. Summary

Expressing the rational model in terms of a linear in the parameters model makes it possible to use least squares algorithms for parameter estimation. On the other hand, the expansion induces inherent correlated errors to give biased estimates while existing linear squares algorithms are applied directly. Therefore, the ELS algorithm has been
developed to set up a basis for concise and unbiased rational model parameter estimation from data sequences.

3. BP algorithm for parameter estimation (Zhu, 2003)

3.1. BP parameter estimation

The rational model can be treated as a neural network, as shown in Figure 1, which is structured with an input layer of regression terms $p_{nj}(t)$ and $p_{dj}(t)$, a hidden layer of $a(t)$ and $b(t)$ with linear activation functions, an output layer of $y(t)$ with a ratio activation function. The activation function at the output layer is the division operation of the numerator polynomial divided by the denominator polynomial.

Leung and Haykin (1993) presented a rational function neural network. The shortcomings compared with the current study are that Leung and Haykin’s network does not have a generalised rational model structure, and correlated errors are not accommodated. Hence Leung and Haykin’s parameter estimation algorithm cannot provide unbiased estimates with noise-corrupted data and is essentially a special implementation of the procedure of Billings and Zhu (1991) and Zhu and Billings (1991) in the case of noise-free data.

To derive the parameter estimation algorithm, with reference to model expressions of Equations (2.5) and (2.6), define the error between the measured output and the model output as

$$e(t) = \frac{1}{2} [y(t) - \hat{y}(t)]^2.$$  \hspace{1cm} (3.1)

The task is to determine a set of parameters in model (2.5) so that the squared error in Equation (3.1) is minimised. The parameter determination can be described as parameter training when the model is interpreted as a neural network structure. To train the parameters $\theta_{nj}$ associated with the numerator polynomial $a(t)$, the local gradients are given by

$$\frac{\partial e(t)}{\partial \theta_{nj}} = -[y(t) - \hat{y}(t)] \frac{\partial e(t)}{\partial a(t)} \frac{\partial a(t)}{\partial \theta_{nj}}$$

$$= -[y(t) - \hat{y}(t)] \frac{1}{b(t)} p_{nj}(t).$$  \hspace{1cm} (3.2)

To train the parameters $\theta_{dj}$ associated with the denominator polynomial $b(t)$, the local gradients are given by

$$\frac{\partial e(t)}{\partial \theta_{dj}} = -[y(t) - \hat{y}(t)] \frac{\partial e(t)}{\partial b(t)} \frac{\partial b(t)}{\partial \theta_{dj}}$$

$$= [y(t) - \hat{y}(t)] \frac{a(t)}{b^2(t)} p_{dj}(t).$$  \hspace{1cm} (3.3)

To carry out recursive computation, a time index $k$ is introduced to the parameter sets $\theta_{nj}(k)$ and $\theta_{dj}(k)$, as they are updated following the time sequence. Convergence has been proved with $\lim_{k \to \infty} \theta_{nj}(k) \to \theta_{nj}$, $\lim_{k \to \infty} \theta_{dj}(k) \to \theta_{dj}$ (Zhu, 2003). Therefore, the parameter variation at time $k+1$ can be defined as follows:

$$\Delta \theta_{nj}(k+1) = \theta_{nj}(k+1) - \theta_{nj}(k) = -\eta_{nj} \frac{\partial e(k+1)}{\partial \theta_{nj}(k+1)},$$

$$\Delta \theta_{dj}(k+1) = \theta_{dj}(k+1) - \theta_{dj}(k) = -\eta_{dj} \frac{\partial e(k+1)}{\partial \theta_{dj}(k+1)}.$$  \hspace{1cm} (3.4)

where $\eta_{nj}$ and $\eta_{dj}$ are the learning rates.

The integrated parameter estimation procedure can be summarised in terms of the BP computation as follows.

(1) Initialisation

Set all model parameters to uniformly distributed random numbers with zero mean and small variance, or set up the initial parameters using the
prior information extraction techniques, such as a least squares estimator.

(2) Parameter estimation

For \( l = 0 \) \(-\) \((L - 1)\) (epoch number)
For \( k = 1 \) \(-\) \(N\) (training data length)
Forward computation:

\[
a(k + 1) = \sum_{j=1}^{\text{num}} p_{nj}(k + 1) \theta_{nj}(k)
\]
\[
b(k + 1) = \sum_{j=1}^{\text{den}} p_{dj}(k + 1) \theta_{dj}(k),
\]
\[
\hat{y}(k + 1) = \frac{a(k + 1)}{b(k + 1)},
\]
\[
e(k+1) = y(k+1) - \hat{y}(k+1).
\]

Backward computation:

\[
\frac{\partial e(k+1)}{\partial \theta_{nj}(k+1)} = -e(k+1) \frac{1}{b(k+1)} p_{nj}(k + 1)
\]
\[
\frac{\partial e(k+1)}{\partial \theta_{dj}(k+1)} = e(k+1) \frac{a(k + 1)}{b^2(k+1)} p_{dj}(k + 1)
\]
\[
\theta_{nj}(k+1) = \theta_{nj}(k) - \eta_{nj} \frac{\partial e(k+1)}{\partial \theta_{nj}(k+1)}
\]
\[
\theta_{dj}(k+1) = \theta_{dj}(k) - \eta_{dj} \frac{\partial e(k+1)}{\partial \theta_{dj}(k+1)}
\]

3.1.1. Initialisation

The first step in the above computation is to initialise the parameters. A sensible choice will provide tremendous help for the resultant parameter estimation. A wrong choice of the initial parameters can lead to two critical phenomena, premature saturation, and divergence. In the case of premature saturation, the instantaneous sum of squared errors tends to become very large. A general initialisation technique (Haykin, 1994) is to uniformly randomise the initial parameters with zero mean and small variance. In this study, the initial parameters will be determined using an ordinary or ELS algorithm.

Although biased, they should not be far away from the true parameters in Euclidean distance.

3.1.2. Learning rate

The back propagation estimator algorithm provides an approximation to the trajectory in parameter space computed in terms of the steepest descent. The smaller the learning rate, the smaller the change to the model parameters will be from one iteration to the next and the smoother the trajectory will be in parameter space. However, a small learning rate will possibly cause slower convergence of the parameter estimation. If the learning rate is set too large, the estimator may become unstable. The choice of learning rates is discussed below:

(1) Constant:

\[
\eta_{nj} = \text{constant}_1 \quad \eta_{dj} = \text{constant}_2.
\]

The disadvantage is that the learning rate cannot properly adjust in the early stage (large learning rate) and in the later stage (small learning rate) of the learning process.

(2) Inversely decayed sequence:

\[
\eta_{nj}(k) = \eta_{dj}(k) = \frac{1}{k}.
\]

The disadvantage is that the learning rate decays too fast and this may not be suitable when long training data sequences are available.

(3) Generalised delta rule (Jacobs, 1988; Rumelhart, Hinton, & Williams, 1986):

\[
\theta_{nj}(k+1) = \theta_{nj}(k) - \eta_{nj} \frac{\partial e(k+1)}{\partial \theta_{nj}(k+1)}
\]
\[
= \theta_{nj}(k) - \eta_{nj} [\theta_{nj}(k) - \theta_{nj}(k - 1)]
\]
\[
\theta_{dj}(k+1) = \theta_{dj}(k) - \eta_{dj} \frac{\partial e(k+1)}{\partial \theta_{dj}(k+1)}
\]
\[
= \theta_{dj}(k) - \eta_{dj} [\theta_{dj}(k) - \theta_{dj}(k - 1)].
\]

where \( \alpha_{nj} \) and \( \alpha_{dj} \) are positive numbers called momentum constants. This has been extensively studied and can increase the learning rate and avoid instability. However, this choice involves more computation and requires the know-how to set up the parameters \( \alpha_{nj} \) and \( \alpha_{dj} \).

(4) Linearly decayed sequence

To overcome the above disadvantages, a new learning rate sequence has been proposed as follows (Zhu, 2003):

\[
\eta_{nj}(k) = \eta_{dj}(k) = \frac{\eta_0 - \eta_{\text{end}}}{LN} (LN + k) + \eta_0,
\]

where \( L \) is the number of iteration epoch, \( N \) is the training data length, \( l = 0, \ldots, L - 1 \) for the \( l \)th epoch, and \( \eta_0 \) \( > \eta_{\text{end}} \) are positive constants for the initial and end learning rates, respectively. Figure 2 shows the learning rate variation.
3.1.3. Stopping criterion

Two criteria (Kramer & Sangiovanni-Vincentelli, 1989) have been used to stop neural network training. These are formulated by means of local or global minimum of the error surface.

(1) Gradient stopping criterion

This states that the BP converges when the Euclidean norm of the gradient vector reaches a sufficiently small gradient threshold. The drawback of this criterion is longer training and complexity in the computation of the gradient vector.

(2) Error measure stopping criterion

This states that the BP converges when the absolute rate of change in the mean squared error per epoch is sufficiently small. Typically, 0.01% per epoch is used. The drawback is that the error may still be correlated for non-linear models, even though the mean squared error or its variation rate is very small (Billings & Voon, 1983; Billings & Zhu, 1994b).

To overcome the above drawbacks, a higher order correlation function, \( R_{χ^2}(τ) \) and \( R_{χ^2}(τ) \), satisfy

\[
R_{χ^2}(τ) = \begin{cases} 
k > 0 \quad τ = 0 \\
0 \quad \text{otherwise}
\end{cases} \quad \text{(3.14)}
\]

the estimated parameters are considered to be unbiased. Otherwise, the training procedure will continue until the above conditions are satisfied. In practice, the 95% confidence limits, \( \pm 1.96/\sqrt{N} \), are used as the stopping thresholds.

3.2. Statistics of the estimator

For the bias,

\[
\text{Bias} \ θ_{nj} = θ_{nj} - E[θ_{nj}(k + 1)] = E[e(k + 1)] E\left[\frac{1}{b(k + 1)} p_{nj}(k + 1)\right] = 0.
\]

\[
\text{Bias} \ θ_{dj} = θ_{dj} - E[θ_{dj}(k + 1)] = E[e(k + 1)] E\left[\frac{a(k + 1)}{b^2(k + 1)} p_{nj}(k + 1)\right] = 0.
\]

Therefore, the estimator will produce unbiased estimates of the nonlinear rational model parameters. For the variance,

\[
\text{Var} \ θ_{nj} = E[(θ_{nj}(k + 1) - E[θ_{nj}(k + 1)])^2] = σ_{χ^2}^2 σ_{nj}^2,
\]

\[
\text{Var} \ θ_{dj} = E[(θ_{dj}(k + 1) - E[θ_{dj}(k + 1)])^2] = σ_{χ^2}^2 σ_{dj}^2,
\]

where

\[
σ_{nj}^2 = E\left[\left(\frac{1}{b(k + 1)} p_{nj}(k + 1)\right)^2\right]
\]

\[
σ_{dj}^2 = E\left[\left(\frac{a(k + 1)}{b^2(k + 1)} p_{dj}(k + 1)\right)^2\right]. \quad \text{(3.17)}
\]

Therefore, the variance of each parameter estimate is associated with the noise variance and the second-order moment of the regression term.

3.3. Summary

The BP-based algorithm provides an alternative approach to estimate rational model parameters. Obviously, there is no need to expand rational models into linear in the parameter expressions. Some techniques, such as learning rate and stopping threshold, play important roles in the iterative
training and learning process, which need to be carefully considered in the computations.

4. Orthogonal least squares (OLS) algorithm for term selection and parameter estimation (Zhu & Billings, 1993; Billings & Zhu, 1994a)

4.1. Orthogonal parameter estimation
Consider an orthogonal transformed expression of Equation (2.5):

$$Y(t) = \sum_{j=1}^{num} p_{nj}(t)\theta_{nj} - y(t) \sum_{j=2}^{den} p_{dj}(t)\beta_{dj} + \zeta(t)$$

$$= \omega(t)G + \zeta(t)$$

$$= \sum_{j=1}^{num} w_{nj}(t)g_{nj} - y(t) \sum_{j=2}^{den} w_{dj}(t)g_{dj} + \zeta(t), \quad (4.1)$$

where

$$\omega(t) = [\omega_n(t) \quad \omega_d(t)] = [\omega_n(t) \cdots \omega_{num}(t) \omega_{d2}(t) \cdots \omega_{d(den)}(t)]$$

$$G^T = [G_n \quad G_d] = [g_{n1} \cdots g_{num} \quad g_{d1} \cdots g_{dden}]. \quad (4.2)$$

Define the orthogonal regression matrix $W$ as

$$W = \Phi^T^{-1}$$

$$W^T = [\omega^T (1) \quad \omega^T (N)] = \left[\begin{array}{cccc} \omega^T_n(1) & \cdots & \omega^T_n(N) \\ \omega^T_d(1) & \cdots & \omega^T_d(N) \end{array} \right]$$

$$T = \left[ \begin{array}{cccc} 1 & t_{12} & \cdots & t_{1n} \\ 0 & 1 & \cdots & . \\ . & 0 & 1 & \cdots \\ . & . & \cdots & . \\ . & . & \cdots & t_{n-1n} \\ 0 & 0 & \cdots & 1 \end{array} \right], \quad (4.3)$$

where $\Phi$ is the regressor matrix defined in Equation (2.8). There are several typical algorithms to compute the elements of $T$, such as Gram–Schmidt, Householder, or Givens transforms (Billings & Zhu, 1994a). The orthogonality of the matrix $W$ gives

$$W^TW = D = \text{diag} \{ d_{n1} \cdots d_{num} \quad d_{d1} \cdots d_{dden} \}. \quad (4.4)$$

The unbiased parameter vector $G$ is estimated with

$$\hat{G} = [W^TW - \sigma_e^2 \Psi_{orth}]^{-1}[W^TY - \sigma_e^2 \Psi_{orth}], \quad (4.5)$$

where

$$\Psi_{orth} = \text{diag} \left[ \frac{1}{e_{n1}^2} \cdots \frac{1}{e_{num}^2} \frac{1}{e_{d2}^2} \cdots \frac{1}{e_{dden}^2} \right],$$

$$\Psi_{orth} = \left[ \begin{array}{cccc} p_{d1}e_{n1} & \cdots & p_{d1}e_{num} & e_{d1} \\ \cdots & \cdots & \cdots & \cdots \\ p_{d1}e_{d2} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ p_{d1}e_{dden} \end{array} \right]^T, \quad (4.6)$$

where over-bar denotes time average value. The computation of the elements in Equation (4.6) can be found in Zhu and Billings (1993). Accordingly, the original parameter vector in Equation (2.5) can be recovered by

$$\Theta = T^{-1}G. \quad (4.7)$$

4.2. Structure detection (term selection)
The orthogonal property can be applied to select regression terms and estimate the associated parameters simultaneously. A criterion of error reduction ratio (ERR) has been introduced to select the most important regression terms to contribute to reduce estimation errors from a large candidate term base (Billings & Zhu, 1994a). In definition, individual ERR from numerator and denominator polynomials is calculated, respectively, by

$$\text{err}_{nj} = \frac{g_{nj}^2 e_{nj}^2 \sigma_n^2}{\sigma_n^4 \sigma_b^2} + 2g_{nj} e_{nj} \sigma_n^2 \sigma_b^2, \quad (4.8)$$

$$\text{err}_{dj} = \frac{g_{dj}^2 e_{dj}^2 \sigma_d^2}{\sigma_d^4 \sigma_b^2} + 2g_{dj} e_{dj} \sigma_d^2 \sigma_b^2, \quad (4.8)$$

$$\sigma_l^2 = \frac{1}{N} \sum Y^2(k) \quad \sigma_b^2 = \frac{1}{N} \sum b^2(k).$$

It should be noticed that the larger the ERR, the more contribution of its regression term to reduce the estimation errors. Therefore, the candidate term with the largest ERR in each selection sequence should be taken in as the model regression terms.

An iterative learning procedure for rational structure detection has been developed as follows.

Step 1. Fit a deterministic rational model (that is no noise model) initially with 3–4 regression terms (choice of the terms comes from experience, actually it is not critical) such as Gram–Schmidt, Householder, or Givens transforms. In definition, a criterion of error reduction ratio (ERR) has been introduced to select the most important regression terms and estimate the associated parameters simultaneously. A criterion of error reduction ratio (ERR) has been introduced to select the most important regression terms to contribute to reduce estimation errors from a large candidate term base (Billings & Zhu, 1994a). In definition, individual ERR from numerator and denominator polynomials is calculated, respectively, by

$$\text{err}_{nj} = \frac{g_{nj}^2 e_{nj}^2 \sigma_n^2}{\sigma_n^4 \sigma_b^2} + 2g_{nj} e_{nj} \sigma_n^2 \sigma_b^2, \quad (4.8)$$

$$\text{err}_{dj} = \frac{g_{dj}^2 e_{dj}^2 \sigma_d^2}{\sigma_d^4 \sigma_b^2} + 2g_{dj} e_{dj} \sigma_d^2 \sigma_b^2, \quad (4.8)$$

$$\sigma_l^2 = \frac{1}{N} \sum Y^2(k) \quad \sigma_b^2 = \frac{1}{N} \sum b^2(k).$$

Set up a series of ascending weights of the model variance $\sigma^2_e$ (such as 0.1, 0.25, 0.5, 0.75, 0.95, 1, 1, . . . , 1).

Set up a cut-off point (COP), which can be determined by trial and error approach, as

$$\text{COP} \geq 1 - \frac{1}{\sigma_b^2} + \sum \text{bias [err]} + \frac{\sigma_e^2}{\sigma_f^2},$$

$$\text{bias [err]} = \left[ \begin{array}{c} \frac{g_{nj}^2 e_{nj}^2}{\sigma_n^4 \sigma_b^2} + 2g_{nj} e_{nj} \sigma_n^2 \sigma_b^2 \\ \frac{g_{dj}^2 e_{dj}^2}{\sigma_d^4 \sigma_b^2} + 2g_{dj} e_{dj} \sigma_d^2 \sigma_b^2 \end{array} \right],$$

where numerator terms

$$\text{bias [err]} = \left[ \begin{array}{c} \frac{g_{nj}^2 e_{nj}^2}{\sigma_n^4 \sigma_b^2} + 2g_{nj} e_{nj} \sigma_n^2 \sigma_b^2 \\ \frac{g_{dj}^2 e_{dj}^2}{\sigma_d^4 \sigma_b^2} + 2g_{dj} e_{dj} \sigma_d^2 \sigma_b^2 \end{array} \right],$$

where denominator terms.
Step 3. Do a search through a pre-setup full rational model candidate term base to select significant terms according to the ERR values and estimate their associated parameters.

Step 4. Repeat Steps 2 and 3 until the computations converge, the preset maximum number of the iterations exceeded, or a specified number of terms are selected in the model.

4.3. Summary
This algorithm delivers a more realistic procedure to fit models to data in case of which terms should be included in the models before estimating their associated parameters.

5. Model validation (Zhang, Zhu, & Longden, 2007, 2009; Zhu, Zhang, & Longden, 2007; Zhu, Zhang, & Longden, 2010)

5.1. Introduction to correlation-based model validation
Consider a generalised single-input and single-output (SISO) nonlinear parametric model:

\[ y(t) = \hat{y}(t) + \varepsilon(t) = \hat{f}(y^{t-1}, u^{t-1}, \varepsilon^{t-1}) + \varepsilon(t), \]  

(5.1)

where \( \hat{f}(\cdot) \) is the identified nonlinear model. \( y^{t-1}, u^{t-1}, \) and \( \varepsilon^{t-1} \) are measured output, input, and residual (the difference between the measured output and the one step ahead model predicted output) vectors with delayed elements from 1 to \( r \), respectively. It should be noticed that when above model is properly identified, the residual \( \varepsilon(t) \) should be reduced to a random noise sequence denoted by \( e(t) \) in this section (without being confused with those used in other sections) with zero mean and finite variance (Billings & Zhu, 1994a). In other words, the residual sequence from a properly identified model should be uncorrelated to the delayed residuals, inputs, and outputs (Ljung, 1999).

It should be noticed that all validation methods developed based on nonlinear models have included all linear model validation as their simplified cases. However, the linear model-based validation test methods can and often do fail when applied to nonlinear model validation.

5.2. Combined ODACF and ODCCF for model validation
Combined omnidirectional auto-correlation function (ODACF) is defined as follows:

\[
\rho_{\text{dc}}(\tau) = \begin{cases} 
0.5 & \text{if} \ |\max(R_{\text{dc}})| < |\min(R_{\text{dc}})| \\
\max(R_{\text{dc}}) & \text{else}
\end{cases}
\]  

(5.2)

Combined omnidirectional cross-correlation functions (ODCCF) is defined as follows:

\[
\begin{align*}
\rho_{\text{cc}}(\tau) &= \begin{cases} 
0.5 & \text{if} \ |\max(R_{\text{cc}})| < |\min(R_{\text{cc}})| \\
\max(R_{\text{cc}}) & \text{else}
\end{cases} \\
\rho_{\text{cc}}(\tau) &= \min(R_{\text{cc}}).
\end{align*}
\]  

(5.3)

In the above definition,

\[
\begin{align*}
R_{\text{cc}} &= \{r_{u^c}(\tau), r_{u^c}(\tau), r_{e^c}(\tau), r_{e^c}(\tau), r_{u^c}(\tau), r_{u^c}(\tau)\} \\
R_{\text{dc}} &= \{r_{u^d}(\tau), r_{u^d}(\tau), r_{e^d}(\tau), r_{e^d}(\tau), r_{u^d}(\tau), r_{u^d}(\tau)\},
\end{align*}
\]  

(5.4)

\[
\begin{align*}
\alpha'(k) &= |\varepsilon'(k)| - \frac{1}{N} \sum_{k=1}^{N} |\varepsilon'(k)| \\
\beta'(k) &= |u'(k)| - \frac{1}{N} \sum_{k=1}^{N} |u'(k)|,
\end{align*}
\]  

(5.5)

where \( r_{cc}(\tau) \) denotes correlation function and the prime (‘) in Equations (5.4) and (5.5) denotes that the mean level has been removed from the corresponding data sequence.

Then, the validity tests for a properly identified model are derived as

\[
\begin{align*}
\rho_{\text{cc}}(\tau) &= \begin{cases} 
1, \tau = 0 \\
0, \text{otherwise}
\end{cases} \\
\rho_{\text{cc}}(\tau) &= 0, \forall \tau.
\end{align*}
\]  

(5.6)

(5.7)

Compared with the other correlation test-based methodologies, this approach enhances the power of nonlinear model validity tests and significantly reduces the number of correlation plots. For large \( N \), the correlation function estimates given in Equations (5.2) and (5.3) are still asymptotically normal with zero mean and finite variance in accordance with the central limit theorem (Bowker & Lieberman, 1972). The standard deviations are \( 1/\sqrt{N} \) and the 95% confidence limits are therefore approximately at \( \pm 1.95/\sqrt{N} \).

5.3. Summary
It should be mentioned that the method is not only applicable to rational models, but also for validation of identified neural networks and fuzzy logic models (Zhang, Zhu, & Longden 2009).

6. U-model and U-control (Zhu, 1989; Zhu & Guo, 2002)

6.1. Introduction
So far, almost all control system design approaches, no matter if they are linear or nonlinear plant-based, have taken a unique procedure, set up a specified control system performance/index and then by inverting the integrated function of the plant and design performance in some way, obtain the controller output (i.e., the input to the plant).
There is no problem at all with such procedures for all linear plant-based control system design subject to stability considerations, because the linear superposition principle makes the inverse function easily resolved. However, when the plants are subject to nonlinear dynamic equations (particularly non-affine models), the inverse functions are more complicated, intractable, or even impossible with analytical solutions, except step-by-step numerical computations.

**Research question 1:** Can a class of nonlinear dynamic plants, described by smooth nonlinear models in terms of polynomials, be designed directly using the approaches developed from linear control systems?

**Research question 2:** Is there a general approach to resolve the inverse function of nonlinear plants?

With such insight, the U-model methodology has been proposed, where the origin, but not in general U-model expression, appeared in one of the authors’ Ph.D. thesis (Zhu, 1989). In progression along this route, Zhu, Warwick and Douce (1991) further proposed to use a Newton–Raphson algorithm (Gerald, 1978) root solver to obtain the controller output, which makes the U-model type of control system feasible. The first time the U-model was named in a study of pole-placement controller design for nonlinear plants was by Zhu and Guo (2002), which is a simple mapping from ordinary linear and nonlinear differential equations to time-varying polynomials in terms of the plant input $u(t - 1)$ (i.e., the controller output). The U-model covers almost all existing smooth nonlinear discrete time models as subsets.

Since then, within the first decade, the U-model has been used in designing feedforward control of multi-input and multi-output nonlinear systems (Ali, Fouad, Muhammad, & Jamil, 2010), nonlinear leaking minimum square and multi-output nonlinear systems (Ali, Fouad, Muhammad, & Tahir, 2005), minimum square error IMC (internal model control) (Muhammad & Haseebiddon, 2005), and general predictive control of nonlinear plants (Du, Wu, & Zhu, 2012) have also been investigated. It should be noted that the U-model associated publications are still in a very beginning stage. These have had no rigorous analytical description and top journal publication till today.

**The other research question:** How is the U-model design procedure different from classical procedures in terms of efficiency and effectiveness?

The questions have been the justification for initiating the study. Xu, Zhu, Zhao and Li (2013) have presented a comprehensive survey for the first decade of development on U-model-based control system design.

### 6.2. U-model

Consider a general SISO discrete time dynamic plant described by the following polynomial model (P-model):

$$ y(t) = f(P(*), \Theta) $$

$$ P(*) = P(y(t - 1), \ldots, y(t - n), u(t - 1), \ldots, u(t - n)) $$

(6.1)

where $y(t) \in R$ and $u(t - 1) \in R$ are the output and input signals of the plant, respectively, at discrete time instant $t(1, 2, \ldots)$ and $n$ is the plant order. $P(*) = P(y(t - 1), \ldots, y(t - n), u(t - 1), \ldots, u(t - n)) \in R^{k+1}$ is the regression variable vector spanned from the delayed outputs and inputs, and $\Theta = [\theta_0 \cdots \theta_L] \in R^{k+1}$ is the associated parameter vector. Function $f(\cdot)$ is a smooth linear or nonlinear function. The P-model can be further expressed in terms of regression equation as below:

$$ y(t) = \sum_{l=0}^{L} p_l(t) \theta_l, $$

where the regression terms $p_l(t)$ are the products of past inputs and outputs such as $u(t - 1) y(t - 3), u(t - 1) u(t - 2)$, $y^{i}(t - 1)$, and $\theta_l$ are the associated parameters. Typically, for example, linear time-invariant difference equation-based plant models and NARMA (nonlinear autoregressive moving average) models.

The U-model is defined as, under a U-mapping from P-model, a control-oriented model expression below:

$$ y(t) = f(\Upsilon(*), U(t - 1)), \Upsilon(*) = \Upsilon(y(t - 1), \ldots, y(t - n), u(t - 2), \ldots, u(t - n), \Theta), U(t - 1) = \{ \text{const} u(t - 1) u^2(t - 1) \cdots u^M(t - 1) \}.$$

Correspondingly, its regression equation is given as below:

$$ y(t) = \sum_{j=0}^{M} \lambda_j(t) u^j(t - 1). $$

(6.2)

This is expanded from the above nonlinear function $f(\cdot)$ as a polynomial with respect to $u(t - 1)$, where $M$ is the degree of model input $u(t - 1)$, parameter $\lambda_j(t)$ is a function of past inputs, outputs $(u(t-2), \ldots, u(t-n), y(t-1), \ldots, y(t-n))$, and the parameters $(\theta_0 \cdots \theta_L)$.

### 6.3. U-control – U-model-based control system design procedure

Define the desired plant output as $U(t)$, it is clear that the setting,

$$ y(t) = U(t), $$

then

$$ U(t) = \sum_{j=0}^{M} \lambda_j(t) u^j(t - 1). $$

(6.4)
Accordingly, the task of the design is to determine the desired plant output $U(t)$ according to a specified performance index, for example, pole placement control (PPC) (Zhu & Guo, 2002), $RU(t) = Tw(t) - Sy(t)$ and general predictive control (GPC) (Du, Wu, & Zhu, 2012), $J_p = E[Y_d ^TQY_d + \Delta U^T \lambda \Delta U]$, and then by resolving one of the roots of Equation (6.4) to obtain the controller output. The block diagram, Figure 3, shows a general U-model-based closed-loop control system structure. It should be noted that so far it is still an open research question of how to use the U-model approach to design rational model-based control systems. Therefore, fundamental research should be performed in the future research development.

6.4. Summary

Design of nonlinear dynamic control systems has been widely recognised as a challenging issue. The key point in the design is a general model prototype with conciseness, flexibility, and manipulability, while keeping little loss of precision. This is the origin of the U-model with insight and procedure. In terms of time-varying parameter polynomials, the U-model almost covers all existing smooth nonlinear discrete time models as its subsets. Based on the U-prototype, without complex transformation to its original model and no need for linearisation at all, a nonlinear control system can be directly designed by linear control system design approaches, such as PPC, GPC, sliding mode control (SMC), and so on. Even for linear control system design, the U-model provides another procedure for solutions.

The major contribution of the U-model-based design procedure can be listed in the following order.

1. In methodology, those well-known approaches developed from linear systems can be directly applied to nonlinear control system design, which significantly reduces the design complexity and effectively provides straightforward computational algorithms.

2. In design, it obtains desired plant output first (compared to designing desired controller output in classical framework) and then works out the controller output from U-model in a relaxed root-resolving routine (compared to resolving complex solutions from the inverse of the whole designed systems).

3. For linear control system design, it provides new insight and solutions within a more general and effective frame work.

7. Conclusions

With regard to the presented algorithms, the ELS algorithm sets up a solid and concise basis for expanding this classical approach into nonlinear rational model term selection and parameter estimation. The insight presented in the study could be a useful indication to stimulate other complex model identification with proper extension of classical least squares algorithms. The BP algorithm provides a different angle to study the model parameter estimation issues. However, the model structure detection via PB computation (i.e., regression term selection) still remains unresolved. The OLS algorithm provides a step-by-step procedure to select model regression terms and estimate their associated parameters simultaneously. The new model validation method provides a generic routine to effectively examine the model validity with significantly reduced correlation plots. In control system design, the U-model and hence U-control provide a new procedure from a classical framework, which could provide a concise solution for rational model-based control systems design in the future studies. It should be noted again that the structure of the rational model brings various advantages, which cannot be easily contributed through polynomial model sets. On the other hand, the rational model presents a number of new challenges for research, demonstration, and applications.

Acknowledgements

The authors are grateful to the editors and the anonymous reviewers for their helpful comments and constructive suggestions with regard to the revision of the paper.

Funding

This work is partially supported by the National Nature Science Foundation of China [grant number 61004080], [grant number 61273188]; Shandong Provincial Natural Science Foundation, China [grant number ZR2011FM003]; the Fundamental Research Funds for the Central Universities of China; Development of Key Technologies Project of Qingdao Economic and Technological Development Zone [grant number 2011-2-52]; Taishan Scholar Construction Engineering Special funding.
Notes on contributors

Quanmin Zhu is professor in Control Systems at Faculty of Environment and Technology, University of the West of England, Bristol, UK. He obtained his MSc degree from Harbin Institute of Technology, China in 1983 and PhD degree from Faculty of Engineering, University of Warwick, UK in 1989. His main research interest is in the area of nonlinear system modelling, identification, and control. His other research interests are in investigating electrodynamics of acupuncture points and sensory stimulation effects in human body, modelling of human meridian systems, and building up electro-acupuncture instruments. Currently, Professor Zhu is acting as a member of Editorial Committee of Chinese Journal of Scientific Instrument, an editor (and founder) of International Journal of Modelling, Identification and Control, and an editor of International Journal of Computer Applications in Technology.

Yongji Wang received the undergraduate degree in electrical engineering from Shanghai Railway University, Shanghai, P.R. China, the MS degree and the PhD degree in automation from Huazhong University of Science and Technology, Wuhan, P.R. China, in 1982, 1984, and 1990, respectively. He has been with Huazhong University of Science and Technology, Wuhan, P.R. China, since 1984, where he is currently a professor of Control Engineering. His main interest is in intelligent control and autonomous mobile robot, and he has done research in neural network control, predictive control, adaptive control, and most recently, flight vehicle optimal control. Dr. Wang is a member of IEEE, the president of Hubei Automation Association, China, a member of council of the Chinese Automation Association, standing member of council of Intelligent Robot Committee of Chinese Artificial Intelligence Society. He is a regional editor (Asia and Pacific) of International Journal of Modelling Identification and Control.

Dongya Zhao received BEng degree from Shandong University, Jinan, China, in 1998, MSc degree from Tianhua Institute of Chemical Machinery & Automation, Lanzhou, China, in 2002, and PhD degree from Shanghai Jiao Tong University, Shanghai, China, in 2009. He was a research fellow in Nanyang Technological University during July 2011 to July 2012. Since 2002, he has been with College of Chemical Engineering, China University of Petroleum, where he is currently an associate professor. His research interests include robot control, sliding mode control, process modelling and control, and nonlinear system control and analysis.

Shaoxuan Li received the PhD degree from the Department of Computer and System Science, Nan Kai University, in 1997. He was a lecturer in the Department of Electrical Engineering, Hebei University of Technology, Tianjin from 1994 to 1998, and was with Department of Automation, Shanghai Jiao Tong University as the postdoctoral research fellow from 1998 to 2000. Since then, he has been with the Department of Automation, Shanghai Jiao Tong University, China, and he is currently a full professor in this department, and in charge of the head of this department from 2004 to 2013. His research interests cover system identification and controller tuning, adaptive control, satisfying optimization and predictive control, and so on. He is the author or co-author of four monographs and about 200 technical papers.

Stephen A. Billings is a professor in the Department of Automatic Control and Systems Engineering, University of Sheffield, UK, and leads the Signal Processing and Complex Systems research group. His research interests include system identification and information processing for nonlinear systems, NARMAX methods, model validation, prediction, spectral analysis, adaptive systems, nonlinear systems analysis and design, neural networks, wavelets, fractals, machine vision, cellular automata, spatiotemporal systems, fMRI and optical imagery of the brain, synthetic biology, and related fields.

References

Ali, S.S.A., Fouad, M.A., Muhammad, S., & Jamil, M.B. (2010). U-model based learning feedforward control of MIMO nonlinear systems. Electrical Engineering, 93(8), 405–415.

Billings, S.A., & Chen, S. (1989). Identification of non-linear rational systems using a prediction error estimation algorithm. International Journal of Systems Science, 20, 467–494.

Billings, S.A., & Mao, K.Z. (1998). Structure detection for nonlinear rational models using genetic algorithms. International Journal of Systems Science, 29, 223–231.

Billings, S.A., & Voon, W.S.F. (1983). Structure detection and model validity tests in the identification of nonlinear systems. IEE Proceedings, D, 130, 190–199.

Billings, S.A., & Zhu, Q.M. (1991). Rational model identification using extended least squares algorithm. International Journal of Control, 54, 529–546.

Billings, S.A., & Zhu, Q.M. (1994a). A structure detection algorithm for nonlinear dynamic rational models. International Journal of Control, 59, 1339–1463.

Billings, S.A., & Zhu, Q.M. (1994b). Nonlinear model validation using correlation tests. International Journal of Control, 60, 1107–1120.

Billings, S.A., & Zhu, Q.M. (1995). Model validation tests for multivariable nonlinear models including neural networks. International Journal of Control, 62, 749–766.

Bowker, A.H., & Lieberman, G.J. (1972). Engineering statistics. Englewood Cliffs, NJ: Prentice Hall.

Butt, N.R., Muhammad, S., & Tahir, K. (2005). An adaptive root-solving controller for tracking of nonlinear dynamic plants. Proceeding of the 2005 IEEE International Conference on Industrial Electronics and Control Applications, Quito, Ecuador, pp. 144–149.

Chen, S., & Billings, S.A. (1989). Representations of nonlinear system: The NARMAX model. International Journal of Control, 48, 1013–1032.

Dimitrov, S.D., & Kamenski, D.I. (1991). A parameter estimation method for rational functions. Computers and Chemical Engineering, 15, 657–662.

Du, W.X., Wu, X.L., & Zhu, Q.M. (2012). Direct design of U-model based generalized predictive controller (UMGPC) for a class of nonlinear (polynomial) dynamic plants. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 226, 27–42.
Ford, I., Titterington, D.M., & Kitsos, C.P. (1989). Recent advances in non-linear experimental design. *Technometrics, 31*, 49–60.

Gerald, C.F. (1978). *Applied numerical analysis*. Reading, MA: Addison-Wesley.

Haber, R., & Unbehauen, H. (1990). Structure identification of nonlinear dynamic system – A survey on input/output approaches. *Automatica, 26*, 651–677.

Haykin, S. (1994). *Neural networks*. Oxford: Macmillan College.

Jacobs, R.A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks, 1*, 295–307.

Kambhampati, C., Mason, J.D., & Warwick, K. (2000). A stable one-step-ahead predictive control of nonlinear systems. *Automatica, 36*, 485–495.

Kamenski, D.I., & Dimitrovl, S.D. (1993). Parameter estimation in differential equation by application of rational functions. *Computers and Chemical Engineering, 17*, 643–651.

Kramer, A.H., & Sangiovanni-Vincentelli, A. (1989). Efficient parallel learning algorithms for neural networks. In D.S. Touretzky (Ed.), *Advances in neural information processing systems* (pp. 40–48). San Mateo, CA: Morgan Kaufmann.

Leung, H., & Haykin, S. (1993). Rational function neural network. *Neural Computation, 5*, 928–938.

Ljung, L. (1999). *System identification theory for the user* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Muhammad, S., & Haseebiddon, M. (2005). U-model-based internal model control for non-linear dynamic plants. *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering, 219*(6), 449–458.

Ponton, J.W. (1993). The use of multivariable rational functions for non-linear data presentation and classification. *Computers and Chemical Engineering, 17*, 1047–1052.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning representations by back propagating errors. *Nature (London), 323*, 533–536.

Sontag, E.D. (1978). Realization theory of discrete time nonlinear systems: Part I – The bounded case. *IEEE Transaction on Circuits and Systems, 26*, 342–356.

Zhu, Q.M. (1989). *Identification and control of nonlinear systems* (doctoral dissertation). University of Warwick, Coventry.

Zhu, Q.M. (2003). A back propagation algorithm to estimate the parameters of nonlinear dynamic rational models. *Applied Mathematical Modelling, 27*, 169–187.

Zhu, Q.M. (2005). An implicit least squares algorithm for nonlinear rational model parameter estimation. *Applied Mathematical Modelling, 29*, 673–689.

Zhu, Q.M., & Billings, S.A. (1991). Recursive parameter estimation for nonlinear rational models. *Journal of Systems Engineering, 1*, 63–67.

Zhu, Q.M., Warwick, K., & Douce, J.L. (1991). Adaptive general predictive controller for nonlinear systems. *IEE Proceedings – Control Theory and Applications, 138*(1), 33–40.

Zhu, Q.M., & Guo, L.Z. (2002). A pole placement controller for nonlinear dynamic plants. *Journal of Systems and Control Engineering, Proceedings of the Institution of Mechanical Engineers Part I, 216*(6), 467–476.

Zhu, Q.M., & Longden, A. (2007). Development of omni-directional correlation functions for nonlinear model validation. *Automatica, 43*, 1519–1531.

Zhu, Q.M., & Longden, A. (2010). A correlation test-based validity monitoring procedure for online detecting the quality of nonlinear adaptive noise cancellation. *International Journal of Systems Science, 41*, 1043–1055.